M46E アプリ操作説明書

目次

1. はじめに

本ドキュメントでは、M46E アプリケーション(以下、M46E アプリ)の概要と基本的な操作方法について説明します。また、具体的なネットワーク構成を基にした設定ファイルの記述例についても記載しています。

2. 動作環境

2.1. OS

M46E アプリは Namespace 機能および、macvlan デバイス機能がサポートされている Linux カーネルを搭載した OS 上で動作します。以下のディストリビューションについては、動作確認済みです。

名称	バージョン	カーネルバージョン	備考
CentOS	6.3	2.6.32	
Ubuntu	11.10	3.0.0	
Debian GNU/Linux	6.0(squeeze)	3.0.6 改(※)	OpenBlocks AX3(ぷらっとホーム社製)上に Namespace サポートカーネルを組み込んだ環境で確認。

2.2. ファイル構成

本アプリケーションは以下のファイルで構成されています。

ファイル名	説明
m46eapp	M46E アプリ 実行ファイル
m46ectl	サポートコマンド 実行ファイル
m46e.conf	設定ファイル。ファイル名は任意。
m46e_startup.sh	スタートアップスクリプト(オプション)。ファイル名は任意。

3. アプリケーションの設定

M46E アプリの動作には、設定ファイルが必要になります。この章では、設定ファイルで設定可能な項目の概要を説明します。巻末に付録として設定ファイルのサンプルを掲載しているので、そちらも参考にしてください。

3.1. 共通設定

アプリケーションに共通な設定項目です。設定ファイルの[general]セクションに定義されており、パラメータには以下のものがあります。

plane_name

M46E アプリで管理する Plane の名前。

M46E アプリを識別する為の識別子として使用します。
サポートコマンドで指定するアプリケーション名にはこのパラメータで設定した値を使用します。
同一ホスト内で複数の M46E アプリを起動する場合は、他と重複しない値を設定してください。
このパラメータは省略できません。
設定例）

```
plane_name = m46e0
```

plane_id

M46E アプリの属する Plane ID。
IPv6 アドレスと同じ形式で HEX を 16bit 毎に ： 区切りで設定してください。
このパラメータは省略可能です。省略した場合は Plane ID 無し(ALL0)として扱われます。
設定例）

```
plane_id = 0:64
```

tunnel_mode

M46E アプリの動作モード。
以下の値が設定可能です。

- 0：通常モード
- 1：M46E-AS モード
- 2：M46E-PR モード

このパラメータは省略できません。
設定例）

```
tunnel_mode = 0
```

m46e_unicast_prefix

M46E の unicast prefix address。
IPv4 ユニキャストパケットをカプセル化する際の prefix アドレスとして使用されます。
IPv6 ユニキャストアドレス形式で[アドレス/prefix 長]のフォーマットで設定してください。
tunnel_mode が通常モード(0)の場合、prefix 長は最大 96 までとなり、M46E-AS モード(1)の場合、prefix 長は最大 80 までとなります。tunnel_mode が M46E-PR モード(2)の場合、本パラメータは使用されません。
このパラメータは省略できません。
設定例）

```
m46e_unicast_prefix = 2001:db8:0:46::/64
```

m46e_pr_src_addr_unicast_prefix

M46E-PR の送信元 unicast prefix address。
IPv4 ユニキャストパケットをカプセル化する際の src prefix アドレスとして使用されます。
IPv6 ユニキャストアドレス形式で[アドレス/prefix 長]のフォーマットで設定してください。
tunnel_mode が M46E-PR モード(2)の場合のみ有効で、prefix 長は最大 96 までとなります。
このパラメータは省略できません。

設定例）

```
m46e_pr_src_addr_unicast_prefix = 2001:db8:0:33::/64
```

m46e_multicast_prefix

M46E の multicast prefix address。

IPv4 マルチキャストパケットをカプセル化する際の prefix アドレスとして使用されます。

IPv6 マルチキャストアドレス形式で[アドレス/prefix 長]のフォーマットで設定してください。

tunnel_mode が通常モード(0)の場合、prefix 長は最大 96 までとなり、M46E-AS モード(1)の場合、prefix 長は最大 80 までとなります。tunnel_mode が M46E-PR モード(2)の場合、本パラメータは使用されません。

このパラメータは省略できません。

設定例）

```
m46e_multicast_prefix = ff01:db8:0:46::/64
```

debug_log

デバッグログを出力するかどうかを指定します。

M46E アプリの動作ログは全て syslog に出力されます。このパラメータは priority が DEBUG レベルのログを出力するかどうかの判定に使用されます。

以下の値が設定可能です。

・yes：出力する

・no　：出力しない（デフォルト）

このパラメータは省略可能です。省略された場合はデバッグログを出力しません。

設定例）

```
debug_log = yes
```

daemon

daemon 化(バックグラウンド動作)するかどうかを指定します。

以下の値が設定可能です。

・yes：daemon 化する（デフォルト）

・no　：daemon 化しない

このパラメータは省略可能です。省略された場合、M46E アプリは daemon 化され、バックグラウンドで動作します。

設定例）

```
daemon = yes
```

startup_script

スタートアップスクリプトのファイルパス。

このパラメータが指定されている場合、指定されたファイルを M46E アプリ起動時に実行します。

詳細は「3.7 章 その他の設定(スタートアップスクリプト)」を参照してください。

※このパラメータには実行権限のあるファイルをフルパスで指定してください。

このパラメータは省略可能です。省略された場合、スクリプトは何も実行されません。

設定例）

```
startup_script = /etc/m46e/m46e_startup.sh
```

route_sync

経路同期を行うかどうかを指定します。

以下の値が設定可能です。

・yes：経路同期を行う

・no ：経路同期を行わない（デフォルト）

このパラメータは省略可能です。省略された場合、M46E アプリは経路同期を行いません。

設定例）

```
route_sync = yes
```

route_entry_max

経路同期で管理する経路情報エントリの最大数。

設定可能な値の範囲は 1～65535 です。

このパラメータは省略可能です。省略された場合、経路情報エントリの最大数は 256 に設定されます。

M46E の経路表に設定されうる経路エントリ数よりも十分大きな値となるように設定してください。

設定例）

```
route_entry_max = 256
```

3.2. Path MTU 管理設定

Path MTU 管理に関連する設定項目です。設定ファイルの[pmtud]セクションに定義されており、パラメータには以下のものがあります。

type

Path MTU 管理機能の動作タイプ。

以下の値が設定可能です。

・0：Path MTU 管理をおこなわない（デフォルト）

・1：M46E アプリ内で最小となる MTU サイズを Path MTU サイズとして保持する。

・2：送信先ホスト毎に Path MTU サイズを個別管理する。

このパラメータは省略可能です。省略された場合、Path MTU 管理はおこなわれません。

設定例）

```
type = 2
```

expire_time

Path MTU サイズ保持時間。

Path MTU サイズを保持する時間を秒単位で設定します。設定可能な値の範囲は 301～65535 です。

このパラメータは省略可能です。省略された場合、保持時間は 600 秒(10 分)に設定されます。

設定例）

```
expire_time = 600
```

3.3. トンネルデバイス設定

カプセル化対象のパケットをルーティングさせる為の仮想トンネルデバイスに関する設定項目です。設定ファイルの[tunnel]セクションに定義されており、パラメータには以下のものがあります。

tunnel_name

M46E アプリが生成するトンネルデバイス名。

カプセル化、デカプセル化対象のパケットのルーティング先となるトンネルデバイスの名称を指定します。Linux のネットワークデバイス名の制限の為、半角英数字で 16 文字以内の値を設定してください。

同一ホスト内で複数の M46E アプリを起動する場合は、他と重複しない値を設定してください。

このパラメータは省略できません。

設定例）

```
tunnel_name = m46e0
```

ipv4_address

VR 用のトンネルデバイスに付与する IPv4 アドレス。

prefix 長も含めた IPv4 アドレス形式で設定してください。

このパラメータは省略可能です。省略した場合はトンネルデバイスに IPv4 アドレスは設定されません。

設定例）

```
ipv4_address = 192.168.0.1/24
```

ipv4_hwaddr

VR 用のトンネルデバイスに設定する MAC アドレス。

このパラメータは省略可能です。省略した場合はトンネルデバイス生成時に OS が自動で割り当てた MAC アドレスがそのまま使用されます。MAC アドレスを固定したいなどの特別な理由が無い限り、このパラメータを設定することは推奨しません。

設定例）

```
ipv4_hwaddr = xx:xx:xx:xx:xx:xx
```

ipv6_address

Backbone ネットワーク用のトンネルデバイスに付与する IPv6 アドレス。

prefix 長も含めた IPv6 アドレス形式で設定してください。

このパラメータは省略可能です。省略した場合はトンネルデバイスに IPv6 アドレスは設定されません。

設定例）

```
ipv6_address = 2001:db8:a:b:c:d::1/64
```

ipv6_hwaddr

Backbone ネットワーク用のトンネルデバイスに設定する MAC アドレス。

このパラメータは省略可能です。省略した場合はトンネルデバイス生成時に OS が自動で割り当てた MAC アドレスがそのまま使用されます。MAC アドレスを固定したいなどの特別な理由が無い限り、

このパラメータを設定することは推奨しません。

設定例）

```
ipv6_hwaddr = xx:xx:xx:xx:xx:xx
```

mtu

Backbone ネットワーク側のトンネルデバイスの MTU サイズ。

設定可能な値の範囲は 1280～65521 です。Stub ネットワーク側のトンネルデバイスは(このパラメータで指定されたサイズ－40(IPv6 ヘッダ長)の値が MTU サイズとして設定されます。

このパラメータは省略可能です。省略した場合は MTU サイズは 1500 に設定されます。

設定例）

```
mtu = 1500
```

ipv4_default_gw

VR 用のトンネルデバイスを VR 内のデフォルトゲートウェイにするかどうかを指定します。

以下の値が設定可能です。

・yes：デフォルトゲートウェイにする

・no ：デフォルトゲートウェイにしない (デフォルト)

このパラメータは省略可能です。省略された場合はトンネルデバイスをデフォルトゲートウェイに設定しません。

設定例）

```
ipv4_default_gw = yes
```

3.4. 物理ネットワークデバイス設定

M46E アプリの VR 内で管理する物理ネットワークデバイスに関する設定項目です。設定ファイルの[device]セクションに定義されています。複数のネットワークデバイスを管理対象とすることも可能で、その場合は、[device]セクションを管理するデバイスの数分、設定ファイルに記述します。[device]セクションのパラメータには以下のものがあります。

physical_dev

VR 内で管理する物理デバイス名。

このパラメータは省略不可です。

設定例）

```
physical_dev = eth0
```

name

管理対象デバイスの VR 内での名前。

ホスト側のデバイス名とは別の名前を使用したい場合に、このパラメータでデバイス名を指定できます。VR 内で eth0 からの連番を振り直したい、などの用途に使用できます。

このパラメータは省略可能です。省略した場合は、physical_dev の値がそのまま VR 内の名前として使用されます。

設定例）

```
name = eth0
```

ipv4_address

このデバイスに付与する IPv4 アドレス。

prefix 長も含めた IPv4 アドレス形式で設定してください。

このパラメータは省略可能です。省略した場合は IPv4 アドレスは設定されません。

設定例）

```
ipv4_address = 192.168.1.1/24
```

ipv4_gateway

デフォルトゲートウェイの IPv4 アドレス。

このデバイスの先に Stub ネットワークのデフォルトゲートウェイとなるルータが存在する場合、このパラメータでそのゲートウェイの IPv4 アドレスを指定します。

デフォルトゲートウェイは[tunnel]セクションの ipv4_default_gw の項目も含めて一つだけ設定するようにしてください。複数のデバイスでデフォルトゲートウェイを設定した場合、パケットが正常にルーティングされなくなります。

このパラメータは省略可能です。省略された場合はデフォルトゲートウェイを設定しません。

設定例）

```
ipv4_gateway = 192.168.1.10
```

mtu

このデバイスに設定する MTU サイズ。

対応する物理デバイスによっては MTU サイズが変更できない場合がある為、特別な理由が無い限り、このパラメータを設定することは推奨しません。MTU サイズを変更したい場合は、M46E アプリ起動前に対応する物理デバイスの MTU サイズを ifconfig などで変更した後に M46E アプリを起動するようにしてください。

このパラメータは省略可能です。省略した場合は、対応する物理デバイスの MTU サイズが設定されます。

設定例）

```
mtu = 1500
```

hwaddr

このデバイスに設定する MAC アドレス。

このパラメータは省略可能です。省略した場合は VR 内にデバイスを移動する時に OS が自動で割り当てた MAC アドレスがそのまま使用されます。MAC アドレスを固定したいなどの特別な理由が無い限り、このパラメータを設定することは推奨しません。

設定例）

```
hwaddr = xx:xx:xx:xx:xx:xx
```

3.5. M46E-AS モード固有設定

M46E-AS モードの動作に関する設定項目です。設定ファイルの[m46e-as]セクションに定義され

ています。 [m46e-as]セクションのパラメータには以下のものがあります。

shared_ipv4_address

M46E-AS の共有する IPv4 アドレス。

このパラメータは省略不可です。

設定例）

```
shared_ipv4_address = 192.168.1.1
```

start_port

M46E-AS の管理するポートの先導番号。0～65535 の範囲で port_num で割切れる数値で設定してください。また、0 以外の数値を設定する場合は port_num よりも大きな値を設定してください。

このパラメータは省略不可です。

設定例）

```
start_port = 1024
```

port_num

M46E-AS の管理するポートの数。

このパラメータは省略不可です。

設定例）

```
port_num = 8
```

3.6. M46E-PR モード固有設定

M46E-PR モードの動作に関する設定項目です。設定ファイルの[m46e-pr]セクションに定義されています。複数の M46E-PR エントリを設定することも可能で、その場合は、[m46e-pr]セクションを管理するデバイスの数分、設定ファイルに記述します。**[m46e-pr]セクションは、ipv4_address と ipv6_address が 1 対となるように設定してください。**[m46e-pr]セクションのパラメータには以下のものがあります。

ipv4_address

M46E-PR が属する IPv4 Network のネットワークアドレス。

prefix 長も含めた IPv4 アドレス形式で設定してください。

このパラメータは省略可能です。

設定例）

```
ipv4_address = 192.168.0.1/24
```

ipv6_address

M46E-PR が属する IPv6 Network のネットワークアドレス。

prefix 長も含めた IPv6 アドレス形式で設定してください。

このパラメータは省略可能です。

設定例）

```
ipv6_address = 2001:db8:a:b:c:d::1/64
```

3.7. その他の設定(スタートアップスクリプト)

共通設定でスタートアップスクリプトを指定した場合、そのスクリプトが M46E アプリの運用開始直前(カプセル化処理開始前)に実行されます。スクリプトの実行は、VR 内部、Backbone 側のネットワーク空間で各々一度ずつ実行されるので、それぞれの空間で実行したい処理を個別に記述できます。

スクリプトは、いくつかの引数を伴って呼び出されます。詳細は下記のスクリプトサンプルを参照してください。

```
#!/bin/bash
################################################################################
# M46E アプリスタートアップスクリプトサンプル
# アプリ起動時にメインループ突入前に下記の引数を伴って呼ばれる。
#
#    ・Plane 名
#        --> 設定ファイルの[device] plane_name の値
#    ・ネットワーク空間(bb | stub)
#        --> どちらのネットワーク空間からこのスクリプトが起動されているか
#    ・トンネルモード
#        --> 設定ファイルの[device] tunnel_mode の値
#    ・unicast plane prefix
#        --> 設定ファイルの[device] m46e_unicast_prefix と plane_id を連結させた値
#    ・multicast plane prefix
#        --> 設定ファイルの[device] m46e_multicast_prefix と plane_id を連結させた値
#    ・トンネルデバイス名
#        --> 設定ファイルの[tunnel] tunnel_name の値
#
################################################################################

# 引数の展開
plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
    # VR 内部の処理
    # Stub ネットワークの追加の経路などを記述する
    ip route add 192.168.20.0/24 via 192.168.1.1 dev eth0
fi

if [ $nw_type = "bb" ]
then
    # Backbone 側の処理
    # Backbone ネットワークの追加の経路などを記述する
    # 以下の例は上の VR 内部の処理で追加した経路宛のパケットを
    # VR に引き込む為の経路を追加する例
    ip -6 route add ${unicast_prefix}192.168.20.0/120 dev ${tunnel_name}
fi

exit 0
```

4. アプリケーションの起動と終了

4.1. アプリケーションの起動

M46E アプリの起動はコマンドラインから下記のコマンドを実行することでおこないます。

※コマンドの実行は root 権限を持ったユーザでおこなってください。

```
#m46eapp -f [設定ファイル]
```

4.1.1. アプリケーション起動時の経路設定

M46E アプリは、アプリケーション起動時に設定ファイルの内容を元に、VR 内の Routing Table と Backbone 側の Routing Table に対して経路の自動設定をおこないます。

自動設定される経路情報は以下の通りです。

1. VR 内の経路表
   - [tunnel]セクションの ipv4_address の prefix に対応する IPv4 の経路
   - [device]セクションの ipv4_address の prefix に対応する IPv4 の経路
   - デフォルトゲートウェイの IPv4 の経路
     ([tunnel]セクションの ipv4_default_gw が yes に設定されているか、[device]セクションの ipv4_gateway が設定されている場合のみ)

2. Backbone 側の経路表
   - [tunnel]セクションの ipv4_address を M46E アドレスフォーマットに変換した IPv6 アドレスに対するホストルート(/128 の経路)。
   - 1.で追加した[device]セクションの ipv4_address に対応する経路を M46E アドレスフォーマットに変換した IPv6 アドレスに対する経路(prefix は IPv4 の prefix＋96)。

例えば、設定ファイルが以下のような設定になっていた場合、

```
[general]
unicast_prefix = 2001:db8:0:46::/64
plane_id = 64:1

[tunnel]
name =  m46e0
ipv4_address = 192.168.0.1/24
ipv4_gateway = yes

[device]
physical_name = eth0
name =  eth0
ipv4_address = 192.168.1.1/24
```

M46E アプリ起動後、VR、Backbone 各々の経路表は以下のようになります。

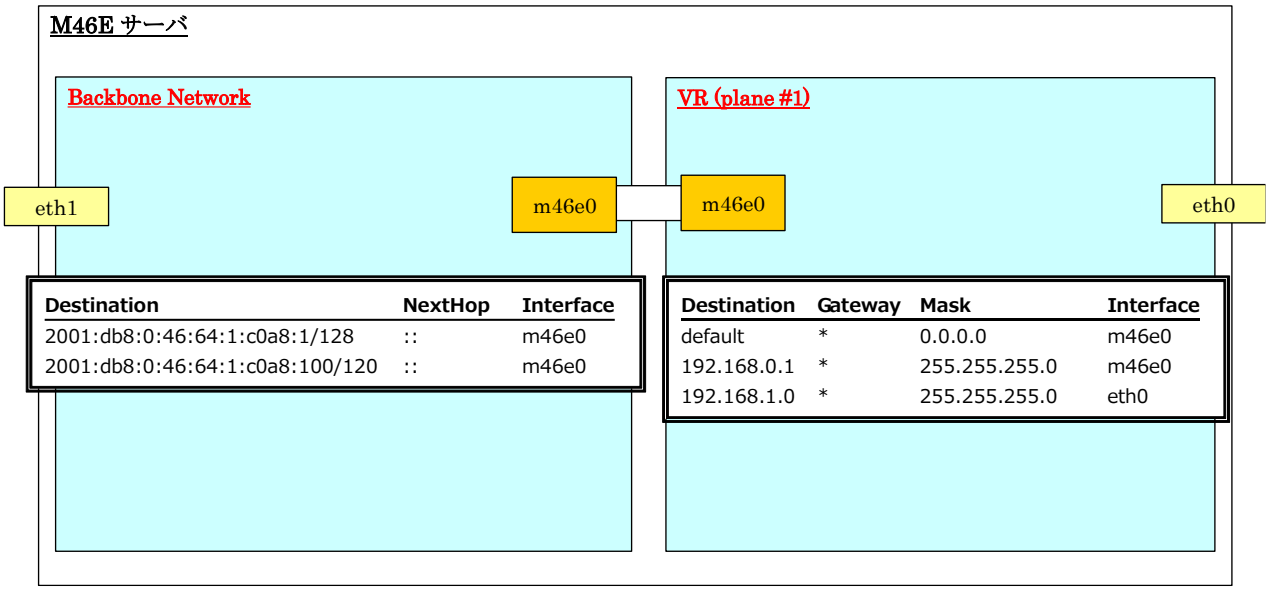

図 1　M46E アプリ起動後の経路表

4.2. アプリケーションの終了

後述の「5.21 アプリケーション終了」で説明するシャットダウン用のコマンドを使用します。一般的なコマンドと同様に Ctrl-C 押下(daemon 化していない場合)や kill コマンドによる強制終了も可能ですが、予期せぬ動作をする可能性がある為、推奨はしません。

4.3. アプリケーションの設定変更

本アプリケーションは、アプリ起動中の動的な設定変更に対応します。後述の「5.22 アプリケーション再起動」説明する再起動用のコマンドを使用します。設定ファイルの値を変更した後にアプリケーションを再起動してください。

5. サポートツール

サポートツールとして、起動中の M46E アプリの状態を表示したり、VR の経路表の変更や IP アドレスの変更をおこなう為のシェル接続機能を提供する外部コマンドを用意しています。

コマンドの書式は以下の通りです。

```
Usage: m46ectl -n PLANE_NAME COMMAND OPTIONS
       m46ectl { -h | --help | --usage }

where COMMAND := { exec shell  |  exec inet   |  show stat   |  show conf   |
                   show pmtu   |  set debug   |  set ffrag   |  set defgw   |
                   set pmtumd  |  set pmtutm  |  set tunmtu  |  set devmtu  |
                   add device  |  del device  |  add pr      |  del pr      |
                   enable pr   |  disable pr  |  show pr     |  load pr     |
                   shutdown    |  restart }

  exec shell     : 指定した M46E アプリの VR に接続されたシェルを起動します。
  exec inet      : 指定した M46E アプリの Stub Network でコマンドを実行します。
  show stat      : 指定した M46E アプリの統計情報を表示します。
  show conf      : 指定した M46E アプリの設定情報を表示します。
  show pmtu      : 指定した M46E アプリの Path MTU 管理テーブルの情報を表示します。
  set debug      : 指定した M46E アプリのデバッグ情報出力を設定します。
  set ffrag      : 指定した M46E アプリの強制フラグメントモードを設定します。
  set defgw      : 指定した M46E アプリのデフォルトゲートウェイをトンネルデバイスに設定します。
  set pmtumd     : 指定した M46E アプリの Path MTU Discovery モードを設定します。
  set pmtutm     : 指定した M46E アプリの Path MTU Discovery のタイマ値を設定します。
  set tunmtu     : 指定した M46E アプリのトンネルデバイスの MTU 値を設定します。
  set devmtu     : 指定した M46E アプリの Stub Network のデバイスの MTU 値を設定します。
  add device     : 指定した M46E アプリの収容デバイスの増設を行います。
  del device     : 指定した M46E アプリの収容デバイスの減設を行います。
 *add pr         : 指定した M46E アプリの PR エントリテーブルに PR エントリを追加します。
 *del pr         : 指定した M46E アプリの PR エントリテーブルから PR エントリを削除します。
 *enable pr      : 指定した M46E アプリの PR エントリテーブルの PR エントリを有効にします。
 *disable pr     : 指定した M46E アプリの PR エントリテーブルの PR エントリを無効にします。
 *show pr        : 指定した M46E アプリの PR エントリテーブルを表示します。
 *load pr        : 指定した M46E アプリの PR テーブルを一括設定します。
  shutdown       : 指定した M46E アプリを終了します。
  restart        : 指定した M46E アプリを再起動します。

  *印で示されるコマンドは、M46E アプリの動作モードが、M46E-PR モードである場合のみ有効。
```

引数の-n で指定する PLANE_NAME には、設定ファイルの[general]セクションの plane_name で設定した M46E アプリ識別名を入れてください。

サポートコマンドの実行は、接続しようとしている M46E アプリを起動したユーザと同じユーザ(通常は root ユーザ)でおこなってください。M46E アプリの実行ユーザと異なるユーザでサポートコマンドを実行した場合、M46E アプリへの接続が拒否されます。

以下に、各コマンドの詳細を説明します。

5.1. シェル起動

M46E アプリの管理する VR 内で動作するシェル以下(VR シェル)を起動します。

VR シェルでは一般的なシェルコマンドが全て使用できますので、VR 内部の経路表の変更(route コマンド)や、IP アドレスの追加、削除(ifconfig コマンド)などがおこなえます。

コマンドを実行すると、以下のような表示と共に、VR シェルが起動します。

```
#m46ectl -n m46e0 exec shell

Type "exit" or [ Ctrl-a q ] to exit the shell

#
```

コマンドを終了する場合は、exit と入力してシェル自体を終了させるか、

```
#exit
```

または、Ctrl-a 押下後、続けて q を入力してください。

```
Ctrl-a q
```

5.2. Stub Network 側コマンド実行

M46E アプリの管理する VR 内で引数で指定されたコマンドを実行します。コマンドは、現状、10 種類(ifconfig / ip / arp / ethtool / iptables / ip6tables /ipmaddr /route /ps /netstat)のみ指定することができます。

```
m46ectl -n m46e0 inet  'ifconfig -a'
```

5.3. 統計情報の表示

M46E アプリでは VR で送受信したパケット数やカプセリング処理をおこなったパケット数などの統計情報を収集しています。収集している情報は以下の表の通りです。統計情報の表示は、M46E アプリの実行モードにより異なります。

収集項目	内容	M46E アプリ実行モード		
		M46E	M46E-AS	M46E-PR
[M46E]				
packet count				
total receive count	総受信パケット数	○	○	○
total send count	総送信パケット数	○	○	○
total drop count	総破棄パケット数	○	○	○
total error count	総エラー数	○	○	○
[TUNNEL IPV4]				
packet count				

receive count				
unicast(forward)	IPv4 ユニキャストパケット受信数	○	○	○
multicast(forward)	IPv4 マルチキャストパケット受信数	○	○	○
broadcast(drop)	IPv4 ブロードキャストパケット受信数	○	○	○
not IPv4 protocol(drop)	非 IPv4 パケット受信数	○	○	○
multicast(drop)	IPv4 マルチキャストパケット受信数	－	－	○
link local multicast(drop)	IPv4 リンクローカルマルチキャストパケット受信数	○	○	－
destination unknown(drop)	宛先 M46E-PR 不明の破棄パケット数	－	－	○
fragment on M46E-AS(drop)	フラグメント不可の破棄パケット数	－	○	－
not TCP/UDP on M46E-AS(drop)	TCP/UDP 以外の破棄パケット数	－	○	－
send count				
send success	IPv4 パケット送信数	○	○	○
send error	IPv4 パケット送信エラー数	○	○	○
send(fragment) success	IPv4 フラグメントパケット送信数	○	○	○
send(fragment) error	IPv4 フラグメントパケット送信エラー数	○	○	○
[TUNNEL IPV6]				
packet count				
receive count				
encap unicast(forward)	IPv4 ユニキャストパケット受信数	○	○	○
encap multicast(forward)	IPv4 マルチキャストパケット受信数	○	○	－
broadcast(drop)	IPv4 ブロードキャストパケット受信数	○	○	○
not IPv6 protocol(drop)	非 IPv6 パケット受信数	○	○	○
ttl over(drop)	TTL=1 パケット受信数	○	○	○
link local multicast(drop)	IPv4 リンクローカルマルチキャストパケット受信数	○	○	－
invalid next header(drop)	IPv6 next header が不正なパケット受信数	○	○	○
send count				
send success	IPv6 パケット送信数	○	○	○
send error	IPv6 パケット送信エラー数	○	○	○
[ICMP]				
packet count				
receive count				
IPv6 icmp packet too big	ICMPv6 (Packet too Big) 受信数	○	○	○
send count				
IPv4 icmp fragment needed				
send success	fragment needed 送信数	○	○	○

send error	fragment needed 送信エラー数	○	○	○

コマンドを入力すると、以下のようにその時点での統計情報が表示されます。以下は、M46E アプリの動作モードが M46E の場合の例です。

```
#m46ectl -n m46e0 show stat

【M46E】

  packet count
    total recieve count             : 7618284
    total send count                : 7618284
    total drop count                : 0
    total error count               : 0

【TUNNEL IPV4】

  packet count
    recieve count                   : 7618266
      unicast(forward)              : 7618266
      multicast(forward)            : 0
      broadcast(drop)               : 0
      not IPv4 protocol(drop)       : 0
      link local multicast(drop)    : 0
    send count                      : 7618266
      send success                  : 7618266
      send error                    : 0
      send(fragment) success        : 0
      send(fragment) error          : 0

【TUNNEL IPV6】

  packet count
    recieve count                   : 18
      encap unicast(forward)        : 18
      encap multicast(forward)      : 0
      broadcast(drop)               : 0
      not IPv6 protocol(drop)       : 0
      ttl over(drop)                : 0
      link local multicast(drop)    : 0
      invalid next header(drop)     : 0
    send count                      : 18
      send success                  : 18
      send error                    : 0

【ICMP】

  packet count
    recieve count
      IPv6 icmp packet too big      : 0
    send count
      IPv4 icmp fragment needed     : 0
        send success                : 0
        send error                  : 0
```

5.4. 設定情報の表示

M46E アプリ動作中の設定情報を表示します。前述の VR シェルでネットワークデバイスの IP アドレスや MAC アドレスなどを変更した場合には、現在の設定値と異なる値が表示されることがあります。

コマンドを入力すると、以下のようにアプリ起動時に読込んだ設定ファイルの内容が表示されます。以下は、M46E アプリの動作モードが M46E の場合の例です。

```
#m46ectl -n m46e0 show conf

Config file name = /home/qnet/Step4/m46e.conf

[general]
plane_name = m46e1
plane_id = 64:1
tunnel_mode = 0
m46e_unicast_prefix = 2001:db8:0:46::/64
m46e_pr_src_addr_unicast_prefix = 2001:db8:0:ff0a::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
debug_log = no
daemon = yes
startup_script = /home/qnet/TrainA-ph7/hyouka/m46e_3to4.sh
force_fragment = no

[m46e-as]
shared_ipv4_address = 192.168.1.1
start_port = 1024
port_num = 8

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e1
ipv4_hwaddr = 42:83:b0:a4:ad:b8
ipv6_hwaddr = 32:f3:a9:c6:37:26
mtu = 1500
ipv4_default_gw = yes

[device]
type = macvlan
macvlan_mode = private
name = eth1
physical_dev = eth1
ipv4_address = 192.168.10.3/24
mtu = 1500
hwaddr = aa:aa:aa:aa:aa:aa
```

5.5. Path MTU 管理テーブルの表示

Path MTU 管理テーブルの内容を表示します。PathMTU 管理テーブルで管理している情報には

1. 送信先ホストの IPv6 アドレス
2. 1.のアドレスに対する MTU サイズ(Packet Too Big 受信時に通知された MTU サイズ)
3. 保持期間

の 3 項目があります。

コマンドを入力すると、以下のようにアプリ起動時に読込んだ設定ファイルの内容が表示されます。default はパケット送信先の IPv6 アドレスが他のどれにも一致しない場合に適応される MTU サイズで、M46E アプリ起動時に、Backbone 側のトンネルデバイスの MTU サイズが設定されます。

```
#m46ectl -n m46e0 show pmtu

                    Dst Addr                    | Path MTU | remain time
------------------------------------------------+----------+------------
default                                         |      1500|          -1
2001:db8:0:46:0:1:c0a8:c815                     |      1400|         530
2001:db8:0:46:0:1:c0a8:c816                     |      1400|         564
```

5.6. デバッグログ出力モード設定

デバッグログの出力モードを設定します。出力モードは on/off の 2 種類あります。通常、出力モードが on であっても十分な量のデバッグ情報は出力されません。デバッグに十分な量のデバッグ情報が出力されるためには、M46E アプリがデバッグ情報出力有りでコンパイルされている必要があります。

```
#m46ectl -n m46e0 debug on
```

5.7. 強制フラグメントモード設定

強制フラグメントのモードを設定します。強制フラグメントモードは on/off の 2 種類あります。強制フラグメントモードが on の場合、M46E アプリは DF ビットが ON のパケットもフラグメント対象として動作します。

```
#m46ectl -n m46e0 ffrag on
```

5.8. デフォルトゲートウェイ設定

Stub Network のデフォルトゲートウェイをトンネルデバイスに設定/解除します。on を指定した場合に設定し、off を指定した場合に解除されます。本コマンドによる設定よりも前にデフォルトゲートウェイが設定されている場合は本コマンドは実行されません。

```
#m46ectl -n m46e0 defgw on
```

5.9. Path MTU Discovery モード設定

Path MTU Discovery の動作モードを設定します。動作モードは、(0 : 動作なし、1 : トンネル毎、2 : ホスト毎) の指定が可能です。Path MTU Discovery モードを変更した場合、変更前の Path MTU Discovery テーブルの情報はクリアされます。

```
#m46ectl -n m46e0 pmtumd 2
```

5.10. Path MTU Discovery タイマ設定

Path MTU Discovery のエントリ設定のタイマ値を設定します。Path MTU Discovery タイマ値は、（301～65535）の範囲で指定可能です。変更した Path MTU Discovery タイマ値は、新規に Path MTU Discovery エントリの生成より有効となります。

```
#m46ectl -n m46e0 pmtutm 600
```

5.11. トンネルデバイス MTU 設定

IPv6 側トンネルデバイスの MTU 設定します。MTU は、(1280～65521) の範囲で指定可能です。また、IPv4 側のトンネルデバイスの MTU は、IPv6 側トンネルデバイスの MTU－40 に自動設定されます。

```
#m46ectl -n m46e0 tunmtu 1400
```

5.12. 収容デバイス MTU 設定

収容デバイスの MTU 設定します。MTU は、（548～65521）の範囲で指定可能です。対応する物理デバイスの MTU サイズより大きな値を指定した場合、MTU サイズが変更できない場合があります。

```
#m46ectl -n m46e0 devmtu 1380
```

5.13. 増設

収容デバイスを増設します。コマンドオプションの詳細は「3.4 物理ネットワークデバイス設定」を参照してください。

```
#m46ectl -n m46e0 add device physical_dev=eth1 name= eth1
ipv4_address=192.168.10.0/24 ipv4_gateway= 192.168.10.1 mtu=1400
hwaddr=aa:aa:aa:aa:aa:aa
```

5.14. 減設

収容デバイスを減設します。コマンドオプションの詳細は「3.4 物理ネットワークデバイス設定」を参照してください。

```
#m46ectl -n m46e0 del device physical_dev=eth1
```

5.15.M46E-PR エントリ追加

M46E-PR エントリテーブルに M46E-PR エントリを追加します。本コマンドは M46E アプリの動作モードが M46E-PR の場合のみ有効です。コマンドオプションの enable は省略可能です。省略された場合、M46E-PR エントリテーブルに disable なエントリとして追加されます。

```
#m46ectl -n m46e0 add pr 192.168.10.0/24 2001:db8::/64 enable
```

5.16.M46E-PR エントリ削除

M46E-PR エントリテーブルから M46E-PR エントリを削除します。本コマンドは M46E アプリの動作モードが M46E-PR の場合のみ有効です。

```
#m46ectl -n m46e0 del pr 192.168.10.0/24
```

5.17.M46E-PR エントリ活性化

M46E-PR エントリテーブルに登録されている M46E-PR エントリを活性化します。活性化された M46E-PR エントリは M46E-PR のカプセル化エントリとして有効となります。本コマンドは M46E アプリの動作モードが M46E-PR の場合のみ有効です。

```
#m46ectl -n m46e0 enable pr 192.168.10.0/24
```

5.18.M46E-PR エントリ非活性化

M46E-PR エントリテーブルに登録されている M46E-PR エントリを非活性化します。非活性化された M46E-PR エントリは M46E-PR のカプセル化エントリとして無効となります。本コマンドは M46E アプリの動作モードが M46E-PR の場合のみ有効です。

```
#m46ectl -n m46e0 disable pr 192.168.10.0/24
```

5.19.M46E-PR エントリ表示

M46E-PR エントリテーブルに登録されている M46E-PR エントリを表示します。本コマンドは M46E アプリの動作モードが M46E-PR の場合のみ有効です。

```
#m46ectl -n m46e0 show pr
```

コマンドを入力すると、以下のように M46E-PR エントリテーブルの内容が表示されます。「*」は活性化された M46E-PR エントリであることを意味しています。

```
#m46ectl -n m46e0 show pmtu

+-------------------------------------------------------------------------+
/     M46E Prefix Resolution Table                                        /
+---+----------+--------------------+---------+-------------------------+
    | Plane ID | IPv4 Network Address | Netmask | M46E-PR Address Prefix |
+---+----------+--------------------+---------+-------------------------+
  * | 64:1     | 192.175.80.0       | /120    | 2001:db8:0:ff46::       |
  * | 64:1     | 192.169.20.0       | /120    | 2001:db8:0:ff46::       |
    | 64:1     | 192.174.70.0       | /120    | 2001:db8:0:ff46::       |
    | 64:1     | 192.168.10.0       | /120    | 2001:db8:0:ff0a::       |
+---+----------+--------------------+---------+-------------------------+
 Note : [*] shows available entry for prefix resolution process.
```

5.20.M46E-PR 一括設定

M46E-PR エントリ設定ファイルを読込み、M46E-PR エントリテーブルに対する設定を一括で行います。本コマンドは M46E アプリの動作モードが M46E-PR の場合のみ有効です。

```
#m46ectl -n m46e0 load pr pr_entry_list.txt
```

M46E-PR エントリ設定ファイルは以下のフォーマットで記述します。

```
# M46E-PR table entry setting list
add pr 192.168.51.0/24 2001:db8:1::/64 enable          (追加：enable エントリ)
add pr 192.168.52.0/24 2001:db8:2::/64 disable         (追加：diable エントリ)
del pr 192.168.53.0/24                                 (削除)
enable pr 192.168.54.0/24                              (活性化)
disble pr 192.168.55.0/24                              (非活性化)
```

5.21.アプリケーション終了

指定した M46E アプリケーションを終了します。コマンド実行後は、即時にプロンプトが返ってきますので、正常にアプリが終了していることを ps コマンドなどで確認してください。

```
#m46ectl -n m46e0 shutdown
```

5.22.アプリケーション再起動

指定した M46E アプリケーションを再起動します。正常にアプリが再起動していることを ps コマンドなどで確認してください。

```
#m46ectl -n m46e0 restart
```

6. 設定例

この章では、具体的なネットワーク構成を例として、M46E を導入する際の設定ファイルの記述例やスタートアップスクリプトの書き方などを説明します。

6.1. Plane ID 無し

Plane ID 無しの M46E ネットワーク構成です。主に企業や大学などの閉域網における拠点間 VPN 接続を想定しています。同一企業内での VPN となるので、Plane ID は必要ありません。

下記のようなネットワーク構成を例として説明します。

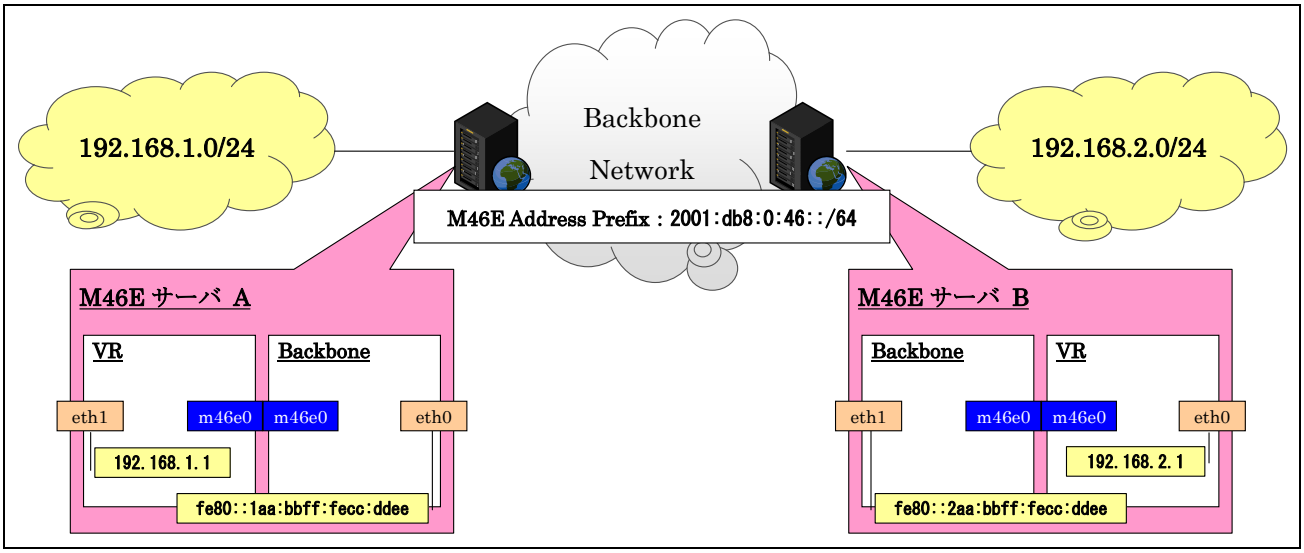

図 2　Plane ID 無しの M46E ネットワーク構成

設定のポイントを以下に箇条書きします。

- Plane ID 無しなので、plane_id の項目は記述しない。
- VR の経路にデフォルトゲートウェイは設定しない。
- 対向拠点のサブネット宛の経路はスタートアップスクリプトを使用して明示的に設定する。
- Path MTU 管理は送信先ホスト毎におこなう。
- ネットワークデバイスの MAC アドレス、MTU サイズはデフォルトのままする。

サーバ A, サーバ B の設定ファイルとスタートアップスクリプトの内容は以下のようになります。

M46E サーバ A の設定

設定ファイル

```
[general]
plane_name = m46e0
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e0_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e0
ipv4_default_gw = no

[device]
name = eth1
physical_dev = eth1
ipv4_address = 192.168.1.1/24
```

スタートアップスクリプト

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 192.168.2.0/24 dev ${tunnel_name}
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}192.168.2.0/120 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth0
fi

exit 0
```

M46E サーバ B の設定

設定ファイル

```
[general]
plane_name = m46e0
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e0_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e0
ipv4_default_gw = no

[device]
name = eth0
physical_dev = eth0
ipv4_address = 192.168.2.1/24
```

スタートアップスクリプト

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 192.168.1.0/24 dev ${tunnel_name}
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}192.168.1.0/120 ¥
    via fe80::1aa:bbff:fecc:ddee dev eth1
fi

exit 0
```

6.2. Plane ID 有り（ISP VPN 接続サービス）

Plane ID 有りの M46E ネットワーク構成です。主に ISP（Internet Service Provider)事業者が顧客に対して VPN 接続サービスを提供する場合を想定しています。顧客毎に Plane ID を割り振ってネットワーク空間を分離することにより、同じ IP アドレスを持ったプライベートネットワークを収容することが可能となります。

下記のようなネットワーク構成を例として説明します。

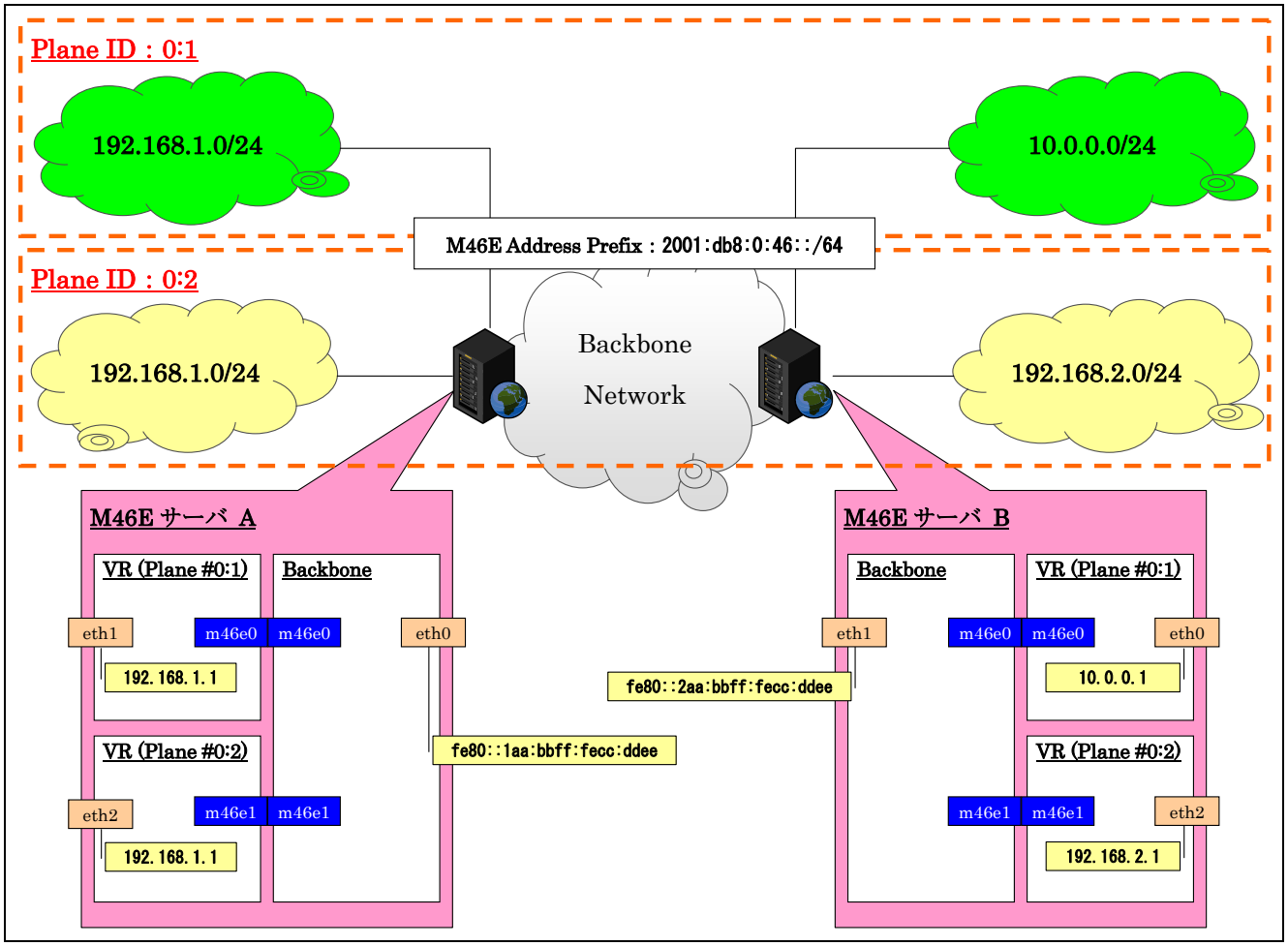

図 3　Plane ID 有り(ISP VPN 接続サービス)のネットワーク構成図

設定ファイルは基本的に Plane ID 無しのパターンと同じです。違いは

- plane_id の項目に各 Plane ID の値を設定する。
- 複数 Plane を収容するので、設定ファイル、スタートアップスクリプトを Plane の数分用意する。

の 2 点となります。

ネットワーク構成図を見てもわかるように、Plane ID が異なっていれば、同一ホスト内で同じ IPv4 アドレスを持つことが可能です。

サーバ A, サーバ B の設定ファイルとスタートアップスクリプトの内容は以下のようになります。

M46E サーバ A の設定

設定ファイル (Plane #0:1)

```
[general]
plane_name = m46e0
plane_id = 0:1
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e0_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e0
ipv4_default_gw = no

[device]
name = eth1
physical_dev = eth1
ipv4_address = 192.168.1.1/24
```

スタートアップスクリプト (Plane #0:1)

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 10.0.0.0/24 dev ${tunnel_name}
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}10.0.0.0/120 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth0
fi

exit 0
```

M46E サーバ B の設定

設定ファイル (Plane #0:1)

```
[general]
plane_name = m46e0
plane_id = 0:1
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e0_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e0
ipv4_default_gw = no

[device]
name = eth0
physical_dev = eth0
ipv4_address = 10.0.0.1/24
```

スタートアップスクリプト (Plane #0:1)

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 192.168.1.0/24 dev ${tunnel_name}
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}192.168.1.0/120 ¥
    via fe80::1aa:bbff:fecc:ddee dev eth1
fi

exit 0
```

M46E サーバ A の設定（続き）

設定ファイル（Plane #0:2)

```
[general]
plane_name = m46e1
plane_id = 0:2
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e1_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e1
ipv4_default_gw = no

[device]
name = eth2
physical_dev = eth2
ipv4_address = 192.168.1.1/24
```

スタートアップスクリプト(Plane #0:2)

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 192.168.2.0/24 dev ${tunnel_name}
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}192.168.2.0/120 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth0
fi

exit 0
```

M46E サーバ B の設定（続き）

設定ファイル(Plane #0:2)

```
[general]
plane_name = m46e1
plane_id = 0:2
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e1_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e1
ipv4_default_gw = no

[device]
name = eth2
physical_dev = eth2
ipv4_address = 192.168.2.1/24
```

スタートアップスクリプト(Plane #0:2)

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 192.168.1.0/24 dev ${tunnel_name}
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}192.168.1.0/120 ¥
    via fe80::1aa:bbff:fecc:ddee dev eth1
fi

exit 0
```

6.3. Plane ID 有り（Global ネットワーク接続）

M46E を経由して Global ネットワーク(インターネット網)に接続するパターンです。

下記のようなネットワーク構成を例として説明します。

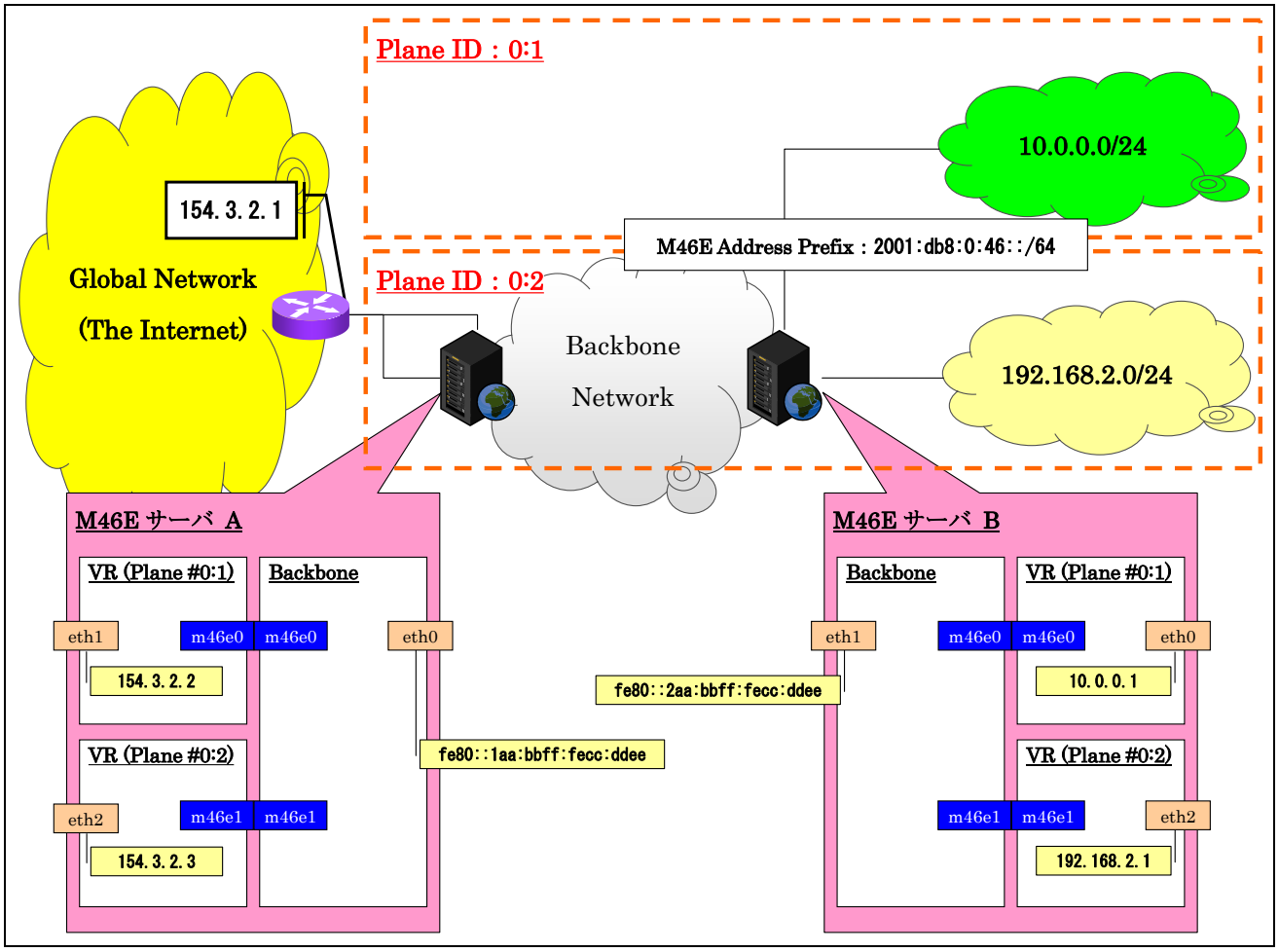

図 4 Plane ID 有り(Global ネットワーク接続)のネットワーク構成図

設定のポイントは

- サーバ A の VR のデフォルトゲートウェイは物理デバイスの先にあるルータ(154.3.2.1)に設定する(Global ネットワークの接続先となっている為)
- サーバ A の Backbone 経路表に M46E prefix address/96 宛のパケットをトンネルデバイスに転送する経路を設定する。
- サーバ B の VR のデフォルトゲートウェイをトンネルデバイスに設定する。
- サーバ A では Global ネットワークへのパケット転送時に NAT 機能を利用する為、iptable コマンドを使用して IP マスカレードの設定を追加する。

となります。

サーバ A, サーバ B の設定ファイルとスタートアップスクリプトの内容は以下のようになります。

M46E サーバ A の設定

設定ファイル (Plane #0:1)

```
[general]
plane_name = m46e0
plane_id = 0:1
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e0_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e0
ipv4_default_gw = no

[device]
name = eth1
physical_dev = eth1
ipv4_address = 154.3.2.2/24
ipv4_gateway = 154.3.2.1
```

スタートアップスクリプト (Plane #0:1)

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 10.0.0.0/24 dev ${tunnel_name}
  iptables -t nat ¥
    -A POSTROUTING -o eth1 -j MASQUERADE
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}10.0.0.0/120 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth0
  ip -6 route add ¥
    ${unicast_prefix}/96 dev ${tunnel_name}
fi

exit 0
```

M46E サーバ B の設定

設定ファイル (Plane #0:1)

```
[general]
plane_name = m46e0
plane_id = 0:1
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e0_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e0
ipv4_default_gw = yes

[device]
name = eth0
physical_dev = eth0
ipv4_address = 10.0.0.1/24
```

スタートアップスクリプト (Plane #0:1)

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}/96 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth1
fi

exit 0
```

M46E サーバ A の設定（続き）

設定ファイル（Plane #0:2）

```
[general]
plane_name = m46e1
plane_id = 0:2
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e1_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e1
ipv4_default_gw = no

[device]
name = eth2
physical_dev = eth2
ipv4_address = 154.3.2.3/24
ipv4_gateway = 154.3.2.1
```

スタートアップスクリプト（Plane #0:2）

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "stub" ]
then
  ip route add 192.168.2.0/24 dev ${tunnel_name}
  iptables -t nat ¥
    -A POSTROUTING -o eth2 -j MASQUERADE
fi

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}192.168.2.0/120 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth0
  ip -6 route add ¥
    ${unicast_prefix}/96 dev ${tunnel_name}
fi

exit 0
```

M46E サーバ B の設定（続き）

設定ファイル（Plane #0:2）

```
[general]
plane_name = m46e1
plane_id = 0:2
m46e_unicast_prefix   = 2001:db8:0:46::/64
m46e_multicast_prefix = ff01:db8:0:46::/64
tunnel_mode = 0
startup_script = /etc/m46e/m46e1_startup.sh

[pmtud]
type = 2
expire_time = 600

[tunnel]
tunnel_name = m46e1
ipv4_default_gw = yes

[device]
name = eth2
physical_dev = eth2
ipv4_address = 192.168.2.1/24
```

スタートアップスクリプト（Plane #0:2）

```
#!/bin/bash

plane_name=$1
nw_type=$2
tunnel_mode=$3
unicast_prefix=$4
multicast_prefix=$5
tunnel_name=$6

if [ $nw_type = "bb" ]
then
  ip -6 route add ¥
    ${unicast_prefix}/96 ¥
    via fe80::2aa:bbff:fecc:ddee dev eth1
fi

exit 0
```

7. トラブルシューティング

この章では、M46E が正常に動作しないなどのトラブルが発生した場合の解決方法について説明します。

7.1. M46E アプリ起動に関するトラブル

M46E アプリ起動に関するトラブルシューティングを以下に示します。

事象	M46E アプリが起動できない。
原因	複数の原因が考えられます。 1. root 権限が無いユーザで M46E アプリを実行している。 2. 設定ファイルの設定が適切でない。 3. plane 名が重複する M46E アプリが既に起動されている。
対策	1. スーパーユーザで M46E アプリを実行してください。 2. 設定ファイルの設定を確認してください。設定ファイルの設定が適切でない場合は、/var/log/messages に設定の不備が出力されます。 3. ps コマンドなどで他の M46E アプリが起動されていないか確認してください。他の M46E アプリが起動されている場合は、当該 M46E アプリの設定ファイルを確認し、plane 名が重複しないように設定してください。

7.2. M46E アプリ実行中の動作に関するトラブル

M46E アプリ実行中の動作に関するトラブルシューティングを以下に示します。

事象	M46E にてカプセリングが行われない。
原因	複数の原因が考えられます。 1. 隣接ルーターから M46E に IPv4 パケットがフォワーディングされていない。 2. 当該 plane の IPv4 経路表に宛先となる経路が設定されていない。 3. IPv6 経路表に宛先となる経路が設定されていない。 4. VMware にて仮想マシンのネットワークアダプタに“e1000”が使用されている。 5. 当該 plane の M46E アプリがシャットダウンされている可能性がある。
対策	1. 隣接ルータの IPv4 経路設定が正しいか確認してください。 2. m46ectl コマンドにて当該 plane の shell を起動し、当該 plane の IPv4 経路表に宛先となる経路が設定されているか確認してください。宛先となる経路が存在しない場合は、ip コマンドなどで宛先となる経路を設定ファイル内で指定されるトンネルデバイスに対してルーティングする経路を設定してください。 （設定例）# ip route add 192.0.0.0/24 dev m46e1 3. ip コマンドなどにて IPv6 経路表に宛先となる経路が設定されているか確認してください。宛先となる経路が存在しない場合は、ip コマンドなどで宛先となる経路を設定してください。

	(設定例) # ip -6 route add 5a46::1:c0:0/120 via fe80::226:2dff:fe0b:b243 dev eth1 4. 仮想マシンのネットワークアダプタに“e1000”が使用されている場合は、ifconfig コマンドなどで当該ネットワークアダプタに対応する仮想インタフェースの promiscus モードを有効に設定してください。 (設定例) # ifconfig eth2 promisc 5. ps コマンドなどでにて当該 plane の M46E アプリが起動されていることを確認してください。M46E アプリが起動されていない場合は、起動してください。

事象	M46E にてデカプセリングが行われない。
原因	複数の原因が考えられます。 1. 隣接ルーターから M46E に IPv6 パケットがフォワーディングされていない。 2. IPv6 経路表に受信する経路が設定されていない。 3. 当該 plane の IPv4 経路表に宛先となる経路が設定されていない。 4. VMware にて仮想マシンのネットワークアダプタに“e1000”が使用されている。 5. 当該 plane の M46E アプリがシャットダウンされている可能性がある。 6. 送信側の M46E の設定と受信側の M46E の設定が一致していない。
対策	1. 隣接ルータの IPv6 経路設定が正しいか確認してください。 2. ip コマンドなどにて IPv6 経路表に受信する経路が設定されているか確認してください。受信する経路が設定されていない場合は、設定ファイルの不備が考えられますので、設定ファイルに誤りが無いか確認してください。 3. m46ectl コマンドにて当該 plane の shell を起動し、当該 plane の IPv4 経路表に宛先となる経路が設定されているか確認してください。宛先となる経路が設定されていない場合は、設定ファイルの不備が考えられますので、設定ファイルに誤りが無いか確認してください。 4. 仮想マシンのネットワークアダプタに“e1000”が使用されている場合は、ifconfigコマンドで当該ネットワークアダプタに対応する仮想インタフェースの promiscus mode を有効に設定してください。 (設定例) # ifconfig eth2 promisc 5. ps コマンドなどでにて当該 plane の M46E アプリが起動されていることを確認してください。M46E アプリが起動されていない場合は、起動してください。 6. M46E prefix や plane_id などの設定値は、送信側 M46E と受信側 M46E で一致させる必要があります。両装置の設定ファイルに誤りがないか確認してください。

付録 1. 設定ファイルサンプル

```
################################################################################
# M46E アプリケーション 設定ファイル サンプル
################################################################################

################################################################################
# 共通設定 (省略不可)
################################################################################
[general]
################################################################################
# Plane 名。Stub ネットワークの名称として使用する。(省略不可)
# 外部コマンドで指定するアプリケーション名はここで設定した値を使用するので、
# 複数 Plane を起動する場合は他と被らない値にすること。
plane_name              = m46e0
################################################################################
# PlaneID。IPv6 表記で設定すること。(省略可)
# 省略時のデフォルト値：0:0
plane_id                = 64:1
################################################################################
# M46E の unicast prefix address。(省略不可)
# IPv6 ユニキャストアドレス形式で設定すること。
# tunnel_mode が通常モード(0)の場合は prefix 長は最大 96 まで、
# M46E-AS モード(1)の場合は最大 80 までとなる。
m46e_unicast_prefix   = 2001:db8:0:46::/64
################################################################################
# M46E-PR の送信元アドレスの unicast prefix address。(省略不可)
# IPv6 ユニキャストアドレス形式で設定すること。
# tunnel_mode に関係なく prefix 長は最大 96 まで。
m46e_pr_src_addr_unicast_prefix   = 2001:db8:0:33::/64
################################################################################
# M46E の multicast prefix address。(省略不可)
# IPv6 マルチキャストアドレス形式で設定すること。
# tunnel_mode が通常モード(0)の場合は prefix 長は最大 96 まで、
# M46E-AS モード(1)の場合は最大 80 までとなる。
m46e_multicast_prefix = ff01:db8:0:46::/64
################################################################################
# M46E の動作モード。以下が設定可能 (省略不可)
#  ・0：通常モード
#  ・1：M46E-AS モード
#  ・2：M46E-PR モード
tunnel_mode             = 0
################################################################################
# デバッグログを出力するかどうか (省略可)
#    yes：出力する
#    no  ：出力しない (デフォルト)
debug_log               = no
################################################################################
# daemon 化(バックグラウンド動作)するかどうか (省略可)
#    yes：daemon 化する (デフォルト)
#    no  ：daemon 化しない
daemon                  = yes
################################################################################
# スタートアップスクリプト (省略可)
# Backbone,Stub 各々のネットワーク空間で、メインループ突入前に実行される。
# スクリプトの引数には、Plane 名、ネットワーク空間(bb | stub)、トンネルモード、
```

```
# unicast plane prefix、multicast plane prefix、トンネルデバイス名が
# この順序で渡される。
#
# 【スクリプト実行例】
#  - Backbone 側
#    startup.sh m46e0 bb 0 2001:db8:0:46:64:1:: ff01:db8:0:46:64:1:: m46e0
#  - Stub 側
#    startup.sh m46e0 stub 0 2001:db8:0:46:64:1:: ff01:db8:0:46:64:1:: m46e0
#
# ※スクリプトファイルは実行権限のあるファイルをフルパスで指定すること。
startup_script         = /etc/m46e/m46e_startup.sh
################################################################################
# 経路同期をするかどうか（省略可）
# IPv4 網の経路表に変更があった場合に IPv6 網側の経路表を更新する。
# IPv6 網の経路表に変更があった場合に IPv4 網側の経路表を更新する。
#   yes：経路同期を行う
#   no ：経路同期を行わない（デフォルト）
route_sync  = no
################################################################################
# 経路同期用の経路表に登録できるエントリの最大数（省略可）
# 設定可能範囲：1～65535
# 省略時のデフォルト値：256
route_entry_max = 256

################################################################################
# M46E-AS モード 専用の設定
# 動作モードが M46E-AS(1)以外の場合は Don't Care(省略可)
#
# M46E-AS モードで動作する場合に IPv6 側に設定する経路情報を
# 本セクションの情報を基に設定する。
# (例)
# ◎[tunnel]ipv4_default_gw=no の場合(The Internet 側に配置される場合)
#   Backbone ネットワーク側の経路情報には
#     ([generic]に設定されている prefix)::0:0:0/80  dev (トンネルデバイス)
#   が設定される。
#   さらに、Stub ネットワーク側の経路表に
#     (共有 IP アドレス)/32  dev (トンネルデバイス)
#   の経路が設定される。
#
# ◎[tunnel]ipv4_default_gw=yes の場合(データサーバ側に配置される場合)
#     address     = 192.168.1.1
#     start_port  = 1024 (0x400)
#     port_num    = 8
#   と設定されていた場合、Backbone ネットワーク側の経路情報には
#     ([generic]に設定されている prefix):c0a8:101:400 / 125
#                                        ~~~~~~~~ ~~~   ~~~
#   が設定される。
#                                      address  port  port_num
#                                                     (8=2^3 なので 128 から 3 を引く)
################################################################################
[m46e-as]
################################################################################
# 共有する IP アドレス（省略不可）
shared_ipv4_address = 192.168.1.1
################################################################################
# 管理するポートの先頭番号(Location が 1 の場合のみ有効)（省略不可）
# ※0～65535 の範囲で port_num で割り切れる数値で設定すること。
# ※0 以外の数値を設定する場合は port_num よりも大きい値を設定すること。
start_port          = 1024
################################################################################
```

```
# 管理するポートの数(Location が 1 の場合のみ有効) (省略不可)
# ※2 の乗数で設定すること
port_num            = 8


################################################################################
# IPv6 網側の Path MTU Discovery 関連設定 (省略可)
################################################################################
[pmtud]
################################################################################
# PMTU Discovery 動作タイプ。以下が設定可能。(省略可)
# ・0 : PMTU Discovery 処理を行わない (デフォルト)
# ・1 : トンネル毎の情報保持
# ・2 : ホスト毎の情報保持
type        = 2
################################################################################
# PMTU 長保持期限(秒) (省略可)
# 設定可能範囲 : 301～65535
# 省略時のデフォルト値 : 600
expire_time = 600


################################################################################
# トンネルデバイス設定 (省略不可)
################################################################################
[tunnel]
################################################################################
# 生成するトンネルデバイス名 (省略不可)
# ※半角英数字で 16 文字まで設定可能とする。
tunnel_name     = m46e0
################################################################################
# Stub ネットワーク用トンネルデバイスに付与する IPv4 アドレス (省略可)
ipv4_address    = 192.168.0.1/24
################################################################################
# Stub ネットワーク用トンネルデバイスに設定する MAC アドレス (省略可)
# 省略時のデフォルト値 : OS が自動設定した値
ipv4_hwaddr     = xx:xx:xx:xx:xx:xx
################################################################################
# Backbone ネットワーク用トンネルデバイスに付与する IPv6 アドレス (省略可)
ipv6_address    = 2001:db8:a:b:c:d::1/64
################################################################################
# Backbone ネットワーク用トンネルデバイスに設定する MAC アドレス (省略可)
# 省略時のデフォルト値 : OS が自動設定した値
ipv6_hwaddr     = xx:xx:xx:xx:xx:xx
################################################################################
# トンネルデバイスの Backbone ネットワーク側の MTU サイズ (省略可)
# 設定可能範囲 : 1280～65521(※)
# ※ 出口となる物理デバイスの MTU 長よりも小さな値を設定した場合、
#    不要なフラグメントを行ったり、正常に通信できなくなったりするので注意すること。
# (Stub ネットワーク側のトンネルデバイスはこのサイズ-40(IPv6 ヘッダ長)に設定される)
# 省略時のデフォルト値 : 1500
mtu             = 1500
################################################################################
# Stub ネットワーク用のトンネルデバイスをデフォルトゲートウェイにするかどうか。(省略可)
#   yes : デフォルトゲートウェイにする
#   no  : デフォルトゲートウェイにしない (デフォルト)
# ※動作モードが M46E-AS モード(1)の場合には、ホストの配置位置の判定にも
#   使用される。詳細は[M46E-AS]セクションのコメント参照。
ipv4_default_gw = yes
```

```
#################################################################################
# デバイス設定（省略可）
#################################################################################
[device]
#################################################################################
# Stub ネットワーク内でのデバイス名（省略可）
# 省略時のデフォルト値：対応する物理デバイスのデバイス名
name            = eth0
#################################################################################
# 対応する物理デバイス名（省略不可）
physical_dev    = eth0
#################################################################################
# デバイスに付与する IPv4 アドレス（省略可）
ipv4_address    = 192.168.1.1/24
#################################################################################
# デフォルトゲートウェイのアドレス（省略可）
# ※デフォルトゲートウェイはトンネルデバイスを含めた全てのデバイスで
#   一つだけ設定すること。
ipv4_gateway    = 192.168.1.10
#################################################################################
# デバイスに設定する MTU サイズ（省略可）
# 省略時のデフォルト値：対応する物理デバイスの MTU サイズ
# ※対応する物理デバイスの MTU サイズより大きな値を設定した場合、
#   MTU サイズが変更できない場合がある。
#   その場合、元の MTU サイズのままで起動する。
mtu             = 1500
#################################################################################
# デバイスに設定する MAC アドレス（省略可）
# 省略時のデフォルト値：OS が自動設定した値
# ※ハードウェア(デバイスドライバ)の制限により、
#   MAC アドレスが変更できない場合がある。
#   その場合、元の MAC アドレスのままで起動する。
hwaddr          = xx:xx:xx:xx:xx:xx

# 複数のデバイスを指定する場合には、 [device]セクションを
# 登録するデバイスの数分繰り返し設定する。
# ※デバイス名が重複しないように注意すること。
[device]
name            = eth1
physical_dev    = eth1
ipv4_address    = 192.168.2.1/24
mtu             = 1500
hwaddr          = xx:xx:xx:xx:xx:xx

# 最低限、必要な設定は physical_dev だけなので、
# 下記の設定のみでアプリ起動は可能。
[device]
physical_dev    = eth2

#################################################################################
# M46E-PR モード 専用の設定
# 動作モードが M46E-PR(2)以外の場合は Don't Care(省略可)
#
# M46E-PR モードで動作する場合に参照する送信先 M46E アドレスを
# 本セクションの情報を基に M46E-PR テーブルに設定する。
# ※省略時はコマンド登録すること。
#
#################################################################################
```

```
# 自拠点の M46E-PR エントリー登録（省略可）
[m46e-pr]
################################################################################
# 自拠点の M46E-PR エントリー(IPv4 ネットワークアドレス)（省略可）
ipv4_address     = 192.168.100.0/25
################################################################################
# 自拠点の M46E-PR エントリー(M46E-PR Prefix)（省略可）
m46e_pr_prefix = 2001:db8:0:33::/64
################################################################################
# 他拠点の M46E-PR エントリー登録（省略可）
[m46e-pr]
################################################################################
# 他拠点の M46E-PR エントリー(IPv4 ネットワークアドレス)（省略可）
ipv4_address     = 192.165.0.0/16
################################################################################
# 他拠点の M46E-PR エントリー(M46E-PR Prefix)（省略可）
m46e_pr_prefix = 2001:db8:0:77::/64
################################################################################

# 複数の他拠点の M46E-PR エントリーを登録する場合は、[m46e-pr]セクションを
# 設定する M46E-PR エントリーの数分繰り返し設定する。
# ※組合せが同一のエントリーを登録しないように注意すること。
#   （ネットマスクが異なれば OK）
```

付録 2. M46E とは

2.1. 概要

M46E(Stateless Automatic IPv4 over IPv6 Tunneling)とは、IPv4 パケットを IPv6 パケットにカプセル化して伝送するトンネル技術の一つです。以下のような特徴があります。

- IPv4 アドレスをそのまま IPv6 アドレス空間にマッピングし、カプセリングします。
- さらに、IPv4 ネットワークを区別する為の識別情報(IPv4 network plane ID)を IPv6 アドレスに追加することによって、IPv4 プライベートアドレスの利用も可能になり、数十億の IPv4 ネットワークを、IPv6 ネットワーク上に収容することが可能となります。

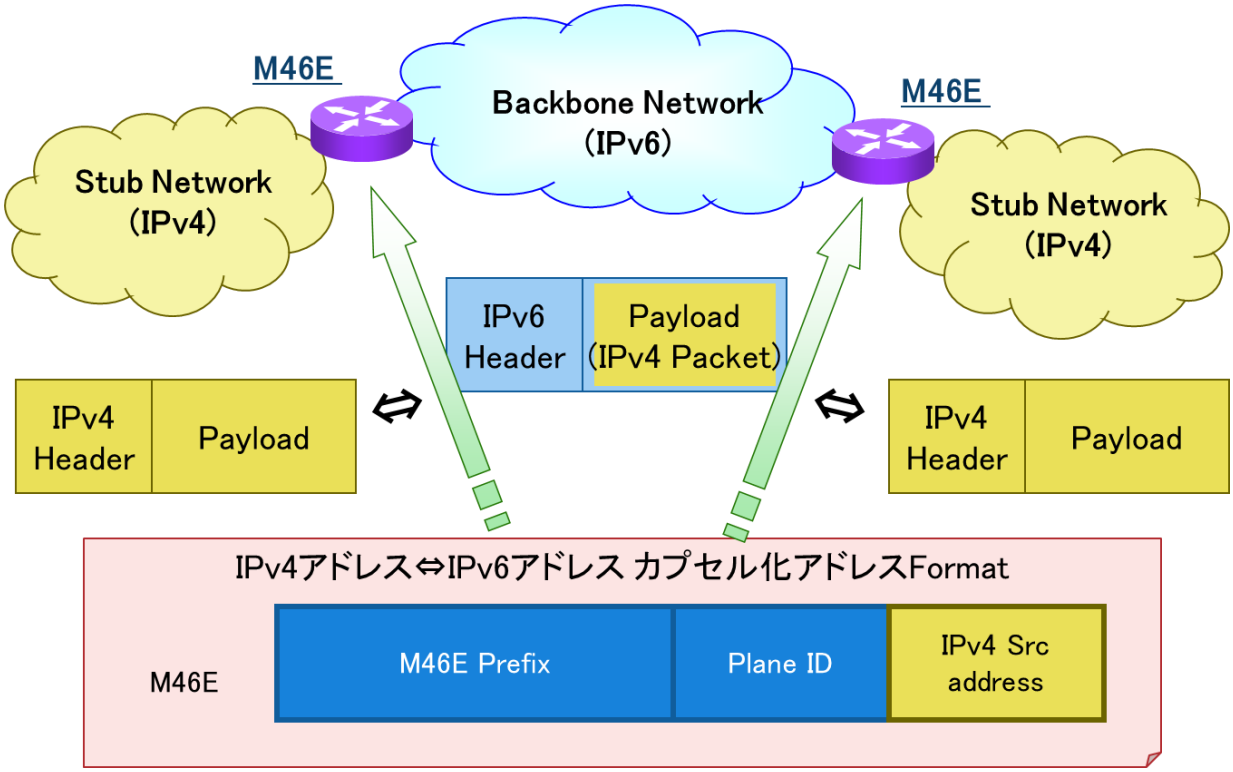

図 5　M46E 概念図

2.2. 機能

本アプリケーションでは、M46E アーキテクチャとして、以下の機能を実装しています。

- IPv4 パケットのカプセル化、および IPv6 パケットのデカプセル化をおこなうカプセリング機能。
- 複数のネットワーク plane を単一ホスト内で独立して管理する VR(Virtual Router)機能。
- バックボーンとなる IPv6 網の MTU サイズに応じてカプセル化前の IPv4 パケットを分割したり、送信元に対して ICMP エラー(Fragment Needed)を返信する Path MTU 管理機能。

2.2.1. カプセリング機能

Stub ネットワークから受信した IPv4 パケットを IPv6 パケットにカプセル化して Backbone ネットワークに転送し、逆に Backbone ネットワークから受信した IPv6 パケットをデカプセル化して Stub ネットワークに転送する機能です。

本アプリケーションでは、カプセリング機能を Linux カーネルの TAN/TUP デバイス(以下、トンネルデバイス)で実現しています。アプリケーション起動時に 1 組のトンネルデバイスペアを生成し、片方を後述する VR 内でのカプセル化用デバイス、もう片方を Backbone ネットワークでのデカプセル化用デバイスとして扱います。

VR 内でトンネルデバイスに転送された IPv4 パケットは、カプセル化の処理を施された後に IPv6 パケットとして Backbone ネットワーク側のトンネルデバイスに転送され、カーネルのルーティング機能によって Backbone ネットワークに送信され、Backbone ネットワーク側のトンネルデバイスに転送された IPv6 パケットはデカプセル化の処理を施された後に VR 内のトンネルデバイスを経由して Stub ネットワークへ送信されます。

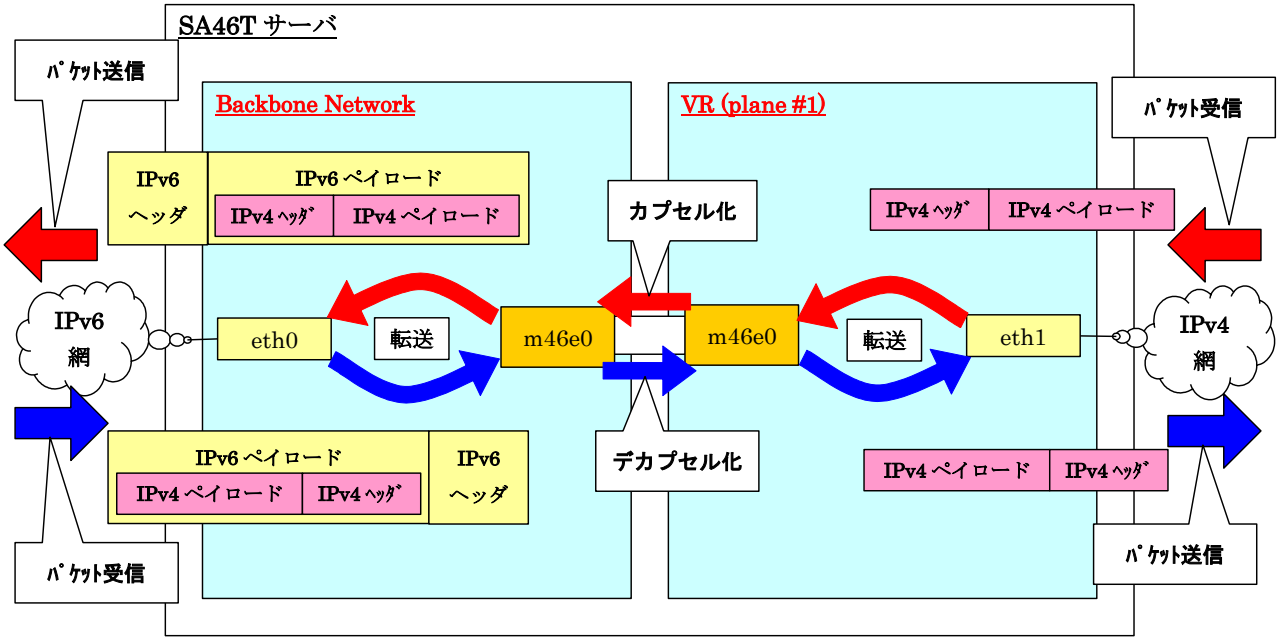

図 6　カプセリング処理

2.2.2. VR(Virtual Router)機能

Plane 毎に独立した Routing Table と管理対象となるネットワークデバイスを定義することによって、Plane ID の異なるネットワークを 1 台の M46E サーバに収容することが可能になります。

VR が異なれば、同じ IPv4 アドレスでも異なる経路として扱われるので、1 台の M46E サーバ内で同じ IP アドレス、同じ経路を設定することが可能になります。

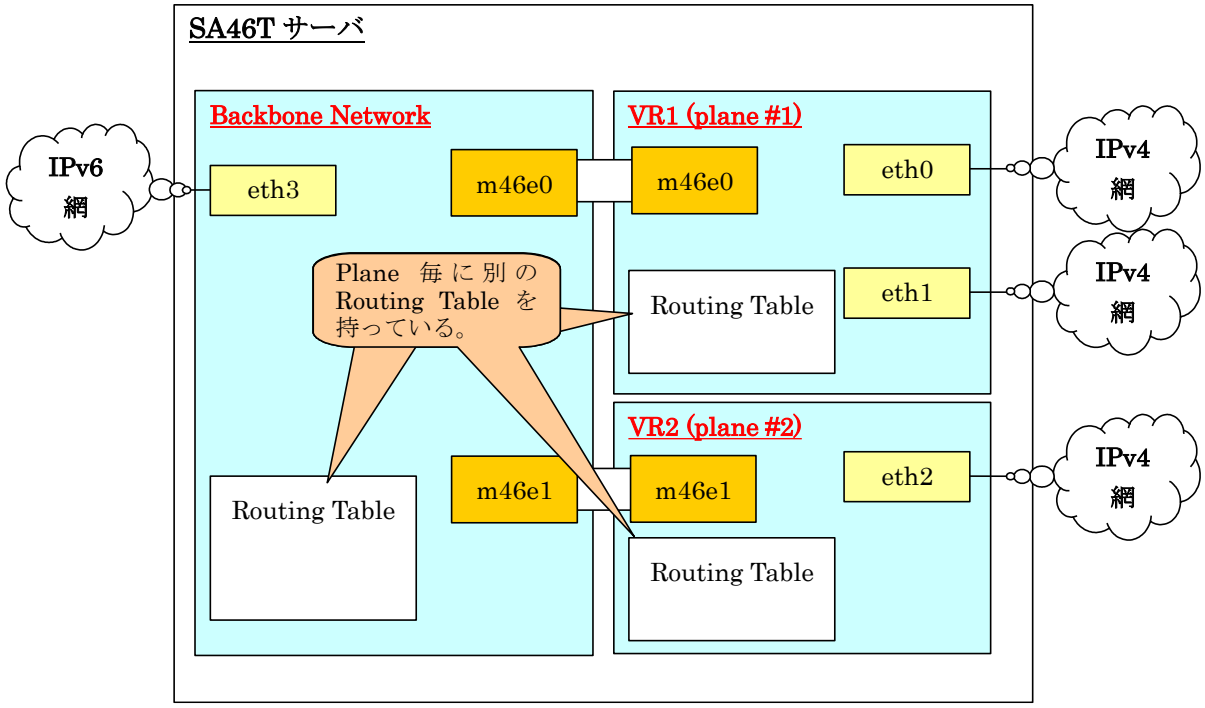

図 7　Virtual Router 機能

2.2.3 Path MTU 管理機能

Backbone ネットワークである IPv6 網から ICMPv6 エラー(Packet Too Big)を受信した際に、Path MTU サイズを保持しておき、以降のカプセリング処理で Path MTU サイズを考慮したフラグメント処理をおこなう機能。カプセリング処理によって IPv6 ヘッダ(40byte)分のパケットサイズ増加が発生するので、それに伴うパケット不通を回避する為の機能です。

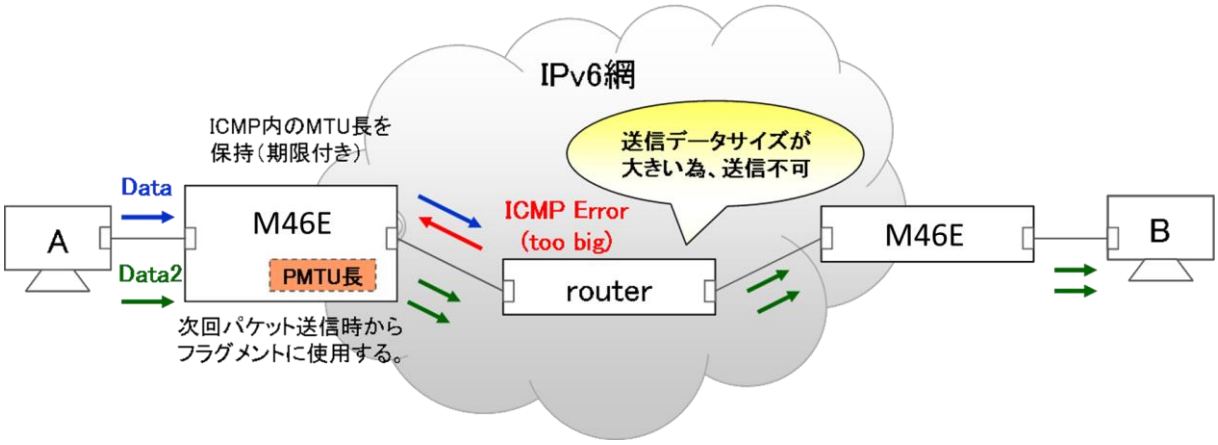

図 8　Path MTU 管理機能