

# ME6E アプリ操作説明書

---

# 目次

1. はじめに .....	1
2. 動作環境 .....	1
2.1. OS .....	1
2.2. ファイル構成 .....	1
3. アプリケーションの設定 .....	1
3.1. 共通設定 .....	1
3.2. カプセリング機能 設定 .....	3
3.3. 代理 ARP 機能 設定 .....	6
3.4. 代理 NDP 機能 設定 .....	7
3.5. Stub Network MAC アドレス管理機能 設定 .....	8
3.6. Prefix Resolution 機能 設定 .....	9
3.7. その他の設定(スタートアップスクリプト) .....	9
4. アプリケーションの起動と終了 .....	11
4.1. アプリケーションの起動 .....	11
4.2. アプリケーションの終了 .....	11
4.3. アプリケーションの設定変更 .....	11
5. サポートツール .....	12
5.1. 統計情報の表示 .....	13
5.2. 設定情報の表示 .....	14
5.3. 代理 ARP テーブルの表示 .....	16
5.4. 代理 NDP テーブルの表示 .....	16
5.5. Prefix Resolution テーブルの表示 .....	17
5.6. 代理 ARP エントリの追加 .....	17
5.7. 代理 NDP エントリの追加 .....	17
5.8. Prefix Resolution エントリの追加 .....	18
5.9. 代理 ARP エントリの削除 .....	18
5.10. 代理 NDP エントリの削除 .....	18
5.11. Prefix Resolution エントリの削除 .....	19
5.12. Prefix Resolution エントリの一括削除 .....	19
5.13. Prefix Resolution エントリの有効化 .....	20
5.14. Prefix Resolution エントリの無効化 .....	20
5.15. Prefix Resolution エントリの一括設定 .....	20
5.16. アプリケーション終了 .....	21
6. 設定例 .....	22
6.1. Plane ID 無し:Stub Network が IPv4:静的エントリ設定 .....	23
6.2. Plane ID 無し:Stub Network が IPv4:動的エントリ設定(ユニキャスト) .....	25
6.3. Plane ID 無し:Stub Network が IPv6:静的エントリ設定 .....	29

6.4. Plane ID 有り:Stub Network が IPv4:静的エントリ設定 .....	31
6.5. Plane ID 有り:Stub Network が IPv4:動的エントリ設定(マルチキャスト) .....	35
付録 1. 設定ファイルサンプル .....	39

## 1. はじめに

本ドキュメントでは、ME6E アプリケーション(以下、ME6E 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

ME6E アプリは bridge 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
me6eapp	ME6E アプリ 実行ファイル
me6ectl	サポートコマンド 実行ファイル
me6e.conf	設定ファイル。ファイル名は任意。
me6e_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

ME6E 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[common]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 mode**

ME6E 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 : ME6E-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
```

### ***plane\_name***

ME6E アプリで管理する Plane の名前。

ME6E 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 ME6E 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 = me6e0
```

### ***debug\_log***

デバッグログを出力するかどうかを指定します。

ME6E 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、ME6E 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 ME6E 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/me6e/me6e_startup.sh
```

### 3.2. カプセリング機能 設定

カプセリング機能に関連する設定項目です。設定ファイルの[capsuling]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。

#### me6e unicast prefix

ME6E の unicast prefix address。

ユニキャストの Ether フレーム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。

prefix 長は最大 80 までとなります。マルチキャスト転送機能を有効にする場合は、prefix 長は 64 以内に設定してください。

このパラメータは省略できません。

設定例)

```
me6e_unicast_prefix = 2001:db8:0::/48
```

#### me6e multicast prefix

ME6E の multicast prefix address。

マルチキャストの Ether フレーム、及びブロードキャストの Ether フレーム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。ME6E multicast prefix は、RFC3306 準拠で設定してください。

prefix 長は 96 固定となります。

このパラメータは省略できません。

設定例)

```
me6e_multicast_prefix = ff38:0030:2001:db8:0::
```

#### plane id

ME6E 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 = 0:1
```

#### hop limit

ME6E マルチキャストアドレスにてカプセル化転送する ME6E マルチキャストパケットに設定される hop\_limit 数。

このパラメータは省略可能です。省略した場合は 64 として扱われます。

設定例)

```
hop_limit = 64
```

#### ***backbone\_physical\_dev***

Backbone Network に接続する物理インタフェース名。

このパラメータは省略できません。

設定例)

```
backbone_physical_dev = eth1
```

#### ***stub\_physical\_dev***

Stub Network に接続する物理インタフェース名。

このパラメータは省略できません。

設定例)

```
stub_physical_dev = eth2
```

#### ***tunnel\_name***

ME6E アプリが生成するトンネルデバイス名。

Stub Network 側のパケット送受信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 15 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 ME6E 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 = me6etun0
```

#### ***tunnel\_mtu***

トンネルデバイスの MTU サイズ。

設定可能な値の範囲は 1280~65535 です。

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)

```
tunnel_mtu = 1500
```

#### ***tunnel\_hwaddr***

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 OS により自動設定されます。

設定例)

```
tunnel_hwaddr = 22:33:44:55:11:22
```

#### ***bridge\_name***

ME6E アプリが生成するブリッジデバイス名。

Linux のネットワークデバイス名の制限の為、半角英数字で 15 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 ME6E 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)

```
bridge_name = me6ebr0
```

### **bridge\_hwaddr**

ブリッジ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 OS により自動設定されます。

設定例)

```
bridge_hwaddr = aa:bb:cc:dd:ee:ff
```

### **l2multi\_l3uni**

マルチキャストの Ether フレーム、及びブロードキャストの Ether フレームを ME6E ホストアドレスでカプセル化して送信するかどうかを指定します。

以下の値が設定可能です。

- yes : 動作する
- no : 動作しない (デフォルト)

本パラメータは Backbone Network が IPv6 Multicast Routing に未対応の場合に指定してください。

このパラメータは省略可能です。省略した場合、本機能は動作しません。

設定例)

```
l2multi_l3uni = no
```

### **me6e\_host\_address**

ME6E のホストアドレス。

同一の Plane に属する ME6E のホストアドレスを指定します。

本パラメータは Backbone Network が IPv6 Multicast Routing に未対応の場合に指定してください。

このパラメータは、**l2multi\_l3uni** が “yes” に設定されている場合は省略できません。**l2multi\_l3uni** が省略されている、或いは “no” に設定されている場合は省略可能です。

設定例)

```
me6e_host_address = 2001:db8:0:46::1
```

### **me6e\_pr\_nicast\_prefix**

ME6E-PR の unicast prefix address。

動作モードが 1 (=ME6E-PR) の場合に、ユニキャストの Ether フレームをカプセル化する際の source 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。

prefix 長は最大 80 までとなります。

このパラメータは、動作モードに 1 (=ME6E-PR) が設定されている場合には省略できません。

動作モードが 0 (=通常モード) に設定されている場合は省略可能です。

設定例)



**me6e\_pr\_unicast\_prefix = 2001:db8:ffe6::/48**

### 3.3. 代理 ARP 機能 設定

代理 ARP 機能に関連する設定項目です。設定ファイルの[proxy\_arp]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。

#### **arp\_enable**

代理 ARP 機能の動作を指定します。

以下の値が設定可能です。

- yes : 動作する (デフォルト)
- no : 動作しない

このパラメータは省略可能です。省略された場合、代理 ARP 機能は動作します。

設定例)

**arp\_enable = yes**

#### **arp\_entry\_update**

ARP テーブルへの動的エントリ設定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 : 動作する (デフォルト)
- no : 動作しない

このパラメータは省略可能です。省略された場合、ARP テーブルへの動的エントリ設定が動作します。

設定例)

**arp\_entry\_update = yes**

#### **arp\_aging\_time**

ARP エントリ保持時間。

ARP エントリ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)

**arp\_aging\_time = 600**

#### **arp\_entry\_max**

ARP エントリの登録最大数。

ARP エントリの登録最大数を設定します。設定可能な値の範囲は 1～65535 です。

このパラメータは省略できません。

設定例)

**arp\_entry\_max = 128**

また、続けて IP アドレス = MAC アドレス の形式で ARP エントリを指定します。複数の ARP エ

ントリの指定が可能で、その場合はエントリ数分を続けて列挙します。

設定例)

**192.168.0.10 = 00:1D:73:E6:BE:10**

**192.168.0.11 = 00:1D:73:E6:BE:11**

### 3.4. 代理 NDP 機能 設定

代理 NDP 機能に関連する設定項目です。設定ファイルの[proxy\_ndp]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。

#### **ndp enable**

代理 ndp 機能の動作を指定します。

以下の値が設定可能です。

- yes : 動作する (デフォルト)
- no : 動作しない

このパラメータは省略可能です。省略された場合、代理 NDP 機能は動作します。

設定例)

**ndp\_enable = yes**

#### **ndp entry update**

NDP テーブルへの動的エントリ設定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 : 動作する (デフォルト)
- no : 動作しない

このパラメータは省略可能です。省略された場合、NDP テーブルへの動的エントリ設定が動作します。

設定例)

**ndp\_entry\_update = yes**

#### **ndp aging time**

NDP エントリ保持時間。

NDP エントリ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)

**ndp\_aging\_time = 600**

#### **ndp entry max**

NDP エントリの登録最大数。

NDP エントリの登録最大数を設定します。設定可能な値の範囲は 1~65535 です。

このパラメータは省略できません。

設定例)

**ndp\_entry\_max = 128**

また、続けて IP アドレス = MAC アドレス の形式で NDP エントリを指定します。複数の NDP エントリの指定が可能で、その場合はエントリ数分を続けて列挙します。

設定例)

**fec0::10 = 00:1D:73:E6:BE:10**

**fec0::11 = 00:1D:73:E6:BE:11**

### 3.5. Stub Network MAC アドレス管理機能 設定

Stub Network MAC アドレス管理機能に関連する設定項目です。設定ファイルの[mng\_macaddr]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。

#### **mng\_macaddr\_enable**

Stub Network MAC アドレス管理機能の動作を指定します。

以下の値が設定可能です。

- yes : 動作する (デフォルト)
- no : 動作しない

このパラメータは省略可能です。省略された場合、Stub Network MAC アドレス管理機能は動作します。

設定例)

**mng\_macaddr\_enable = yes**

#### **mac\_entry\_update**

Stub Network 内ホストの ME6E アドレスの動的設定を動作させるかどうかを指定します。

以下の値が設定可能です。

- yes : 動作する (デフォルト)
- no : 動作しない

このパラメータは省略可能です。省略された場合、Stub Network 内ホストの ME6E アドレスの動的設定が動作します。

設定例)

**mac\_entry\_update = yes**

#### **mac\_valid\_lifetime**

Stub Network 内ホストの ME6E アドレスの有効時間。

ME6E アドレスの有効と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)

**mac\_valid\_lifetime = 600**

#### **mac\_entry\_max**

Stub Network 内ホストの ME6E アドレスのエントリの登録最大数。

Stub Network 内ホストの ME6E アドレスの登録最大数を設定します。設定可能な値の範囲は 1～65535 です。

このパラメータは省略できません。

設定例)

```
mac_entry_max = 128
```

また、続けて、`hosthw_addr` = MAC アドレス の形式で Stub Network 内ホストの MAC アドレスを指定します。複数のホストの MAC アドレスの指定が可能で、その場合はホスト数分、続けて列挙します。

設定例)

```
hosthw_addr = 00:44:55:66:77:11
```

```
hosthw_addr = 00:44:55:66:77:12
```

### 3.6. Prefix Resolution 機能 設定

Prefix Resolution 機能に関連する設定項目です。設定ファイルの[me6e\_pr]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。また、複数の ME6E-PR エントリを設定することも可能で、その場合は、[me6e\_pr]セクションを管理するエントリ数分、設定ファイルに記述します。

[me6e\_pr]セクションは、`macaddr` と `pr_prefix` が 1 対となるように設定してください。

#### ***macaddr***

ME6E-PR 配下の Host の MAC アドレス。

ME6E-PR 配下の Host の MAC アドレスを MAC アドレス形式で設定してください。

このパラメータは、動作モードに 1 (=ME6E-PR) が設定されている場合には省略できません。

動作モードが 0 (=通常モード) に設定されている場合には省略可能です。

設定例)

```
macaddr = B0:99:28:1F:59:4E
```

#### ***pr\_prefix***

ME6E-PR の prefix address。

ME6E-PR が属する IPv6 ネットワークのネットワークアドレスを 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。

このパラメータは、動作モードに 1 (=ME6E-PR) が設定されている場合には省略できません。

動作モードが 0 (=通常モード) に設定されている場合には省略可能です。

設定例)

```
pr_prefix = 2001:db8:ff57:73::/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 ME6E 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。

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
#####
#ME6E 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呼ばれる。
#####

# 以下の例は対向拠点の Stub Network 内の MAC アドレス “00:00:c0:a8:01:15” である host 宛
# のパケットを送信するための経路を追加する例

# ・MAC アドレス “00:00:c0:a8:01:15” を持つホスト宛の経路
# 2001:db8:0:ff31::c0a8:0115/128
# ・ME6E の Backbone 側の近隣ルータアドレス
# fe80::2aa:bbff:fecc:ddee
# ・Backbone 側の物理インタフェース名
# eth0

ip -6 route add ¥
2001:db8:0:ff31::c0a8:0115/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth0

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

ME6E 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。

```
#me6eapp -f [設定ファイル]
```

### 4.2. アプリケーションの終了

後述の「5.16 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)による終了も可能です。また、kill コマンドによる強制終了も可能ですが、ME6E アプリが正常に終了されないため推奨はしません。

kill -9 プロセス番号 でME6E アプリを強制終了した場合、ME6E アプリ起動時に生成されたブリッジデバイスが削除されません。この状態では ME6E アプリの再起動に失敗しますので、再起動する前にブリッジデバイスの削除をおこなってください。

※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。

#### — ブリッジデバイス削除手順 —

- 1) ブリッジデバイスのインタフェースダウン

```
#ifconfig ブリッジデバイス名 down
```

- 2) ブリッジデバイスの削除

```
#brctl delbr ブリッジデバイス名
```

### 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ていません。起動中のアプリケーションの PlaneID や、ME6E Prefix などの値を変更したい場合は、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 ME6E アプリの状態を表示したり、代理 ARP テーブルや代理 NDP テーブルの表示、エントリの設定をおこなう為の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: me6ectl -n plane_name COMMAND
      me6ectl { -h | --help | --usage }
where COMMAND := { show stat
                    show conf
                    show arp
                    show ndp
                    show pr
                    add arp IPv4Addr MACAddr
                    add ndp IPv6Addr MACAddr
                    add pr MACAddr Ipv6Addr/Prefixlen mode
                    del arp IPv4Addr
                    del ndp IPv6Addr
                    del pr MACAddr
                    enable pr MACAddr
                    disable pr MACAddr
                    delall pr
                    load pr
                    shutdown }
```

show stat	: 指定した ME6E アプリの統計情報を表示します。
show conf	: 指定した ME6E アプリ起動時に読込んだ設定情報を表示します。
show arp	: 指定した ME6E アプリの代理 ARP テーブルを表示します。
show ndp	: 指定した ME6E アプリの代理 NDP テーブルを表示します。
*show pr	: 指定した ME6E アプリの PR テーブルを表示します。
add arp	: 指定した ME6E アプリの代理 ARP テーブルに ARP エントリを追加します。
add ndp	: 指定した ME6E アプリの代理 NDP テーブルに NDP エントリを追加します。
*add pr	: 指定した ME6E アプリの PR テーブルに PR エントリを追加します。
del arp	: 指定した ME6E アプリの代理 ARP テーブルから ARP エントリを削除します。
del ndp	: 指定した ME6E アプリの代理 NDP テーブルから NDP エントリを削除します。
*del pr	: 指定した ME6E アプリの PR テーブルから PR エントリを削除します。
*delall pr	: 指定した ME6E アプリの PR テーブルから全ての PR エントリを一括削除します。
*enable pr	: 指定した ME6E アプリの PR テーブルの PR エントリを有効にします。
*disable pr	: 指定した ME6E アプリの PR テーブルの PR エントリを無効にします。
*load pr	: 指定した ME6E アプリの PR テーブルの PR エントリを一括設定します。
shutdown	: 指定した ME6E アプリをシャットダウンします。

\*印で示されるコマンドは、ME6E アプリの動作モードが、ME6E-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[common]セクションの plane\_name で設定した ME6E 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 ME6E 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。 ME6E 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、ME6E 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. 統計情報の表示

ME6E アプリではカプセル化処理をおこなったパケット数や代理 ARP 機能や代理 NDP 機能で送受信したパケット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。

収集項目	内容	備考
<b>[Capsuling]</b>		
Success count	カプセル化に成功したパケット数	
Failure count	カプセル化に失敗したパケット数(送信エラー)	
<b>[DeCapsuling]</b>		
Success count	デカプセル化に成功したパケット数	
Failure count(Unmatched Ether IP)	デカプセル化した失敗したパケット(Next Header が Ether IP でない)	
Failure count(Send error)	デカプセル化に失敗したパケット(送信数)	
<b>[Proxy ARP]</b>		
Recieve ARP Request count	ARP Request 受信数	
Send ARP Reply count	ARP Reply 送信数	
Recieve disease not arp request	対象外 ARP Request 受信(アドレス種別が対象外)	
ARP Reply send error count	ARP Reply 送信エラー数	
<b>[Proxy NDP]</b>		
Recieve NS count	Neighbor Solicitation パケット受信数	
Send NA count	Neighbor Advertisement パケット送信数	
NA send error count	Neighbor Advertisement パケット送信エラー数	

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。

#me6ectl -n me6e0 show stat	
-----	
Statistics information	
-----	
<b>【Capsuling】</b>	
Success count	: 2716582
Failure count	: 0
<b>【DeCapsuling】</b>	
Success count	: 2716582
Failure count(Unmatch EtherIP )	: 0
Failure count(Send error)	: 0
<b>【Proxy ARP】</b>	
Recieve ARP Request count	: 2
Send ARP Reply count	: 2
Recieve disease not arp request	: 0
ARP Reply send error count	: 0
<b>【Proxy NDP】</b>	
Recieve NS count	: 1
Send NA count	: 1
NA send error count	: 0



## 5.2. 設定情報の表示

ME6E アプリ起動時に読み込んだ設定ファイルの情報を表示します。あくまでも ME6E アプリ起動時に読み込んだ設定ファイルの内容なので、ip コマンドなど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ていた場合には、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。

```
#me6ectl -n me6e0 show conf
```

---

Configuration information

---

Config file name = /etc/me6e/me6e.conf

[common]

plane\_name = me6e0  
tunnel\_mode = 0  
debug\_log = no  
daemon = yes

[capsuling]

me6e\_address\_prefix = 2001:db8::/48  
me6e\_multicast\_prefix = ff01:db8:0:46::  
plane\_id = 64:1  
backbone\_physical\_dev = eth1  
stub\_physical\_dev = eth2  
tunnel\_name = me6etun0  
tunnel\_mtu = 1500  
tunnel\_hwaddr = 42:f5:a3:ee:ea:31  
bridge\_name = me6ebr0  
bridge\_hwaddr = aa:bb:cc:dd:ee:ff  
l2multi\_l3uni = no

[proxy\_arp]

ndp\_enable = yes  
ndp\_entry\_update = no  
arp\_aging\_time = 600  
arp\_entry\_max = 125

IPv4 Address	MAC Addr
192.168.100.40	0:26:2d:b:b3:c3
192.168.100.82	0:20:ed:84:3:a1
192.168.100.121	0:26:2d:b:da:d5
192.168.100.25	0:3:47:b1:e7:3d

[proxy\_ndp]

ndp\_enable = yes  
ndp\_entry\_update = no  
ndp\_aging\_time = 600  
ndp\_entry\_max = 125

IPv6 Address	MAC Addr
2001:db8:0:ff31::c0a8:6428	0:26:2d:b:b3:c3
2001:db8:0:ff31::c0a8:6479	0:26:2d:b:da:d5
2001:db8:0:ff31::c0a8:6452	0:20:ed:84:3:a1
2001:db8:0:ff31::c0a8:6419	0:3:47:b1:e7:3d

[mng\_macaddr]

mng\_macaddr\_enable = yes  
mac\_entry\_update = no  
mac\_aging\_time = 600  
mac\_entry\_max = 128  
hosthw\_addr = 0:20:ed:69:d0:cb

---

### 5.3. 代理 ARP テーブルの表示

「3.3 代理 ARP 管理機能」で説明した代理 ARP テーブルの内容を表示します。代理 ARP テーブルで管理している情報には

1. IPv4 アドレス
2. MAC アドレス
3. Aging time

の 3 項目があります。

コマンドを入力すると、以下のように代理 ARP テーブルの ARP エントリが表示されます。

```
#me6ectl -n me6e0 show arp
```

IPv4 Address	MAC Address	aging time
192.168.100.40	0:26:2d:b:b3:c3	static
192.168.100.82	0:20:ed:84:3:a1	static
192.168.100.121	0:26:2d:b:da:d5	456
192.168.100.25	0:3:47:b1:e7:3d	423

### 5.4. 代理 NDP テーブルの表示

「3.4 代理 NDP 管理機能」で説明した代理 NDP テーブルの内容を表示します。代理 NDP テーブルで管理している情報には

1. IPv6 アドレス
2. MAC アドレス
3. Aging time

の 3 項目があります。

コマンドを入力すると、以下のように代理 NDP テーブルの NDP エントリが表示されます。

```
#me6ectl -n me6e0 show ndp
```

IPv6 Address	MAC Address	aging time
2001:db8:0:ff31::c0a8:6428	0:26:2d:b:b3:c3	static
2001:db8:0:ff31::c0a8:6479	0:26:2d:b:da:d5	static
2001:db8:0:ff31::c0a8:6452	0:20:ed:84:3:a1	589
2001:db8:0:ff31::c0a8:6419	0:3:47:b1:e7:3d	587

## 5.5. Prefix Resolution テーブルの表示

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルの内容を表示します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

Prefix Resolution テーブルで管理している情報には

1. Plane ID
2. MAC アドレス
3. Net マスク
4. ME6E-PR アドレスプレフィックス

の 4 項目があります。

コマンドを入力すると、以下のように Prefix Resolution テーブルの PR エントリが表示されます。

```
#me6ectl -n me6e0 show pr
```

ME6E Prefix Resolution Table				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix
*	1	0:16:3e:20:52:a1	/128	2001:db8:ff10:10::
*	1	0:16:3e:6e:c3:77	/128	2001:db8:ff20:20::
*	1	0:16:3e:15:df:28	/128	2001:db8:ff20:30::

Note : [\*] shows available entry for prefix resolution process

## 5.6. 代理 ARP エントリの追加

「3.3 代理 ARP 管理機能」で説明した代理 ARP テーブルに ARP エントリを追加します。

コマンドを入力すると、以下のように代理 ARP テーブルの ARP エントリが追加されます。

```
#me6ectl -n me6e0 add arp 192.168.100.27 00:26:2d:b:b3:c4
```

IPv4 Address	MAC Address	aging time
192.168.100.40	0:26:2d:b:b3:c3	static
192.168.100.82	0:20:ed:84:3:a1	static
192.168.100.121	0:26:2d:b:da:d5	456
192.168.100.25	0:3:47:b1:e7:3d	423
192.168.100.27	0:26:2d:b:b3:c4	static

## 5.7. 代理 NDP エントリの追加

「3.4 代理 NDP 管理機能」で説明した代理 NDP テーブルに NDP エントリを追加します。

コマンドを入力すると、以下のように代理 NDP テーブルの NDP エントリが追加されます。

```
#me6ectl -n me6e0 add arp 2001:db8:0:ff31::c0a8:6429 0:26:2d:b:b3:c4
```

IPv6 Address	MAC Address	aging time
2001:db8:0:ff31::c0a8:6428	0:26:2d:b:b3:c3	static
2001:db8:0:ff31::c0a8:6429	0:26:2d:b:b3:c4	static
2001:db8:0:ff31::c0a8:6479	0:26:2d:b:da:d5	static
2001:db8:0:ff31::c0a8:6452	0:20:ed:84:3:a1	589
2001:db8:0:ff31::c0a8:6419	0:3:47:b1:e7:3d	587

## 5.8. Prefix Resolution エントリの追加

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルに Prefix Resolution エントリを追加します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

コマンドを入力すると、以下のように Prefix Resolution テーブルの Prefix Resolution エントリが追加されます。

```
#me6ectl -n me6e0 add pr 0:16:3e:6e:c3:77 2001:db8:ff20:20::/64 enable
```

ME6E Prefix Resolution Table				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix
*	1	0:16:3e:20:52:a1	/128	2001:db8:ff10:10::
*	1	0:16:3e:6e:c3:77	/128	2001:db8:ff20:20::
*	1	0:16:3e:15:df:28	/128	2001:db8:ff20:30::

Note : [\*] shows available entry for prefix resolution process

## 5.9. 代理 ARP エントリの削除

「3.3 代理 ARP 管理機能」で説明した代理 ARP テーブルから ARP エントリを削除します。

コマンドを入力すると、以下のように代理 ARP テーブルの ARP エントリが削除されます。

```
#me6ectl -n me6e0 del arp 192.168.100.27
```

IPv4 Address	MAC Address	aging time
192.168.100.40	0:26:2d:b:b3:c3	static
192.168.100.82	0:20:ed:84:3:a1	static
192.168.100.121	0:26:2d:b:da:d5	426
192.168.100.25	0:3:47:b1:e7:3d	393

## 5.10. 代理 NDP エントリの削除

「3.4 代理 NDP 管理機能」で説明した代理 NDP テーブルから NDP エントリを削除します。

コマンドを入力すると、以下のように代理 NDP テーブルから NDP エントリが削除されます。

```
#me6ectl -n me6e0 add arp 2001:db8:0:ff31::c0a8:6429
```

IPv6 Address	MAC Address	aging time
2001:db8:0:ff31::c0a8:6428	0:26:2d:b:b3:c3	static
2001:db8:0:ff31::c0a8:6479	0:26:2d:b:da:d5	static
2001:db8:0:ff31::c0a8:6452	0:20:ed:84:3:a1	589
2001:db8:0:ff31::c0a8:6419	0:3:47:b1:e7:3d	587

### 5.11. Prefix Resolution エントリの削除

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルから Prefix Resolution エントリを削除します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

コマンドを入力すると、以下のように Prefix Resolution テーブルから Prefix Resolution エントリが削除されます。

```
#me6ectl -n me6e0 del pr 0:16:3e:6e:c3:77
```

ME6E Prefix Resolution Table				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix
*	1	0:16:3e:20:52:a1	/128	2001:db8:ff10:10::
*	1	0:16:3e:15:df:28	/128	2001:db8:ff20:30::

Note : [\*] shows available entry for prefix resolution process

### 5.12. Prefix Resolution エントリの一括削除

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルから Prefix Resolution エントリを一括削除します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

コマンドを入力すると、以下のように Prefix Resolution テーブルから Prefix Resolution エントリが一括削除されます。

```
#me6ectl -n me6e0 delall pr
```

ME6E Prefix Resolution Table				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix

### 5.13. Prefix Resolution エントリの有効化

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルから Prefix Resolution エントリを有効化します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

コマンドを入力すると、以下のように Prefix Resolution テーブルから Prefix Resolution エントリが有効化されます。

```
#me6ectl -n me6e0 enable pr 0:16:3e:20:52:a1
```

/ ME6E Prefix Resolution Table /				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix
*	1	0:16:3e:20:52:a1	/128	2001:db8:ff10:10::
*	1	0:16:3e:15:df:28	/128	2001:db8:ff20:30::

Note : [\*] shows available entry for prefix resolution process

### 5.14. Prefix Resolution エントリの無効化

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルから Prefix Resolution エントリを無効化します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

コマンドを入力すると、以下のように Prefix Resolution テーブルから Prefix Resolution エントリが無効化されます。

```
#me6ectl -n me6e0 disable pr 0:16:3e:20:52:a1
```

/ ME6E Prefix Resolution Table /				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix
	1	0:16:3e:20:52:a1	/128	2001:db8:ff10:10::
*	1	0:16:3e:15:df:28	/128	2001:db8:ff20:30::

Note : [\*] shows available entry for prefix resolution process

### 5.15. Prefix Resolution エントリの一括設定

「3.6 Prefix Resolution 機能」で説明した Prefix Resolution テーブルから Prefix Resolution エントリを一括設定します。本コマンドは ME6E アプリの動作モードが ME6E-PR の場合のみ有効です。

コマンドを入力すると、以下のように Prefix Resolution テーブルから Prefix Resolution エントリが一括設定されます。

```
#me6ectl -n me6e0 load pr pr_entry_list.txt
```

ME6E Prefix Resolution Table				
	Plane ID	MAC Address	Netmask	ME6E-PR Address Prefix
*	1	0:16:3e:20:52:a1	/128	2001:db8:ff10:10::
*	1	0:16:3e:15:df:28	/128	2001:db8:ff20:30::

Note : [\*] shows available entry for prefix resolution process

ME6E-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。

```
# ME6E-PR table entry setting list
add pr 0:16:3e:20:52:a1 2001:db8:ff10:10::/64 enable (追加 : enable エントリ)
add pr 0:16:3e:15:df:28 2001:db8:ff30:30::/64 disable (追加 : disable エントリ)
del pr 0:16:3e:6e:c3:77 (削除)
enable pr 0:16:3e:15:df:28 (活性化)
disble pr 0:16:3e:20:52:a1 (非活性化)
```

## 5.16. アプリケーション終了

指定した ME6E 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。

```
#me6ectl -n me6e0 shutdown
```



## 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、ME6E 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。

以下の表に想定される用途の一覧を示します。これらのうち、番号 1,3,4,10,11 のケースについて次項より詳細に説明いたします。その他の番号のケースは、これらのケースの組み合わせとなりますので参考にして設定をおこなってください。

番号	Plane ID	Stub Network のタイプ	エントリ設定方法	想定される用途
1	無し	IPv4	静的	閉域網における IPv4 網の拠点間 VPN 接続。
2			動的(マルチキャスト)	閉域網における IPv4 網の拠点間 VPN 接続。
3			動的(ユニキャスト)	閉域網における IPv4 網の拠点間 VPN 接続。
4		IPv6	静的	閉域網における IPv6 網の拠点間 VPN 接続。
5			動的(マルチキャスト)	閉域網における IPv6 網の拠点間 VPN 接続。
6			動的(ユニキャスト)	閉域網における IPv6 網の拠点間 VPN 接続。
7		IPv4/v6 dual	静的	閉域網における IPv4/v6 dual 網の拠点間 VPN 接続。
8			動的(マルチキャスト)	閉域網における IPv4/v6 dual 網の拠点間 VPN 接続。
9			動的(ユニキャスト)	閉域網における IPv4/v6 dual 網の拠点間 VPN 接続。
10	有り	IPv4	静的	ISP (Internet Service Provider)事業者による IPv4 網の拠点間 VPN 接続サービス。
11			動的(マルチキャスト)	ISP (Internet Service Provider)事業者による IPv4 網の拠点間 VPN 接続サービス。
12			動的(ユニキャスト)	ISP (Internet Service Provider)事業者による IPv4 網の拠点間 VPN 接続サービス。
13		IPv6	静的	ISP (Internet Service Provider)事業者による IPv6 網の拠点間 VPN 接続サービス。
14			動的(マルチキャスト)	ISP (Internet Service Provider)事業者による IPv6 網の拠点間 VPN 接続サービス。
15			動的(ユニキャスト)	ISP (Internet Service Provider)事業者による IPv6 網の拠点間 VPN 接続サービス。
16		IPv4/v6 dual	静的	ISP (Internet Service Provider)事業者による IPv4/v6 網の拠点間 VPN 接続サービス。
17			動的(マルチキャスト)	ISP (Internet Service Provider)事業者による IPv4/v6 網の拠点間 VPN 接続サービス。
18			動的(ユニキャスト)	ISP (Internet Service Provider)事業者による IPv4/v6 網の拠点間 VPN 接続サービス。

## 6.1. Plane ID 無し: Stub Network が IPv4: 静的エントリ設定

Plane ID 無しの場合の ME6E 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。

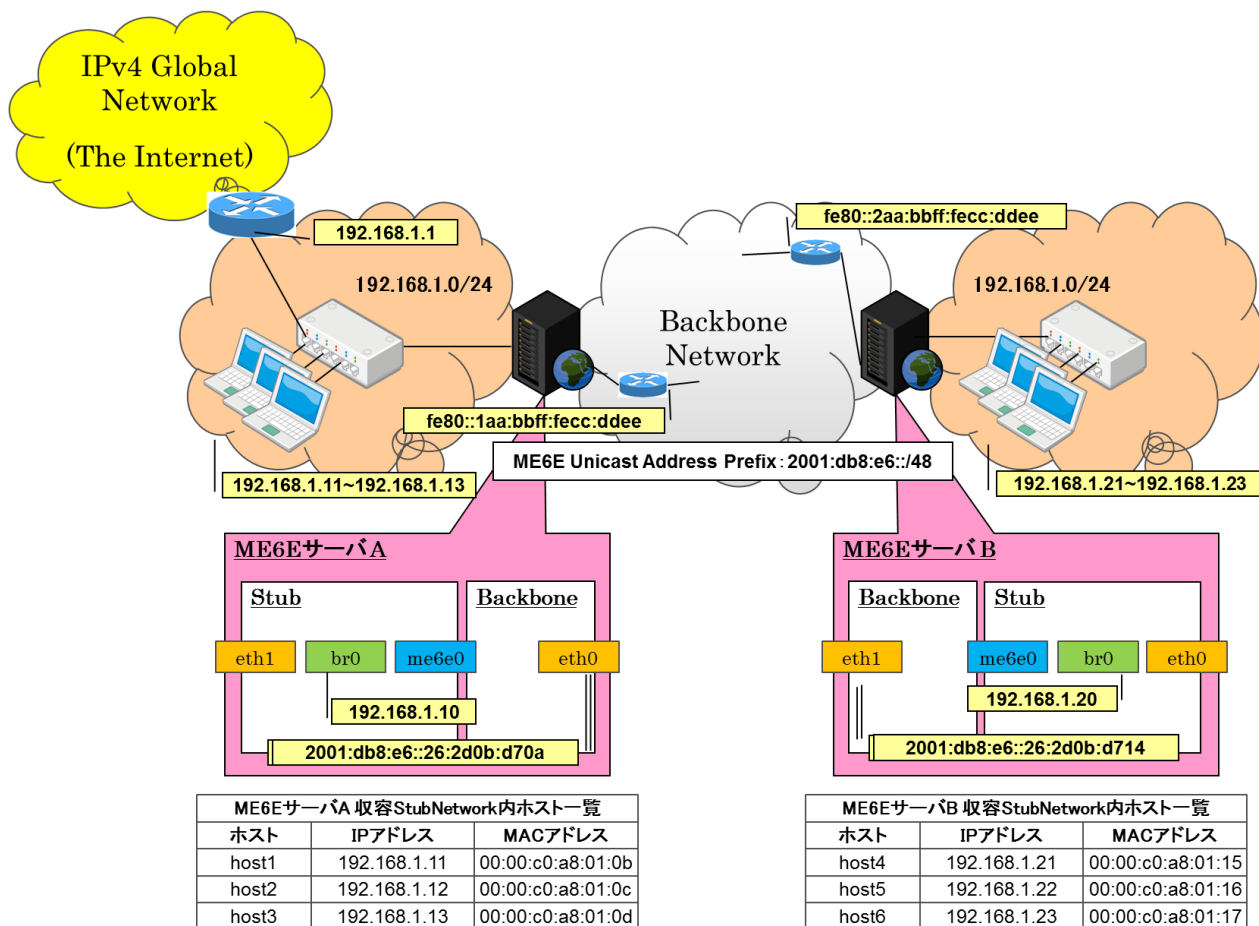


図1 Plane ID 無し : Stub Network が IPv4 : 静的エントリ設定の場合のネットワーク構成

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。
- 静的エントリ設定なので、自拠点の Stub Network 内ホストの MAC アドレスを設定する。
- 静的エントリ設定なので、対向拠点の Stub Network 内ホストの ARP エントリを設定する。
- 対向拠点のホスト宛の経路は、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。対向拠点のホスト宛の経路は、ospf6 等の経路制御プロトコルを用いて動的に設定させることも可能です。

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。

## ME6E サーバ A の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth0
stub_physical_dev = eth1
tunnel_name = me6e0
bridge_name = br0

[proxy_arp]
arp_enable = yes
arp_entry_update = no
arp_entry_max = 255
192.168.1.21 = 00:00:c0:a8:01:15
192.168.1.22 = 00:00:c0:a8:01:16
192.168.1.23 = 00:00:c0:a8:01:17

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:01:0b
hosthw_addr = 00:00:c0:a8:01:0c
hosthw_addr = 00:00:c0:a8:01:0d
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0115/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0116/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0117/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0

exit 0
```

## ME6E サーバ B の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth1
stub_physical_dev = eth0
tunnel_name = me6e0
bridge_name = br0

[proxy_arp]
arp_enable = yes
arp_entry_update = no
arp_entry_max = 255
192.168.1.11 = 00:00:c0:a8:01:0b
192.168.1.12 = 00:00:c0:a8:01:0c
192.168.1.13 = 00:00:c0:a8:01:0d

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:01:15
hosthw_addr = 00:00:c0:a8:01:16
hosthw_addr = 00:00:c0:a8:01:17
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1

exit 0
```

6.2. Plane ID 無し: Stub Network が IPv4:動的エン트리設定(ユニキャスト)

Plane ID 無しの場合の ME6E 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。

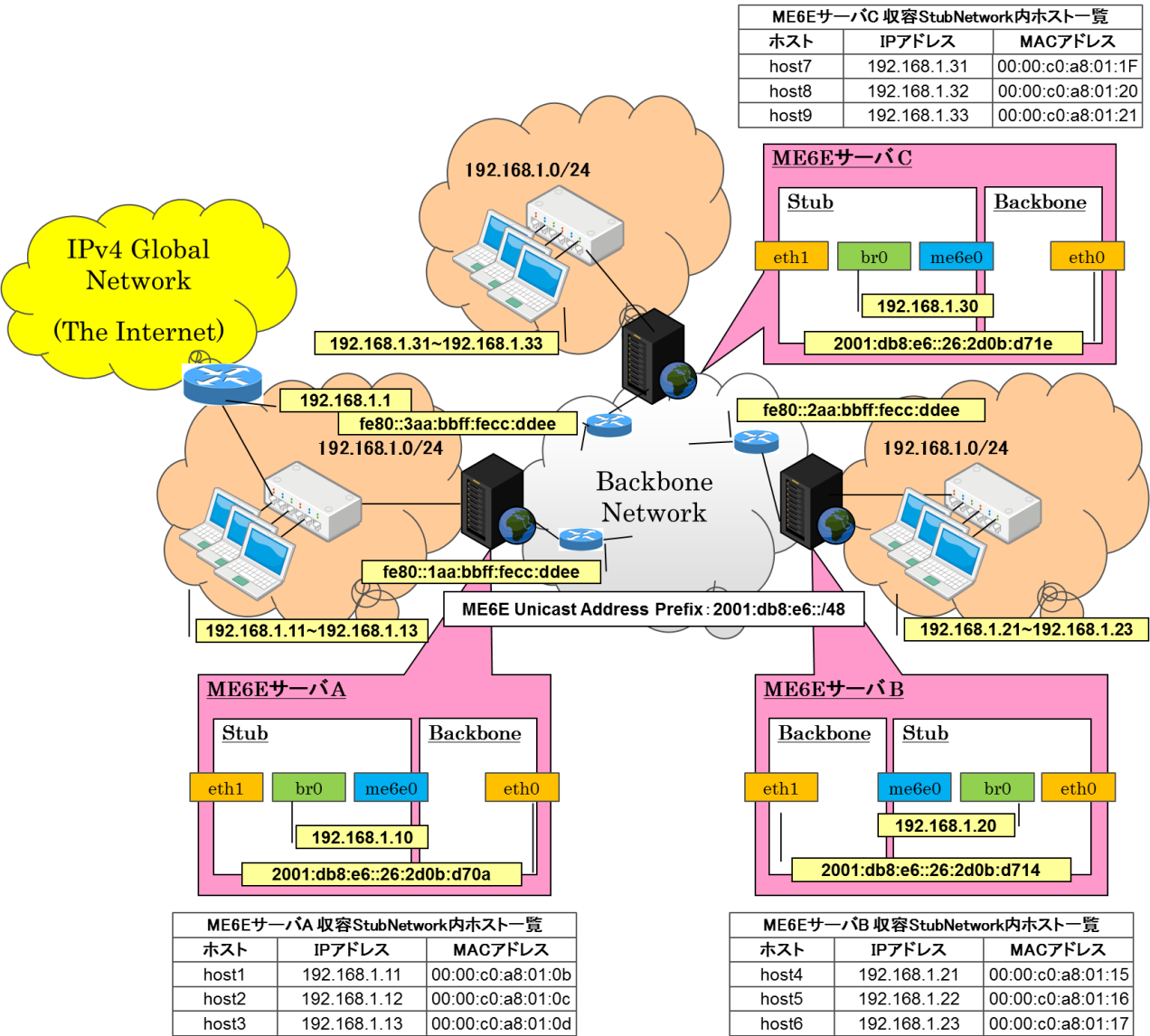


図 2 Plane ID 無し : Stub Network が IPv4 : 動的エン트리設定(ユニキャスト)の場合のネットワーク構成

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。
- 静的エン트리設定なので、自拠点の Stub Network 内ホストの MAC アドレスを設定する。
- 静的エン트리設定なので、対向拠点の Stub Network 内ホストの ARP エントリを設定する。
- 対向拠点のホスト宛の経路は、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。対向拠点のホスト宛の経路は、本経路の設定は、ospf6 等の経路制御プロトコルを用いて動的に設

定させることも可能です。

サーバ A, サーバ B, サーバ C 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。

## ME6E サーバ A の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth0
stub_physical_dev = eth1
tunnel_name = me6e0
bridge_name = br0
l2multi_l3uni = yes
me6e_host_address =2001:db8:e6::26:2d0b:d714
me6e_host_address =2001:db8:e6::26:2d0b:d71e

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0115/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0116/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0117/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:011f/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0120/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0121/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
exit 0
```

## ME6E サーバ B の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth1
stub_physical_dev = eth0
tunnel_name = me6e0
bridge_name = br0
l2multi_l3uni = yes
me6e_host_address =2001:db8:e6::26:2d0b:d70a
me6e_host_address =2001:db8:e6::26:2d0b:d71e

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:011f/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0120/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0121/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
exit 0
```

## ME6E サーバ C の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth0
stub_physical_dev = eth1
tunnel_name = me6e0
bridge_name = br0
l2multi_l3uni = yes
me6e_host_address =2001:db8:e6::26:2d0b:d70a
me6e_host_address =2001:db8:e6::26:2d0b:d714

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010b/128 ¥
via fe80::3aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010c/128 ¥
via fe80::3aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010d/128 ¥
via fe80::3aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0115/128 ¥
via fe80::3aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0116/128 ¥
via fe80::3aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0117/128 ¥
via fe80::3aa:bbff:fecc:ddee dev eth0
exit 0
```

### 6.3. Plane ID 無し: Stub Network が IPv6: 静的エントリ設定

Plane ID 無しの ME6E 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。

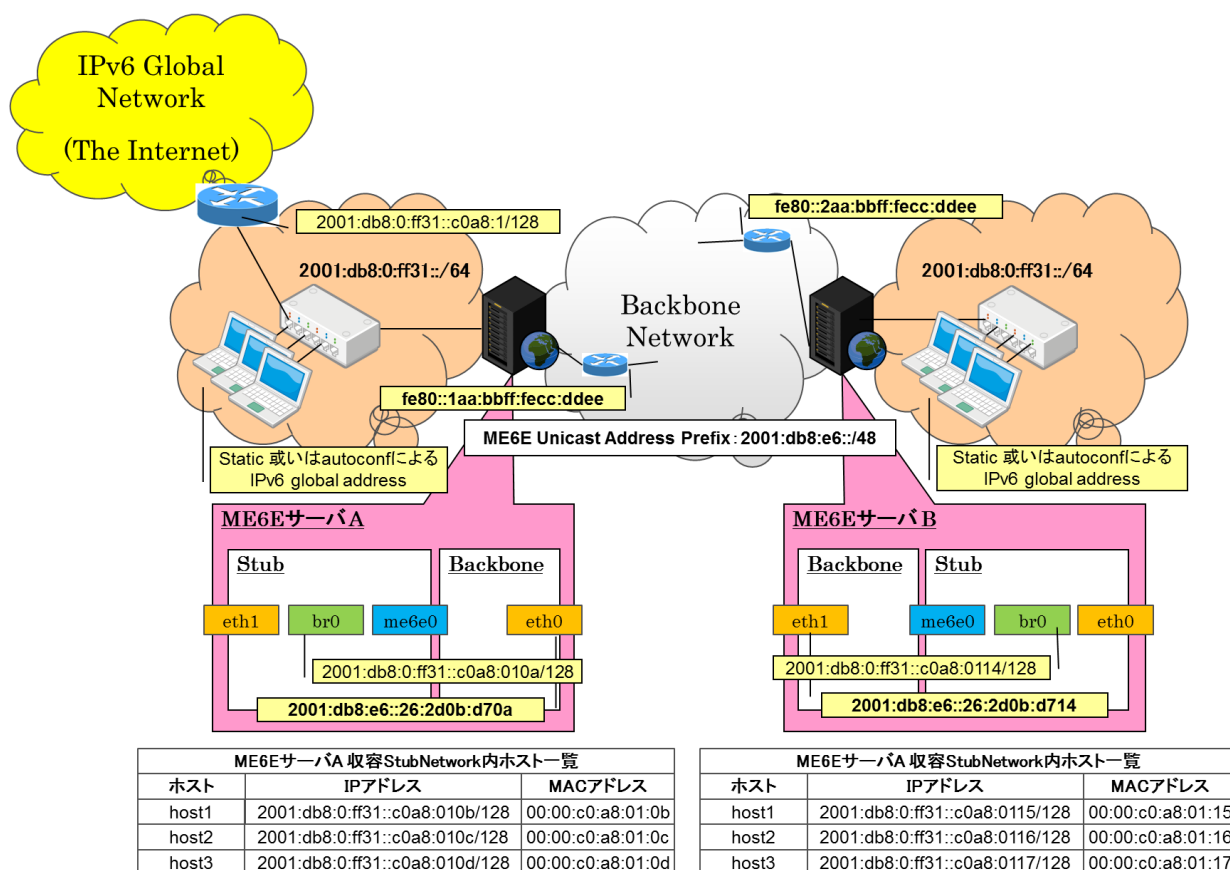


図3 Plane ID 無し : Stub Network が IPv6 : 静的エントリ設定の場合のネットワーク構成

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。
- 静的エントリ設定なので、自拠点の Stub Network 内ホストの MAC アドレスを設定する。
- 静的エントリ設定なので、対向拠点の Stub Network 内ホストの ARP エントリを設定する。
- 対向拠点のホスト宛の経路は、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。対向拠点のホスト宛の経路は、ospf6 等の経路制御プロトコルを用いて動的に設定させることも可能です。

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。



## ME6E サーバ A の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth0
stub_physical_dev = eth1
tunnel_name = me6e0
bridge_name = br0

[proxy_ndp]
ndp_enable = yes
ndp_entry_update = no
ndp_entry_max = 255
2001:db8:0:ff31::c0a8:0115 = 00:00:c0:a8:01:15
2001:db8:0:ff31::c0a8:0116 = 00:00:c0:a8:01:16
2001:db8:0:ff31::c0a8:0117 = 00:00:c0:a8:01:17

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:01:0b
hosthw_addr = 00:00:c0:a8:01:0c
hosthw_addr = 00:00:c0:a8:01:0d
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0115/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0116/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:0117/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0

exit 0
```

## ME6E サーバ B の設定

### 設定ファイル

```
[common]
plane_name = me6e0
startup_script = /etc/me6e/me6e0_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
backbone_physical_dev = eth1
stub_physical_dev = eth0
tunnel_name = me6e0
bridge_name = br0

[proxy_ndp]
ndp_enable = yes
ndp_entry_update = no
ndp_entry_max = 255
2001:db8:0:ff31::c0a8:010b = 00:00:c0:a8:01:0b
2001:db8:0:ff31::c0a8:010c = 00:00:c0:a8:01:0c
2001:db8:0:ff31::c0a8:010d = 00:00:c0:a8:01:0d

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:01:15
hosthw_addr = 00:00:c0:a8:01:16
hosthw_addr = 00:00:c0:a8:01:17
```

### スタートアップスクリプト

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:0:0:c0a8:010d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1

exit 0
```

## 6.4. Plane ID 有り: Stub Network が IPv4: 静的エントリ設定

Plane ID 有りの ME6E 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。

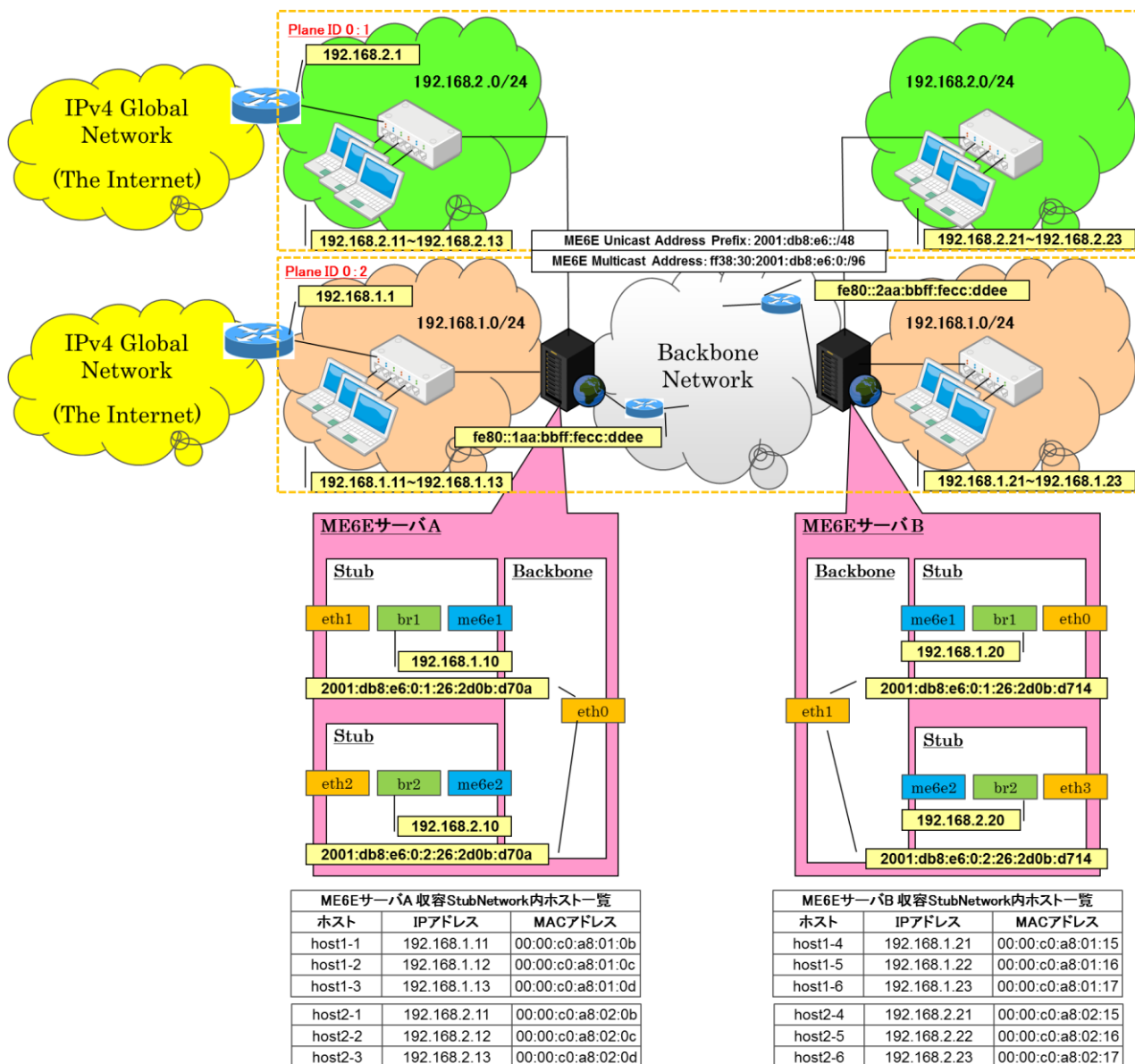


図4 Plane ID 有り: Stub Network が IPv4: 静的エントリ設定の場合のネットワーク構成

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点になります。

本例では、Plane 毎に IPv4 アドレス体系の異なるネットワークを収容する例記載しておりますが、ネットワーク構成図に示すとおり、Plane ID が異なっていれば、同一の IPv4 アドレス体系のネットワークを収容することが可能です。

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。

## ME6E サーバ A の設定

### 設定ファイル(Plane #0:1)

```
[common]
plane_name = me6e1
startup_script = /etc/me6e/me6e1_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:1
backbone_physical_dev = eth0
stub_physical_dev = eth1
tunnel_name = me6e1
bridge_name = br1

[proxy_arp]
arp_enable = yes
arp_entry_update = no
arp_entry_max = 255
192.168.1.21 = 00:00:c0:a8:01:15
192.168.1.22 = 00:00:c0:a8:01:16
192.168.1.23 = 00:00:c0:a8:01:17

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:01:0b
hosthw_addr = 00:00:c0:a8:01:0c
hosthw_addr = 00:00:c0:a8:01:0d
```

### スタートアップスクリプト(Plane #0:1)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:0115/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:0116/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:0117/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0

exit 0
```

## ME6E サーバ B の設定

### 設定ファイル(Plane #0:1)

```
[common]
plane_name = me6e1
startup_script = /etc/me6e/me6e1_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:1
backbone_physical_dev = eth1
stub_physical_dev = eth0
tunnel_name = me6e1
bridge_name = br1

[proxy_arp]
arp_enable = yes
arp_entry_update = no
arp_entry_max = 255
192.168.1.11 = 00:00:c0:a8:01:0b
192.168.1.12 = 00:00:c0:a8:01:0c
192.168.1.13 = 00:00:c0:a8:01:0d

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:01:15
hosthw_addr = 00:00:c0:a8:01:16
hosthw_addr = 00:00:c0:a8:01:17
```

### スタートアップスクリプト(Plane #0:1)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:010b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:010c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:010d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1

exit 0
```

## ME6E サーバ A の設定(続き)

### 設定ファイル(Plane #0:2)

```
[common]
plane_name = me6e2
startup_script = /etc/me6e/me6e2_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:2
backbone_physical_dev = eth0
stub_physical_dev = eth2
tunnel_name = me6e2
bridge_name = br2

[proxy_arp]
arp_enable = yes
arp_entry_update = no
arp_entry_max = 255
192.168.2.21 = 00:00:c0:a8:02:15
192.168.2.22 = 00:00:c0:a8:02:16
192.168.2.23 = 00:00:c0:a8:02:17

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:02:0b
hosthw_addr = 00:00:c0:a8:02:0c
hosthw_addr = 00:00:c0:a8:02:0d
```

### スタートアップスクリプト(Plane #0:2)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:0215/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:0216/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:0217/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0

exit 0
```

## ME6E サーバ B の設定(続き)

### 設定ファイル(Plane #0:2)

```
[common]
plane_name = me6e2
startup_script = /etc/me6e/me6e2_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:2
backbone_physical_dev = eth1
stub_physical_dev = eth3
tunnel_name = me6e2
bridge_name = br2

[proxy_arp]
arp_enable = yes
arp_entry_update = no
arp_entry_max = 255
192.168.2.11 = 00:00:c0:a8:02:0b
192.168.2.12 = 00:00:c0:a8:02:0c
192.168.2.13 = 00:00:c0:a8:02:0d

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = no
mac_entry_max = 255
hosthw_addr = 00:00:c0:a8:02:15
hosthw_addr = 00:00:c0:a8:02:16
hosthw_addr = 00:00:c0:a8:02:17
```

### スタートアップスクリプト(Plane #0:2)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:020b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:020c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:020d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1

exit 0
```

## 6.5. Plane ID 有り: Stub Network が IPv4:動的エントリー設定(マルチキャスト)

Plane ID 有りの ME6E 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。

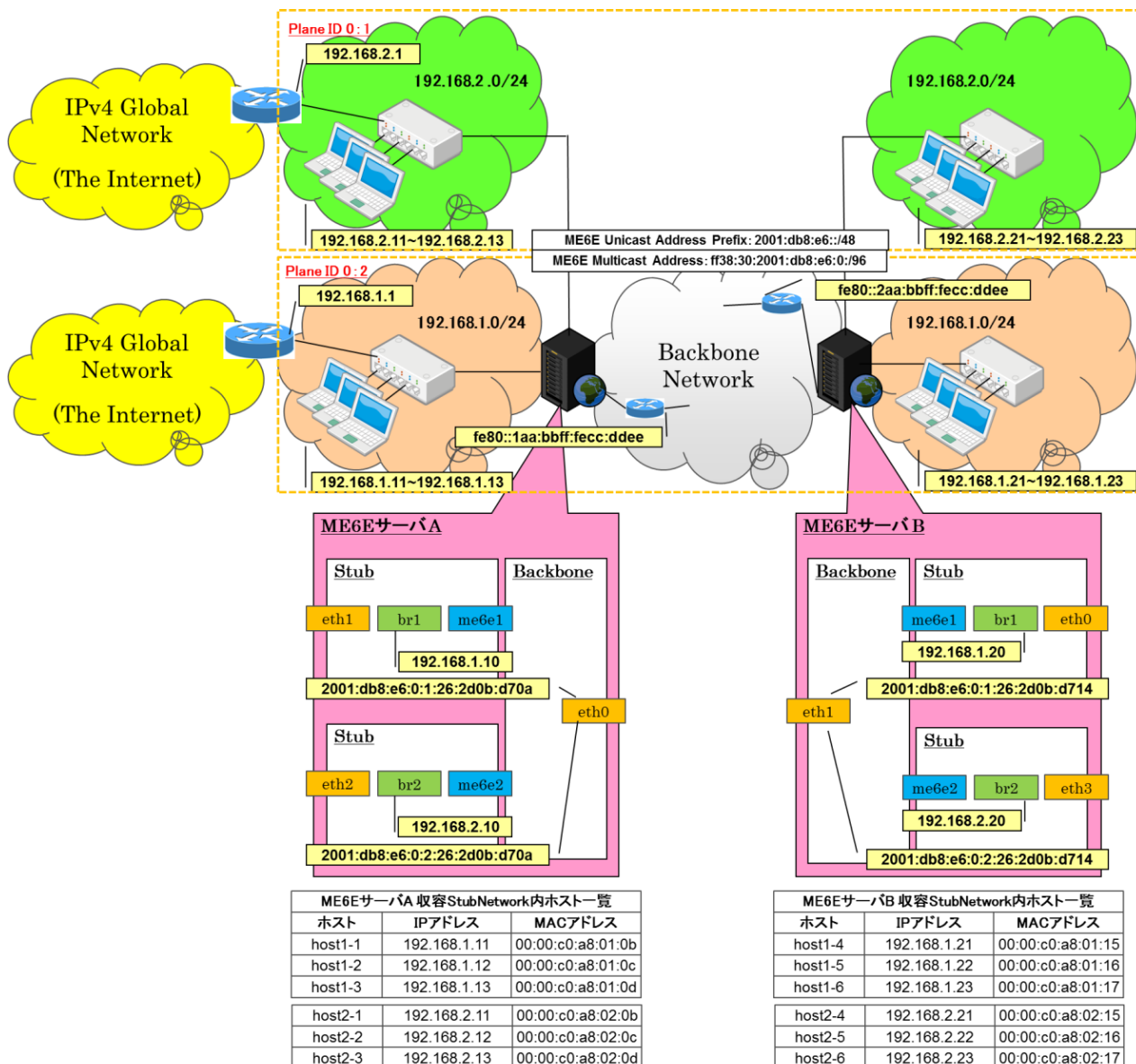


図5 Plane ID 有り : Stub Network が IPv4:動的エントリー設定(マルチキャスト)の場合のネットワーク構成

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 点になります。

本例では、Plane 毎に IPv4 アドレス体系の異なるネットワークを収容する例記載しておりますが、ネットワーク構成図に示すとおり、Plane ID が異なっていれば、同一の IPv4 アドレス体系のネットワークを収容することが可能です。

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。

## ME6E サーバ A の設定

### 設定ファイル(Plane #0:1)

```
[common]
plane_name = me6e1
startup_script = /etc/me6e/me6e1_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:1
backbone_physical_dev = eth0
stub_physical_dev = eth1
tunnel_name = me6e1
bridge_name = br1

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト(Plane #0:1)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:0115/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:0116/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:0117/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0

exit 0
```

## ME6E サーバ B の設定

### 設定ファイル(Plane #0:1)

```
[common]
plane_name = me6e1
startup_script = /etc/me6e/me6e1_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:1
backbone_physical_dev = eth1
stub_physical_dev = eth0
tunnel_name = me6e1
bridge_name = br1

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト(Plane #0:1)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:010b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:010c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:1:0:c0a8:010d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1

exit 0
```



## ME6E サーバ A の設定(続き)

### 設定ファイル(Plane #0:2)

```
[common]
plane_name = me6e2
startup_script = /etc/me6e/me6e2_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:2
backbone_physical_dev = eth0
stub_physical_dev = eth2
tunnel_name = me6e2
bridge_name = br2

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト(Plane #0:2)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:0215/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:0216/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:0217/128 ¥
via fe80::1aa:bbff:fecc:ddee dev eth0

exit 0
```

## ME6E サーバ B の設定(続き)

### 設定ファイル(Plane #0:2)

```
[common]
plane_name = me6e2
startup_script = /etc/me6e/me6e2_startup.sh

[capsuling]
me6e_unicast_prefix = 2001:db8:e6::/48
me6e_multicast_prefix = ff38:0030:2001:db8:e6::
plane_id = 0:2
backbone_physical_dev = eth1
stub_physical_dev = eth3
tunnel_name = me6e2
bridge_name = br2

[proxy_arp]
arp_enable = yes
arp_entry_update = yes
arp_entry_max = 255

[mng_macaddr]
mng_macaddr_enable = yes
mac_entry_update = yes
mac_entry_max = 255
```

### スタートアップスクリプト(Plane #0:2)

```
#!/bin/bash

ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:020b/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:020c/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1
ip -6 route add ¥
2001:db8:e6:0:2:0:c0a8:020d/128 ¥
via fe80::2aa:bbff:fecc:ddee dev eth1

exit 0
```

## 付録 1. 設定ファイルサンプル

```
#####
# ME6E 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)
#####
[common]
#####
# ME6E の動作モード。以下が設定可能 (省略不可)
# ・0 : 通常モード
# ・1 : ME6E-PR モード
tunnel_mode          = 1
#####
# plane 名。(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            = me6e0
#####
# 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             = yes
#####
# 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                = no
#####
# スタートアップスクリプト (省略可)
# アプリ起動前に実行したいコマンド (経路情報の追加など) を記載する。
# ※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/me6e/me6e_startup.sh

#####
# カプセリング固有の設定 (省略不可)
#####
[capsuling]
#####
# ME6E 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。
#
# 【ユニキャストアドレスフォーマット】
# +-----80bit-----+
# +-----N bit-----+M bit-----+-----48bit-----+
# | ME6E address prefix | planID | MAC address |
# +-----128bit-----+
# ※N、M の bit 長は可変です。但し、M は最大 32bit までです。
#
# マルチキャスト転送を動作させる場合は、N は 64bit 以内に設定してください。(RFC3306 準拠)
#
# prefix 長は最大 80bit まで。
me6e_address_prefix   = 2001:db8:0::/48
#####
# ME6E 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。
#
# 先頭から、96bit (ME6E address prefix) まで、
# 当該設定ファイルで指定してください。
# planeID 部分は、アプリケーションが付け足します。
#
# 【マルチキャストアドレスフォーマット】
# +---8bit---+---4bit---+---4bit---+---8bit---+---8bit---+-----64bit-----+1bit---+31bit---+
# | 11111111 | flags | scope | rsv | plen | ME6E address prefix | 1 | planeID |
# +-----96bit-----+-----+-----+
#
# 例.
#
#   me6e_address_prefix が「2001:db8:1111:2222::/64」の場合は、
#   me6e_multicast_prefix は「ff38:0040:2001:db8:1111:2222::」と設定する。
#
#   また、me6e_address_prefix が「2001:db8:1111::/48」のように、
#   prefix 長が、64bit 以内の場合は、
#   me6e_multicast_prefix は「ff38:0030:2001:db8:1111::」と設定する。(RFC3306 準拠)
#   ※マルチキャストの ME6E address prefix フィールドは、ユニキャストアドレスの
#   ME6E address prefix を上位ビットから設定し、残りの下位ビットは、0 に設定する。
#
# prefix 長は 96 固定。
me6e_multicast_prefix = ff38:0030:2001:db8:0::
#####
# PlaneID。IPv6 表記で設定すること。(省略可)
# 省略した場合は PlaneID 無しとして扱う。
# 32bit 固定。
plane_id = 64:1
#####
# マルチキャストのホップリミット数 (省略可)
# 設定可能範囲 : 1~255
# 省略時のデフォルト値 : 64
hop_limit = 64
#####
# Backbone ネットワーク内での物理デバイス名 (省略不可)
backbone_physical_dev = eth1
#####
# Stub ネットワーク内での物理デバイス名 (省略不可)
stub_physical_dev = eth2
#####
# 生成するトンネルデバイス名 (省略不可)
# ※半角英数字で 15 文字まで設定可能とする。
tunnel_name = me6etun0
#####
# トンネル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。
tunnel_mtu = 1500
#####
# 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
# ※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。
#tunnel_hwaddr = 22:33:44:55:11:22
#####
# 生成するブリッジデバイス名 (省略不可)

```

```

# ※半角英数字で 15 文字まで設定可能とする。
bridge_name          = me6ebr0
#####
# ブリッジデバイスに設定する MAC アドレス（省略可）
bridge_hwaddr        = aa:bb:cc:dd:ee:ff
#####
# L2 マルチ-L3 ユニキャスト機能をどうさせるかどうか（省略可）
#   yes : 動作する
#   no  : 動作しない(デフォルト)
l2multi_l3uni        = no
#####
# L2 マルチ-L3 ユニキャスト機能動作時の送信先 ME6E サーバアドレス
# 複数指定可能。
# L2 マルチ-L3 ユニキャスト機能(l2multi_l3umi)に「yes」が設定されている場合は、
# 省略不可。「no」が設定されている場合は、省略可。
me6e_host_address    = 2001:db8:0:64::1
me6e_host_address    = 2001:db8:0:64::2
#####
# ME6-PR の送信元アドレスの unicast prefix address
# IPv6 ユニキャストアドレス形式で設定すること。
me6e_pr_unicast_prefix = 2001:db8:ff10:10::/64
#####
# 代理 ARP 固有の設定（省略不可）
#####
[proxy_arp]
#####
# 代理 ARP 機能を動作させるかどうか（省略可）
#   yes : 動作する(デフォルト)
#   no  : 動作しない
arp_enable           = yes
#####
# ARP テーブルへの動的エントリの追加/更新をおこなうかどうか(省略可)
#   yes : 動作する(デフォルト)
#   no  : 動作しない
arp_entry_update     = no
#####
# APR エントリー保持期限(秒)（省略可）
# 設定可能範囲：301～65535
# 省略時のデフォルト値：600
arp_aging_time       = 600
#####
# APR エントリーの登録できる静的と動的エントリの最大数（省略不可）
# 設定可能範囲：1～65535
arp_entry_max        = 125
#####
# ARP テーブルへの静的エントリ(省略可)
# 複数指定可能。
192.168.0.10         = 00:1D:73:E6:BE:10
192.168.0.11         = 00:1D:73:E6:BE:12
#####
# 代理 NDP 固有の設定（省略不可）
#####
[proxy_ndp]
#####
# 代理 NDP 機能を動作させるかどうか（省略可）
#   yes : 動作する(デフォルト)
#   no  : 動作しない
ndp_enable           = yes

```

```
#####
# NDP テーブルへの動的エントリの追加/更新をおこなうかどうか(省略可)
#   yes : 動作する(デフォルト)
#   no  : 動作しない
ndp_entry_update      = no
#####
# NDP エントリー保持期限(秒) (省略可)
# 設定可能範囲 : 301~65535
# 省略時のデフォルト値 : 600
ndp_aging_time        = 600
#####
# NDP エントリーの登録できる静的と動的エントリの最大数 (省略不可)
# 設定可能範囲 : 1~65535
ndp_entry_max         = 125
#####
# NDP テーブルへの静的エントリ(省略可)
# 複数指定可能。
fec0::10              = 00:1D:73:E6:BE:20
fec0::11              = 00:1D:73:E6:BE:21
#####
# MAC 管理固有の設定 (省略不可)
#####
[mng_macaddr]
#####
# MAC アドレス管理機能を動作させるかどうか (省略可)
#   yes : 動作する(デフォルト)
#   no  : 動作しない
mng_macaddr_enable    = yes
#####
# Stub ネットワーク配下の ME6E アドレスの動的の追加/更新をおこなうかどうか(省略可)
#   yes : 動作する(デフォルト)
#   no  : 動作しない
mac_entry_update      = no
#####
# Stub ネットワーク配下の ME6E アドレス有効期限(秒) (省略可)
# 設定可能範囲 : 301~65535
# 省略時のデフォルト値 : 600
mac_vaild_lifetime    = 600
#####
# Stub ネットワーク配下の ME6E アドレスの登録できる静的と動的エントリの最大数 (省略不可)
# 設定可能範囲 : 1~65535
mac_entry_max         = 128
#####
# Stub ネットワーク配下の ME6E アドレスの静的エントリー(省略可)
# 複数指定可能。
hosthw_addr           = 00:44:55:66:77:11
hosthw_addr           = 00:44:55:66:77:12
#####
# ME6E-PR Table 管理(動作モード : tunnel_mode = 1 の場合、省略不可)
#####
# 自拠点の ME6E-PR エントリー
#####
[me6e_pr]
macaddr               = B0:99:28:1F:59:4E
pr_prefix              = 2001:db8:ff57:73::/64
#####
# 他拠点の ME6E-PR エントリー
```

```
#####  
[me6e_pr]  
macaddr          = B0:99:28:1F:51:21  
pr_prefix        = 2001:db8:ff57:00::/64
```