

ME6E-FP 操作説明書

1.0 版 2017 年 7 月 5 日
富士通株式会社

COPYRIGHT FUJITSU LIMITED 2014 - 2017

改版履歴

版数	日付	修正者	内容
1.0	2017/7/5	玉川	初版作成

目次

1.	はじめに	4
1.1.	ME6E-FP 構成例	4
1.2.	ME6E プロトコル機能概要	5
1.3.	ME6E-FP プロトコル機能概要	6
1.4.	ME6E モジュール構成	7
2.	ME6E コマンド(me6ecli)一覧	8
3.	ARP 代理応答設定コマンド	9
3.1.	ARP 代理応答設定コマンド詳細	9
4.	Backbone 側 NDP 代理応答設定コマンド	11
4.1.	Backbone 側 NDP 代理応答設定コマンド詳細	11
5.	Stub 側 NDP 代理応答設定コマンド	13
5.1.	Stub 側 NDP 代理応答設定コマンド詳細	13
6.	ME6E-PR(prefix resolution)設定コマンド	15
6.1.	ME6E-PR の prefix 設定コマンド詳細	15
7.	ME6E デバイス追加設定コマンド	17
7.1.	ME6E デバイス追加設定コマンド詳細	17
8.	インタフェース Plane ID 設定コマンド	18
8.1.	インタフェース Plane ID 設定コマンド詳細	18
9.	PMTU 設定コマンド	20
9.1.	PMTU 設定コマンド詳細	20
10.	ME6E-FP 設定	22
10.1.	機器毎の設定	22
10.1.1.	ActiveAssist PF1000	22
10.1.2.	CentOS7.2	22
10.2.	ME6E モジュール起動設定	23
10.3.	ME6E モジュール終了設定	23
10.4.	ME6E アドレスの設定	24

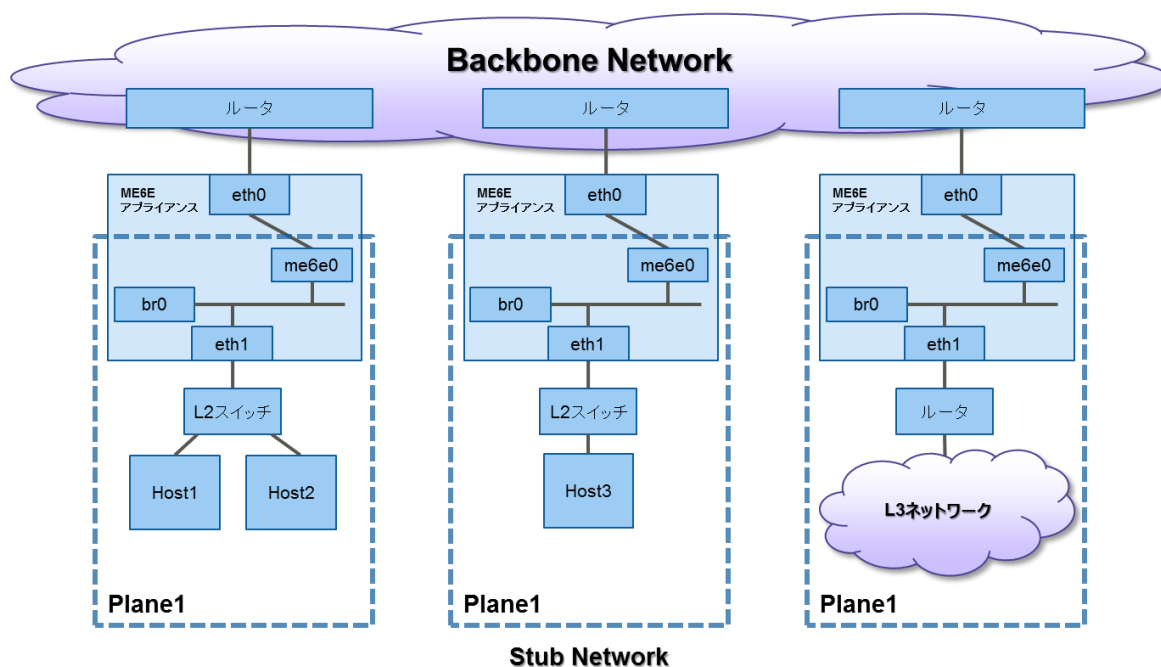
10.5.	ルーティング設定.....	25
10.6.	ME6E-FP 設定例	26
11.	付録.....	33
11.1.	Backbone 側の MTU について	33
12.	OSPF6 設定.....	33

1. はじめに

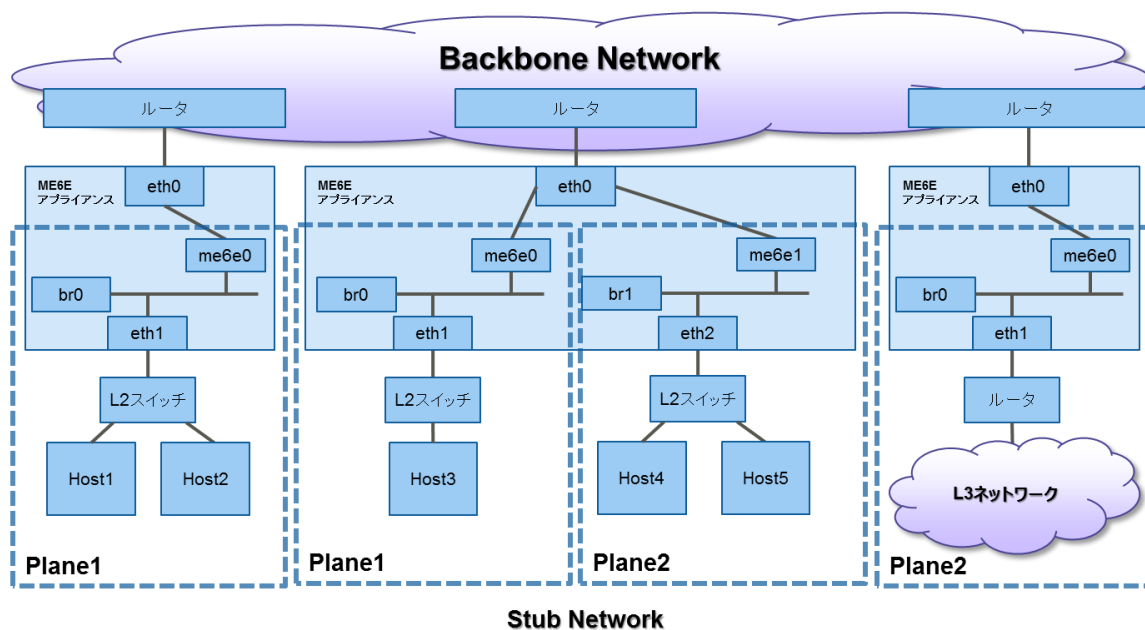
本書では ME6E-FP(Multiple Ethernet - IPv6 address mapping encapsulation – Fixed Prefix)を実行するための設定や設定に関するコマンド(me6ecli)の説明と、設定例について記載する。また、本書に記載するファイルパスは ActiveAssist PF1000 で実行した場合のものである。

1.1. ME6E-FP 構成例

ME6E-FP の構成例について記載する。以下の図の様に ME6E プロトコルは 3 拠点以上の複数拠点でも接続が可能である。また、ME6E アプライアンス配下のネットワーク(以下 Stub ネットワーク)にはホストを直接接続する以外に、L3 ネットワークも接続可能である。

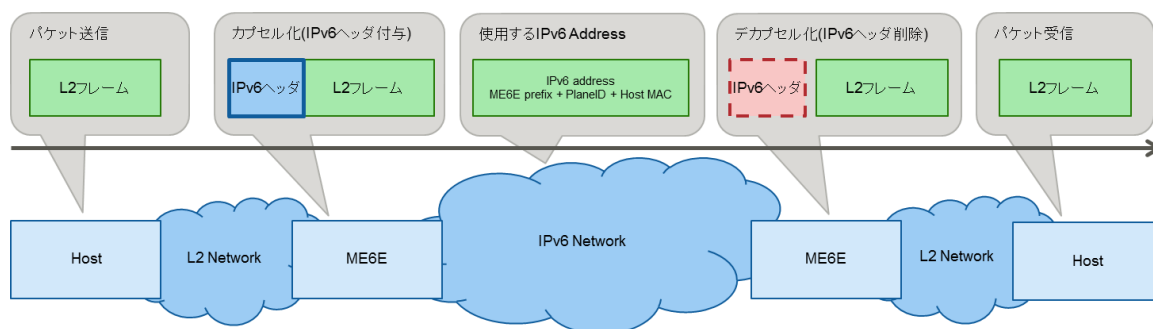


また、ME6E アプライアンスに Stub ネットワークに割り当てられる物理 NIC が複数存在する場合、以下の図の様に ME6E アプライアンス一台で複数の Plane を収容可能である。



1.2. ME6E プロトコル機能概要

ME6E プロトコルの機能概要について記載する。以下の図の様に、ホストから L2 フレームを受信した ME6E モジュールは、送信先に ME6E プロトコルで使用する IPv6 アドレスを設定した IPv6 ヘッダを付与し、IPv6 ネットワーク(以下 Backbone ネットワーク)へパケットを転送する。

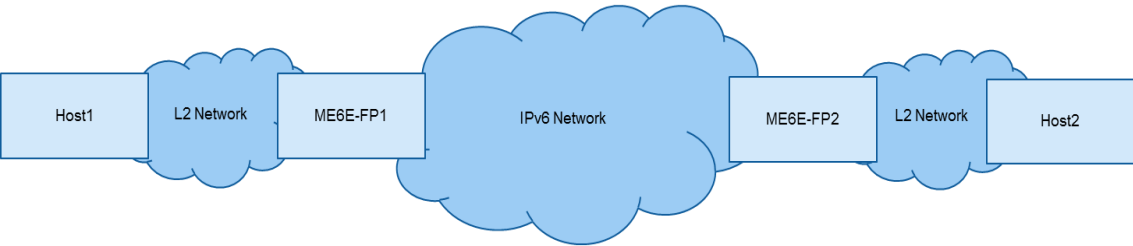


Backbone ネットワークで使用する ME6E プロトコルの IPv6 アドレスは以下の図の様な構成になっている。ME6E Prefix,PlaneID 部分の bit 長は可変である。



1.3. ME6E-FP プロトコル機能概要

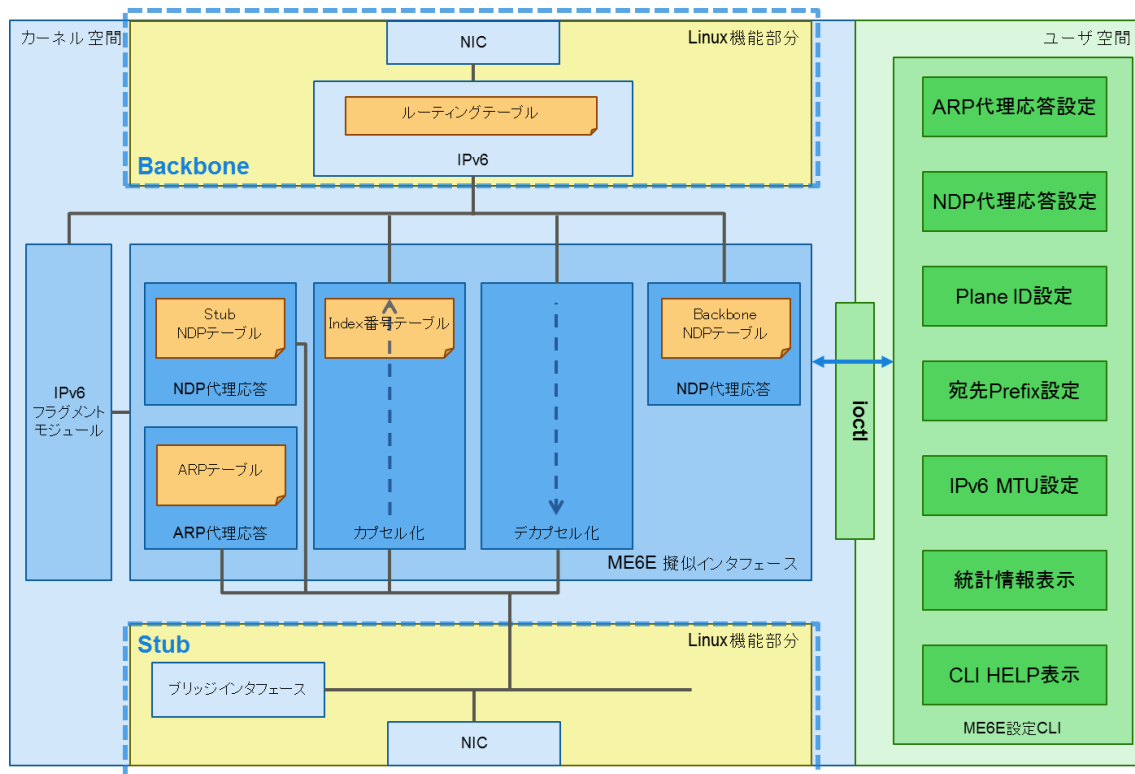
ME6E-FP プロトコルの機能概要について記載する。ME6E-FP は受信した L2 フレームを事前に設定した ME6E Prefix を宛先アドレスに設定し、IPv6 パケットにカプセル化する。宛先として使用する ME6E Prefix は事前に制御アプリを使用して ME6E モジュールに設定する。



PlaneID	MACaddr	ME6E-PR Prefix
-----	default	2001:db8:0:0:1::

1.4. ME6E モジュール構成

ME6E モジュールの構成について記載する。以下の図の様な構成となっており、ME6E モジュールは主にカプセル化、デカプセル化、ARP,NDP 代理応答の機能によって構成されている。また各機能で使用するテーブルの設定は、制御アプリ(以下 ME6E 設定 CLI)から `ioctl` を使用して ME6E モジュールに設定することができる。



2. ME6E コマンド(me6ecli)一覧

ME6E を設定するコマンドの一覧を以下に示す。

コマンド名	説明
pr	ME6E-PR(Prefix Resolusion)設定コマンド。ME6E-FP の宛先設定にも使用する。
arp_proxy	ARP 代理応答設定コマンド。
ndp_proxy backbone	Backbone 側 NDP 代理応答設定コマンド。
ndp_proxy stub	Stub 側 NDP 代理応答設定コマンド。
pmtu	経路毎の MTU 設定コマンド。
dev	ME6E デバイスの追加コマンド。
statistics	ME6E 統計情報出力コマンド。 ME6E モジュールが送受信したパケットの種類などを統計情報で確認できる。
config	ME6E の設定に関するコマンド。ファイルから設定を読み込むことができる。
iif	ME6E デバイスと同じブリッジグループにあるインタフェースの Plane ID を設定するコマンド。
help	各コマンドの説明を参照できる。
exit	me6ecli 終了コマンド。

これらのコマンドは/root/me6e/kernel/me6ecli アプリから実行する。

me6ecli を実行すると、アプリが起動し、コマンド入力モードに移行する。

起動例を以下に示す。(カレントディレクトリは、/root/me6e/kernel)

```
-sh-3.2# ./me6ecli  
me6e >
```


3. ARP 代理応答設定コマンド

本項では ME6E の ARP 代理応答設定コマンドについて記載する。

3.1. ARP 代理応答設定コマンド詳細

[コマンド名]

```
arp_proxy
```

[機能]

ME6E で ARP 代理応答する宛先アドレスの設定に使用する。引数が与えられないと、ARP 代理応答設定コマンドは現在設定されている ARP 代理応答の設定一覧を表示する。

[入力形式]

```
arp_proxy -s <ipv4addr> <MACaddr> <planeid>
arp_proxy -d <ipv4addr> <planeid>
```

[オプション]

-s

指定したアドレスをテーブルに追加する。

-d

指定したアドレスをテーブルから削除する。

[引数の説明]

- ・ <ipv4addr>

宛先 IPv4 アドレス。

- ・ <MACaddr>

指定した宛先の MAC アドレス。

- ・ <planeID>

指定した宛先の plane ID。

設定可能な値は、0-4294967295 (10 進数で指定)。

[備考]

送信したい宛先の IPv4 アドレスと MAC アドレスをセットで登録すること。

[コマンド入力例]

- 宛先アドレスをテーブルに追加する

宛先 IPv4 アドレス「192.168.1.1」、MAC アドレス「00:26:2D:06:B7:B4」、plane ID「1」、のエントリを追加する場合。

```
> arp_proxy -s 192.168.1.1 00:26:2D:06:B7:B4 1
```

- 宛先アドレスをテーブルから削除する

宛先 IPv4 アドレス「192.168.1.1」、plane ID「1」、のエントリを削除する場合。

```
> arp_proxy -d 192.168.1.1 1
```

- 設定一覧を表示する

```
> arp_proxy
```

<出力例>

PlaneID	IPv4addr	MACaddr
1	192.168.1.1	00:26:2D:06:B7:B4
2	192.168.1.2	00:26:2d:06:b8:30

4. Backbone 側 NDP 代理応答設定コマンド

本項では ME6E の Backbone 側 NDP 代理応答設定用コマンドについて記載する。

4.1. Backbone 側 NDP 代理応答設定コマンド詳細

[コマンド名]

```
ndp_proxy backbone
```

[機能]

ME6E で Backbone 側に NDP 代理応答する宛先アドレスの設定に使用する。引数が与えられないと、NDP 代理応答設定コマンドは現在設定されている NDP 代理応答の設定一覧を表示する。

[入力形式]

```
ndp_proxy backbone -s <MACaddr>  
ndp_proxy backbone -d <MACaddr>
```

[オプション]

-s

指定した宛先アドレスをテーブルに追加する。

-d

指定したアドレスをテーブルから削除する。

[引数の説明]

- ・ <MACaddr>

Backbone 側に代理応答を返す MAC アドレス。

[備考]

対向 ME6E からのパケットを受け取るための設定。自分の配下の Stub ネットワークにある物理インターフェースの MAC アドレスを登録すること。

[コマンド入力例]

- 宛先アドレスをテーブルに追加する

MAC アドレス「00:26:2D:06:B7:B4」を設定する場合。

```
> ndp_proxy backbone -s 00:26:2D:06:B7:B4
```

- 宛先アドレスをテーブルから削除する

MAC アドレス「00:26:2D:06:B7:B4」を削除する場合。

```
> ndp_proxy backbone -d 00:26:2D:06:B7:B4
```

- 設定一覧を表示する

```
> ndp_proxy backbone
```

<出力例>

MACaddr

00:26:2D:06:B7:B4

00:26:2d:06:b8:30

5. Stub 側 NDP 代理応答設定コマンド

本項では ME6E の Stub 側 NDP 代理応答設定用コマンドについて記載する。

5.1. Stub 側 NDP 代理応答設定コマンド詳細

[コマンド名]

```
ndp_proxy stub
```

[機能]

ME6E で Stub 側に NDP 代理応答する宛先アドレスの設定に使用する。引数が与えられないと、Stub 側 NDP 代理応答設定コマンドは現在設定されている Stub 側 NDP 代理応答の設定一覧を表示する。

[入力形式]

```
ndp_proxy stub -s <ipv6addr> <macaddr> <planeid>
ndp_proxy stub -d <ipv6addr> <planeid>
```

[オプション]

-s

指定した宛先アドレスをテーブルに追加する。

-d

指定したアドレスをテーブルから削除する。

[引数の説明]

- ・ <ipv6addr>
宛先 IPv6 アドレス。
- ・ <MACaddr>
指定した宛先の MAC アドレス。
- ・ <planeid>
宛先の Plane ID

[備考]

送信したい宛先の IPv6 アドレスと MAC アドレス、Plane ID をセットで登録すること。

[コマンド入力例]

○宛先アドレスをテーブルに追加する。

宛先 IPv6 アドレス「2001:db8:cafe::1」、MAC アドレス「00:26:2D:06:B7:B4」、Plane ID「1」を設定する場合。

```
> ndp_proxy stub -s 2001:db8:cafe::1 00:26:2D:06:B7:B4 1
```

○宛先アドレスをテーブルから削除する。

宛先 IPv6 アドレス「2001:db8:cafe::1」、Plane ID「1」を削除する場合。

```
> ndp_proxy stub -d 2001:db8:cafe::1 1
```

○設定一覧を表示する

```
> ndp_proxy stub
```

<出力例>

PlaneID	MACaddr	IPv6addr
1	00:26:2D:06:B7:B4	2001:db8:cafe::1
2	00:26:2d:06:b8:30	2001:db8:cafe::2

6. ME6E-PR(prefix resolution)設定コマンド

本項では ME6E-PR の prefix 設定用コマンドについて記載する。

6.1. ME6E-PR の prefix 設定コマンド詳細

[コマンド名]

pr

[機能]

ME6E-PR では、宛先 MAC アドレス毎に設定された ME6E の prefix を使用して IPv6 パケットを生成する。本コマンドでは MAC アドレスに紐づける ME6E の prefix を設定する。

[入力形式]

<pre>pr -s pr-prefix <macaddr> <me6e-prefix and planeid> <planeid> pr -s default <me6e-prefix and planeid> pr -d pr-prefix <macaddr> <planeid> pr -d default pr -f <filepath> File format: macaddr,me6e-prefix and planeid,planeid</pre>
--

[オプション]

-s pr-prefix

指定した MAC アドレスの ME6E prefix と PlaneID をテーブルに追加する。

-s default

テーブルに登録されていない MAC アドレスを受信したときに付与する ME6E prefix と PlaneID を設定する。(ME6E-FP の宛先アドレス設定時に使用する)

-d pr-prefix

指定した MAC アドレスの ME6E prefix を削除する。

-d default

デフォルト prefix を削除する。

-f

設定ファイルに記載されている設定を一括設定する。

[引数の説明]

- ・ <macaddr>
指定した宛先の MAC アドレス。
- ・ <me6e-prefix and planeID>
PlaneID を含む ME6E prefix。/80 の IPv6 アドレス。
- ・ <planeid>
設定するネットワークの Plane ID。
- ・ <filepath>
設定ファイルのファイルパス。

[コマンド入力例]

- 宛先アドレスをテーブルに追加する。

MAC アドレス「00:26:2d:06:b8:30」の宛先に ME6E prefix「2001:db8::」、Plane ID「6553601」を設定する場合。

```
> pr -s pr-prefix 00:26:2d:06:b8:30 2001:db8:0:64:1:: 6553601
```

- 宛先アドレスをテーブルから削除する。

MAC アドレス「00:26:2d:06:b8:30」の宛先 ME6E prefix「2001:db8::」、Plane ID「6553601」を削除する場合。

```
> pr -d pr-prefix 00:26:2d:06:b8:30 6553601
```

- 設定一覧を表示する

```
> pr
```

<出力例>

PlaneID	MACaddr	ME6E-PR Prefix
6553601	00:26:2d:06:b8:30	2001:db8:0:64:1:26:2d06:b830

7. ME6E デバイス追加設定コマンド

本項では ME6E デバイス追加設定コマンドについて記載する。

7.1. ME6E デバイス追加設定コマンド詳細

[コマンド名]

dev

[機能]

ME6E を MultiPlane 構成にする場合に使用する。MultiPlane 構成にする場合は plane 毎に異なる ME6E デバイスを作成、ブリッジの設定をする。デバイス名は me6ex(x は数値) で作成され、x の値はデバイスの追加ごとにインクリメントされる

[入力形式]

dev -s

[オプション]

-s

ME6E デバイスをシステム上に追加する。

[引数の説明]

なし。

[備考]

デバイスを削除する場合は ME6E モジュールを rmmod すること。

[コマンド入力例]

○ME6E デバイスをシステム上に追加する。

> dev -s

8. インタフェース Plane ID 設定コマンド

本項ではインタフェース Plane ID 設定コマンドについて記載する。

8.1. インタフェース Plane ID 設定コマンド詳細

[コマンド名]

```
iif
```

[機能]

ME6E デバイスと同じブリッジグループにあるインタフェースの Plane ID を設定する。

[入力形式]

```
iif -s <if_index> <planeid>  
iif -d <if_index>
```

[オプション]

-s

指定したインデックス番号のインタフェースに Plane ID を設定する

-d

指定したインデックス番号のインタフェースの Plane ID 設定を削除する。

[引数の説明]

- ・ <if_index>
インタフェースのインデックス番号
- ・ <planeid>
インタフェースに設定する Plane ID

[コマンド入力例]

○Plane ID を設定する

```
> iif -s 2 1
```

○Plane ID の設定を削除する

```
> iif -d 2
```

○インタフェース番号と Plane ID 設定状況を表示

<出力例>

IIF PlaneID	

2	1
4	2
index	name

1	lo
2	enp96s5
3	enp0s25
4	enp32s0
5	enp0s29f7u5
6	virbr0
7	virbr0-nic
9	me6e0

9. PMTU 設定コマンド

本項では ME6E の PMTU 設定コマンドについて記載する。

9.1. PMTU 設定コマンド詳細

[コマンド名]

pmtu

[機能]

ME6E で経路毎の MTU 設定に使用する。また、pcket too big パケットを Backbone 側から ME6E で受信した際に、動的に MTU を設定する。引数が与えられないと、PMTU 設定コマンドは現在設定されている経路毎の PMTU 設定一覧を表示する。

[入力形式]

pmtu -s <ipv6addr> <MTU> pmtu -d <ipv6addr> pmtu -t <time>
--

[オプション]

-s

指定した経路の MTU をテーブルに追加する。

-d

指定した経路の MTU をテーブルから削除する。

-t

動的設定 MTU が削除される時間を設定する。

[引数の説明]

- ・ <ipv6addr>
宛先 IPv6 アドレス。
- ・ <MTU>
設定する MTU。
- ・ <time>
動的設定 MTU が削除されるまでの時間(sec)

[コマンド入力例]

○指定経路のMTUを設定する

```
> pmtu -s 2001:db8:2:0:1:1234:1234:1234 1300
```

○指定経路のMTUを削除する

```
> pmtu -d 2001:db8:2:0:1:1234:1234:1234
```

○設定一覧を表示する

```
> pmtu
```

<出力例>

```
me6e >pmtu
```

Address	MTU	Life(sec) : initial value = 600
-----	----	-----
2001:db8:2:0:1:1234:1234:1234	1300	static
2001:db8:2:0:1:2222:3333:4444	1406	141
Total entries : 2		

10. ME6E-FP 設定

ME6E-FP の設定について記載する。

10.1. 機器毎の設定

ActiveAssist PF1000、もしくは CentOS7.2 で ME6E アプライアンスを構成する場合の機器毎の設定について記載する。

10.1.1. ActiveAssist PF1000

ActiveAssist PF1000 での起動、ログイン、設定について記載する。

ActiveAssist PF1000 とコンソール端末を RS-232C ケーブル(クロス)で接続し起動する。ログインプロンプトが表示されたら、ユーザ名、パスワードを入力し、ログインする(ユーザ名、パスワードについては ActiveAssist PF1000 取扱説明書参照)。ログイン後は、Linux 端末と同様に操作できる。

ME6E に必要なファイルは以下のディレクトリに配置されている。

/root/me6e/kernel

ME6E の設定に必要なファイルの説明を以下に示す。

ファイル名	説明
me6e.ko	ME6E カーネルモジュール。
ex_ipv6_fragment.ko	ME6E の動作に必要なフラグメント関連の拡張カーネルモジュール。(GPL ライセンス)
me6cli	ME6E の設定をするアプリ。
sample-config	ME6E 設定用コマンドアプリの入力コマンド例
sample-init	ME6E 動作環境用 Linux 設定例

10.1.2. CentOS7.2

CentOS7.2 での設定について記載する。CentOS7.2 では以下の sysctl の値を設定する。

設定項目	初期値	変更後
IPv6 forwarding	0	1

10.2. ME6E モジュール起動設定

ME6E モジュール起動時に実行するコマンドについて記載する。

- カレントディレクトリを ME6E 設定ファイル配置場所に移動する。

```
-sh-3.2# cd /root/me6e/kernel
```

- フラグメントモジュール(ex_ipv6_fragment.ko)を insmod する。

```
-sh-3.2# insmod ex_ipv6_fragment.ko
```

- ME6E モジュールを insmod する。

```
-sh-3.2# insmod me6e.ko
```

※事前に ex_ipv6_fragment.ko を insmod しておかないとエラーになる。

10.3. ME6E モジュール終了設定

ME6E モジュール終了時に実行するコマンドについて記載する。

- ME6E モジュールを rmmod する。

```
-sh-3.2# rmmod me6e
```

- フラグメントモジュールを rmmod する。

```
-sh-3.2# rmmod ex_ipv6_fragment.ko
```

- 作成したブリッジデバイスを削除する。

```
-sh-3.2# brctl delbr br0 ←-----★作成したブリッジ名を指定する
```

10.4. ME6E アドレスの設定

ME6E モジュールを insmod した際に作成される me6e 疑似インタフェース (me6e0) に設定するアドレスについて記載する。

ME6E 疑似インタフェースには、以下の形式のアドレスを設定する。

例) <u>2001:db8:1::1:</u> <u>aabb:ccdd:eeff</u> <u>/80</u>			
①	②	③	④

- ① ME6E の prefix (48bit)。通信相手と一致するように設定。
- ② plane ID(32bit)。通信相手と一致するように設定。
- ③ MAC アドレス領域(48bit)。Stub からのパケットのカプセル化の際、MAC アドレスが入る領域。MAC アドレスと重ならない任意の値を設定。
- ④ マスク値。Prefix+planeID 分の/80 を設定。

(参考) ME6E でカプセル化したパケットのアドレスは以下ようになる。

・宛先アドレス

ME6E-prefix(48bit)	plane ID(32bit)	宛先 MAC address(32bit)
--------------------	-----------------	-----------------------

・送信元アドレス

ME6E-prefix(48bit)	plane ID(32bit)	送信元 MAC address(32bit)
--------------------	-----------------	------------------------

※PlaneID 部は可変であるため、bit 長が変更されることがある。

10.5. ルーティング設定

カプセル化後のアドレスはME6E 疑似インタフェースと同じプレフィックスを持つため、そのままだと ME6E 疑似インタフェースに転送されてしまい Backbone 側に出ていかない。そこで、Backbone 側に転送するような設定を追加する。

以下、ME6E 疑似インタフェース（2001:db8:1::1:11:0:11/80）で、MAC アドレス AA:BB:DD:EE:FF 宛てのパケットをカプセル化する場合の例について説明する。

- 送信先のカプセル化後アドレスを/128 で Backbone 側に転送するように設定。

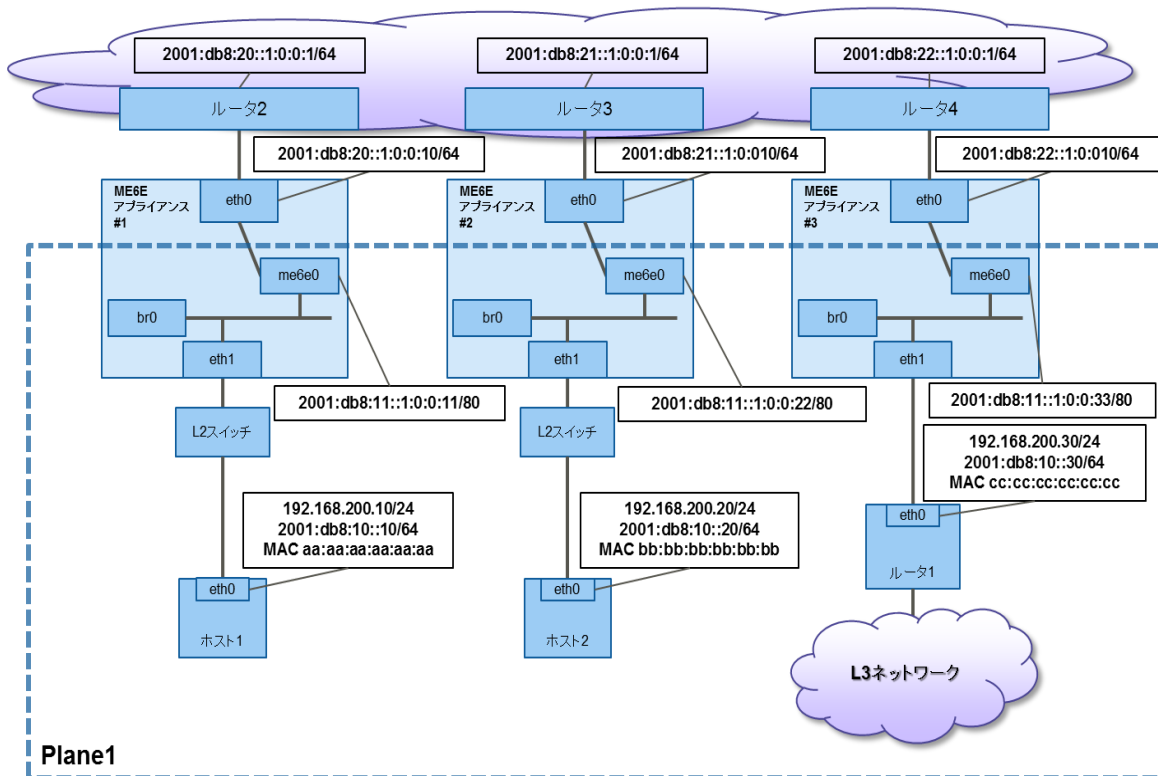
```
-sh-3.2# route -A inet6 add 2001:db8:1::1:aabb:ccdd:eeff/128 gw [backbone のアドレス]
```

（参考）

Stub 側からのパケットは、ブリッジにより ME6E 疑似インタフェースに転送されるので、ルーティングは不要。カプセル化後のパケットは、送信元の Stub 側物理インタフェースから再送信される。

10.6. ME6E-FP 設定例

ネットワーク構成の例を用いて ME6E-FP の設定方法を記載する。以下に ME6E-FP の構成例を示す。



図中のホスト 1、ホスト 2、ルータ 1 配下の L3 ネットワーク間で通信する場合、ME6E アプライアンス#1,#2,#3 に必要な設定を以下に示す。

◆ ME6E アプライアンス#1 の設定

- ① ME6E モジュール配置ディレクトリに移動し、ME6E モジュールを insmod する。

```
cd /root/me6e/kernel
insmod ex_ipv6_fragment.ko
insmod me6e.ko
```

- ② ブリッジインタフェースを作成し、Stub 側の物理インタフェース(br0)、ME6E 疑似インタフェース(me6e0)をブリッジグループに追加する。

```
brctl addbr br0
ifconfig br0 up
ifconfig eth1 up 0.0.0.0
brctl addif br0 eth1
brctl addif br0 me6e0
```

- ③ ME6E 疑似インタフェース(me6e0)にアドレスを設定する。

```
ifconfig me6e0 up
ifconfig me6e0 add 2001:db8:11::1:0:0:11/80
```

- ④ Backbone 側の物理インタフェース(eth0)のアドレス設定、カプセル化パケット (MAC: bb:bb:bb:bb:bb:bb, cc:cc:cc:cc:cc:cc 宛て)のルーティング設定をする。

```
ifconfig eth0 up
ifconfig eth0 add 2001:db8:20::1:0:0:10/64
route -A inet6 add 2001:db8:11::1:bbbb:bbbb:bbbb/128 gw 2001:db8:20::1:0:0:1 dev eth0
route -A inet6 add 2001:db8:11::1:cccc:cccc:cccc/128 gw 2001:db8:20::1:0:0:1 dev eth0
```

- ⑤ me6ecli で ARP 代理応答の設定をする。

```
me6e >arp_proxy -s 192.168.200.20 bb:bb:bb:bb:bb:bb 1
me6e >arp_proxy -s 192.168.200.30 cc:cc:cc:cc:cc:cc 1
```

宛先(ホスト2,ルータ1)のIPv4アドレス、MACアドレス、PlaneIDを登録。(PlaneIDは1)

- ⑥ me6ecli で Stub 側 NDP 代理応答の設定をする。

```
me6e >ndp_proxy stub -s 2001:db8:10::20 bb:bb:bb:bb:bb:bb 1
me6e >ndp_proxy stub -s 2001:db8:10::30 cc:cc:cc:cc:cc:cc 1
```

宛先(ホスト2,ルータ1)のIPv6アドレス、MACアドレス、PlaneIDを登録。(PlaneIDは1)

- ⑦ me6ecli で Stub 側 IF が所属する PlaneID を指定する。

```
me6e >iif
index    name
-----
1        lo
2        eth0
3        eth1
4        eth2
5        tunl0
6        eth2.2@eth2
7        eth2.3@eth2
8        eth2.4@eth2
9        eth2.5@eth2
10       ETHER2
11       ETHER3
12       ETHER4
13       ETHER5
14       IMP0
15       me6e0
16       br0
me6e >iif -s 3 1
```

- ⑧ me6ecli で宛先毎の ME6E Prefix の設定をする。(ME6E+PlaneID の/80 のアドレスを設定する)

```
me6e >pr -s default 2001:db8:11:0:1::
```

設定した prefix でカプセル化し、ME6E パケットを生成する。この例では、ホスト 1 からホスト 2 に対して通信する際に、以下の様に ME6E プロトコルで使用するアドレスを生成してパケットを転送する。

```
2001:db8:11:0:1:bbbb:bbbb:bbbb
```

- ⑨ me6ecli で Backbone 側 NDP 代理応答の設定をする。

```
me6e >ndp_proxy backbone -s aa:aa:aa:aa:aa:aa
```

配下（ホスト 1）の MAC アドレスを登録。

◆ ME6E アプライアンス#2 の設定

- ① ME6E モジュール配置ディレクトリに移動し、ME6E モジュールを insmod する。

```
cd /root/me6e/kernel
insmod ex_ipv6_fragment.ko
insmod me6e.ko
```

- ② ブリッジインタフェースを作成し、Stub 側の物理インタフェース(br0)、ME6E 疑似インタフェース(me6e0)をブリッジグループに追加する。

```
brctl addbr br0
ifconfig br0 up
ifconfig eth1 up 0.0.0.0
brctl addif br0 eth1
brctl addif br0 me6e0
```

- ③ ME6E 疑似インタフェース(me6e0)にアドレスを設定する。

```
ifconfig me6e0 up
ifconfig me6e0 add 2001:db8:11::1:0:0:22/80
```

- ④ Backbone 側の物理インタフェース(eth0)のアドレス設定、カプセル化パケット (MAC: aa:aa:aa:aa:aa:aa, cc:cc:cc:cc:cc:cc 宛て)のルーティング設定をする。

```
ifconfig eth0 up
ifconfig eth0 add 2001:db8:21::1:0:0:10/64
route -A inet6 add 2001:db8:11::1:aaaa:aaaa:aaaa/128 gw 2001:db8:21::1:0:0:1 dev eth0
route -A inet6 add 2001:db8:11::1:cccc:cccc:cccc/128 gw 2001:db8:21::1:0:0:1 dev eth0
```

- ⑤ me6ecli で ARP 代理応答の設定をする。

```
me6e >arp_proxy -s 192.168.200.10 aa:aa:aa:aa:aa:aa 1
me6e >arp_proxy -s 192.168.200.30 cc:cc:cc:cc:cc:cc 1
```

宛先(ホスト1,ルータ1)のIPv4アドレス、MACアドレス、PlaneIDを登録。(PlaneIDは1)

- ⑥ me6ecli で Stub 側 NDP 代理応答の設定をする。

```
me6e >ndp_proxy stub -s 2001:db8:10::10 aa:aa:aa:aa:aa:aa 1
me6e >ndp_proxy stub -s 2001:db8:10::30 cc:cc:cc:cc:cc:cc 1
```

宛先（ホスト 2, ルータ 1）の IPv6 アドレス、MAC アドレス、Plane ID を登録。
（PlaneID は 1）

- ⑦ me6ecli で Stub 側 IF が所属する PlaneID を指定する。

```
me6e >iif
index      name
-----
1          lo
2          eth0
3          eth1
4          eth2
5          tunl0
6          eth2.2@eth2
7          eth2.3@eth2
8          eth2.4@eth2
9          eth2.5@eth2
10         ETHER2
11         ETHER3
12         ETHER4
13         ETHER5
14         IMP0
15         me6e0
16         br0
me6e >iif -s 3 1
```

- ⑧ me6ecli で宛先毎の ME6E Prefix の設定をする。(ME6E+PlaneID の/80 のアドレスを設定する)

```
me6e >pr -s default 2001:db8:11:0:1::
```

設定した prefix でカプセル化し、ME6E パケットを生成する。

- ⑨ me6ecli で Backbone 側 NDP 代理応答の設定をする。

```
me6e >ndp_proxy backbone -s bb:bb:bb:bb:bb:bb
```

配下 (ホスト 2) の MAC アドレスを登録。

◆ ME6E アプライアンス#3 の設定

- ① ME6E モジュール配置ディレクトリに移動し、ME6E モジュールを insmod する。

```
cd /root/me6e/kernel
insmod ex_ipv6_fragment.ko
insmod me6e.ko
```

- ② ブリッジインタフェースを作成し、Stub 側の物理インタフェース(br0)、ME6E 疑似インタフェース(me6e0)をブリッジグループに追加する。

```
brctl addbr br0
ifconfig br0 up
ifconfig eth1 up 0.0.0.0
brctl addif br0 eth1
brctl addif br0 me6e0
```

- ③ ME6E 疑似インタフェース(me6e0)にアドレスを設定する。

```
ifconfig me6e0 up
ifconfig me6e0 add 2001:db8:13::1:0:0:13/80
```

- ④ Backbone 側の物理インタフェース(eth0)のアドレス設定、カプセル化パケット (MAC: aa:aa:aa:aa:aa:aa, bb:bb:bb:bb:bb:bb 宛て)のルーティング設定をする。

```
ifconfig eth0 up
ifconfig eth0 add 2001:db8:13::1:0:0:10/64
route -A inet6 add 2001:db8:11::1:aaaa:aaaa:aaaa/128 gw 2001:db8:13::1:0:0:1 dev eth0
route -A inet6 add 2001:db8:12::1:bbbb:bbbb:bbbb/128 gw 2001:db8:13::1:0:0:1 dev eth0
```

- ⑤ me6ecli で ARP 代理応答の設定をする。

```
me6e >arp_proxy -s 192.168.200.10 aa:aa:aa:aa:aa:aa 1
me6e >arp_proxy -s 192.168.200.20 bb:bb:bb:bb:bb:bb 1
```

宛先(ホスト1, ルータ1)のIPv4アドレス、MACアドレス、PlaneIDを登録。(PlaneIDは1)

- ⑥ me6ecli で Stub 側 NDP 代理応答の設定をする。

```
me6e >ndp_proxy stub -s 2001:db8:10::10 aa:aa:aa:aa:aa:aa 1
me6e >ndp_proxy stub -s 2001:db8:10::20 bb:bb:bb:bb:bb:bb 1
```

宛先（ホスト 2, ルータ 1）の IPv6 アドレス、MAC アドレス、Plane ID を登録。
（PlaneID は 1）

- ⑦ me6ecli で Stub 側 IF が所属する PlaneID を指定する。

```
me6e >iif
index      name
-----
1          lo
2          eth0
3          eth1
4          eth2
5          tunl0
6          eth2.2@eth2
7          eth2.3@eth2
8          eth2.4@eth2
9          eth2.5@eth2
10         ETHER2
11         ETHER3
12         ETHER4
13         ETHER5
14         IMP0
15         me6e0
16         br0
me6e >iif -s 3 1
```

- ⑧ me6ecli で宛先毎の ME6E Prefix の設定をする。(ME6E+PlaneID の/80 のアドレスを設定する)

```
me6e >pr -s default 2001:db8:11:0:1::
```

設定した prefix でカプセル化し、ME6E パケットを生成する。

- ⑨ me6ecli で Backbone 側 NDP 代理応答の設定をする。

```
me6e >ndp_proxy backbone -s cc:cc:cc:cc:cc:cc
```

配下（ルータ 1）の MAC アドレスを登録。

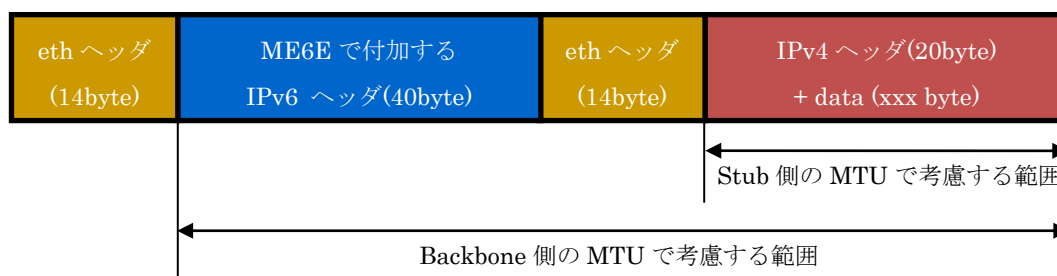
11. 付録

11.1. Backbone 側の MTU について

ME6E では元のパケットの eth ヘッダも含めてカプセル化し、IPv6 ヘッダを新しく付加するため、カプセル化後のパケットのサイズは IPv6 ヘッダ(40byte)+eth ヘッダ(14byte)分増加する。

ME6E でフラグメントを発生させたくない場合は、Backbone 側の MTU（ME6E とブリッジインタフェースを含む）を Stub 側の MTU より 54byte 以上大きく設定すること。

例) IPv4 パケットのカプセル化後のサイズ



12. OSPF6 設定

OSPF6 の設定について記載する。OSPF で経路広告をする際は必ずホストルートの宛先を ME6E デバイスにする。ME6E デバイス以外に設定すると、誤ったルートが登録されてしまい、先頭数パケットをロストするなど不具合が生じる。

以下に quagga で OSPF6 を設定する際の設定例を記載する。本例では eth0 が backbone 側、eth1 が stub 側となっている。

```
localhost# show running-config
Building configuration...

Current configuration:
!
hostname Router
hostname ospf6d@plant
log stdout
!
service advanced-vty
```

```
!  
debug ospf6 lsa unknown  
debug ospf6 neighbor state  
!  
password zebra  
enable password zebra  
!  
interface br0  
    ipv6 nd suppress-ra  
!  
interface eth0  
    ipv6 nd suppress-ra  
    ipv6 ospf6 cost 1  
    ipv6 ospf6 dead-interval 40  
    ipv6 ospf6 hello-interval 10  
    ipv6 ospf6 instance-id 0  
    ipv6 ospf6 priority 76  
    ipv6 ospf6 retransmit-interval 5  
    ipv6 ospf6 transmit-delay 1  
!  
interface eth1  
    ipv6 nd suppress-ra  
    ipv6 ospf6 cost 1  
    ipv6 ospf6 dead-interval 40  
    ipv6 ospf6 hello-interval 10  
    ipv6 ospf6 instance-id 0  
    ipv6 ospf6 priority 76  
    ipv6 ospf6 retransmit-interval 5  
    ipv6 ospf6 transmit-delay 1  
!  
interface eth2  
    ipv6 nd suppress-ra  
!  
interface lo  
!  
interface me6e0  
    ipv6 nd suppress-ra
```

```
end
localhost# show running-config
Building configuration...

Current configuration:
!
hostname Router
hostname ospf6d@plant
log stdout
!
service advanced-vty
!
debug ospf6 lsa unknown
debug ospf6 neighbor state
!
password zebra
enable password zebra
!
interface br0
  ipv6 nd suppress-ra
!
interface eth0
  ipv6 nd suppress-ra
  ipv6 ospf6 cost 1
  ipv6 ospf6 dead-interval 40
  ipv6 ospf6 hello-interval 10
  ipv6 ospf6 instance-id 0
  ipv6 ospf6 priority 76
  ipv6 ospf6 retransmit-interval 5
  ipv6 ospf6 transmit-delay 1
!
interface eth1
  ipv6 nd suppress-ra
!
interface eth2
  ipv6 nd suppress-ra
!
```

```
interface lo
!
interface me6e0
  ipv6 nd suppress-ra
!
interface tunl0
  ipv6 nd suppress-ra
!
router ospf6
  router-id 255.1.1.3
  redistribute kernel
  redistribute connected
  redistribute static
  interface eth0 area 0.0.0.0
  interface eth1 area 0.0.0.0
!
ipv6 route 2001:db8:0:64:1:26:2d06:b83e/128 me6e0
!
ipv6 access-list access6 permit 3ffe:501::/32
ipv6 access-list access6 permit 2001:200::/48
ipv6 access-list access6 permit ::1/128
!
ip forwarding
ipv6 forwarding
!
line vty
!
end
```

以上