

M46E-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 2013-2017

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

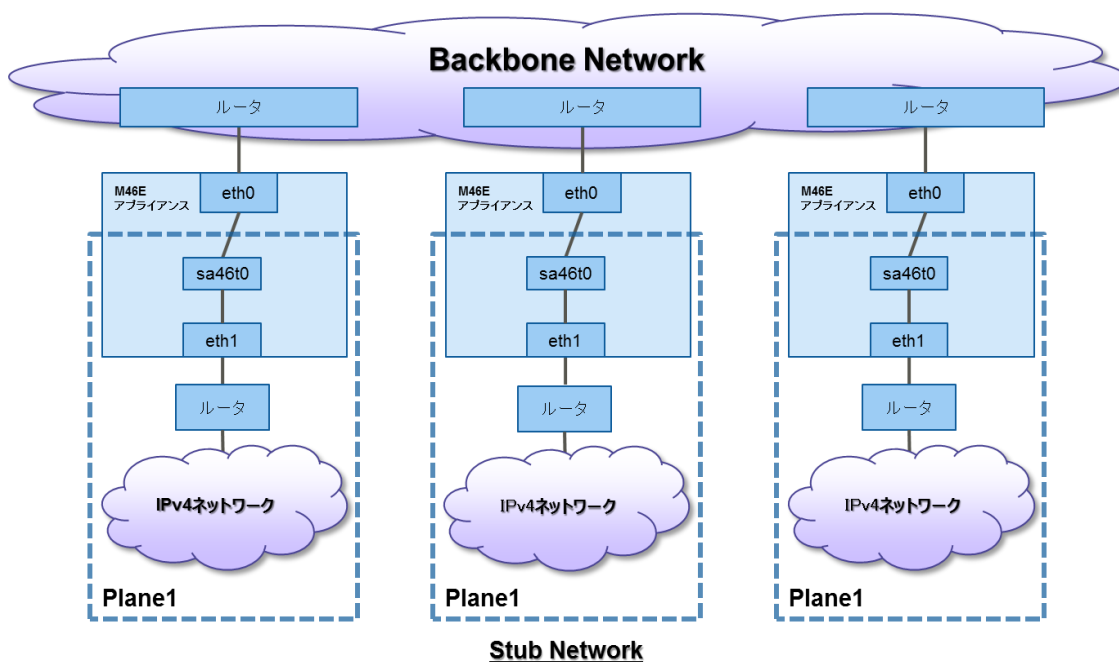
1.	はじめに	3
1.1.	M46E-FP 構成例	3
1.2.	M46E プロトコル機能概要.....	4
1.3.	M46E-FP プロトコル機能概要	5
1.4.	M46E モジュール構成.....	5
2.	M46E コマンド(m46ecli)一覧	6
3.	M46E-PR(prefix resolution)設定コマンド	7
3.1.	M46E-PR コマンド詳細	7
4.	PMTU 設定コマンド.....	10
4.1.	PMTU コマンド詳細	10
5.	Network NameSpace 設定コマンド	13
5.1.	NS(Network NameSpace)コマンド詳細	13
6.	M46E-FP 設定	17
6.1.	機器毎の設定.....	17
6.1.1.	ActiveAssist PF1000	17
6.1.2.	CentOS7.2	17
6.2.	M46E モジュール起動設定.....	18
6.3.	M46E モジュール終了設定.....	18
6.4.	インタフェース、ルーティングテーブルの設定の考え方	19
6.5.	M46E-FP 設定例	20
7.	付録	23
7.1.	M46E の設定ファイル出力.....	23
7.2.	ファイルから M46E の設定読み込み	24
7.3.	sample-init	25
7.4.	M46E の自動起動	26

1. はじめに

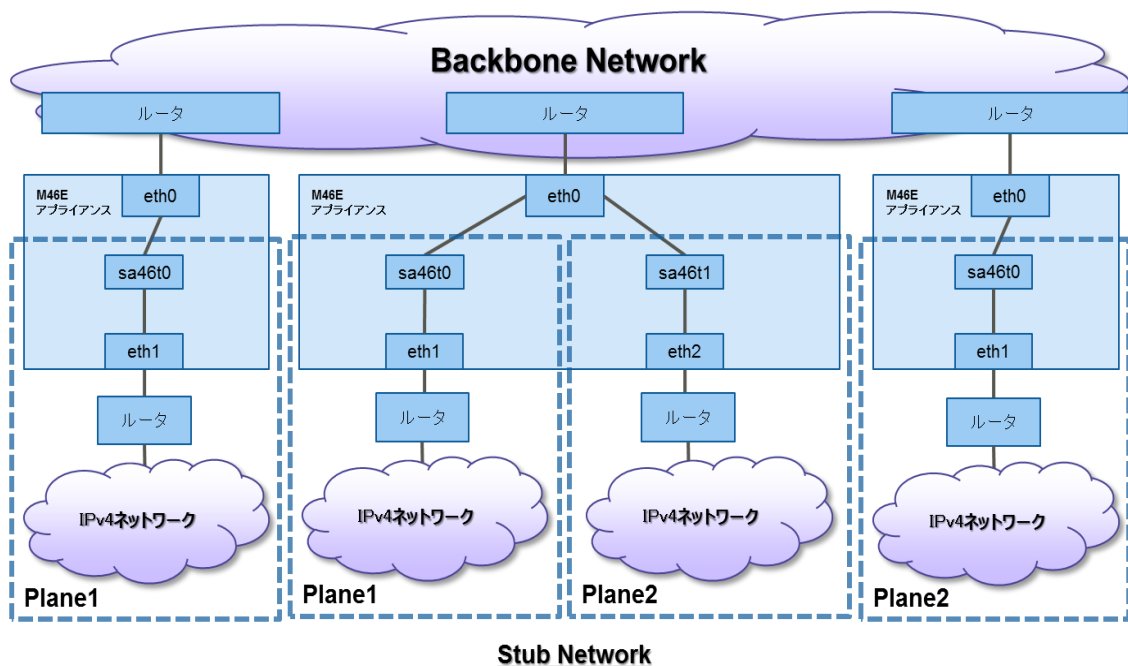
本書では M46E-FP(Multiple IPv4 - IPv6 address mapping encapsulation - Fixed Prefix)を実行するための設定や設定に関するコマンド(m46ecli)の説明と、設定例について記載する。また、本書に記載するファイルパスは ActiveAssist PF1000 で実行した場合のものである。

1.1. M46E-FP 構成例

M46E-FP の構成例について記載する。以下の図の様に M46E 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。

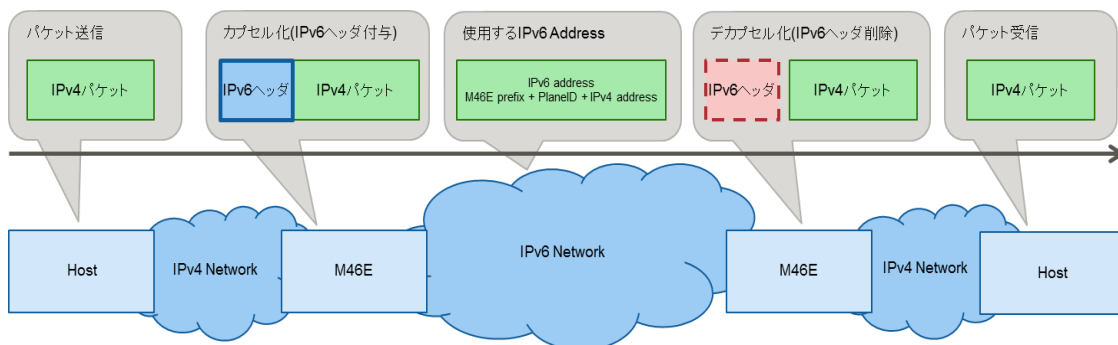


また、M46E アプライアンスに Stub ネットワークに割り当てられる物理 NIC が複数存在する場合、以下の図の様に M46E アプライアンス一台で複数の Plane を収容可能である。



1.2. M46E プロトコル機能概要

M46E 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 IPv4 パケットを受信した M46E モジュールは、送信先に M46E 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。

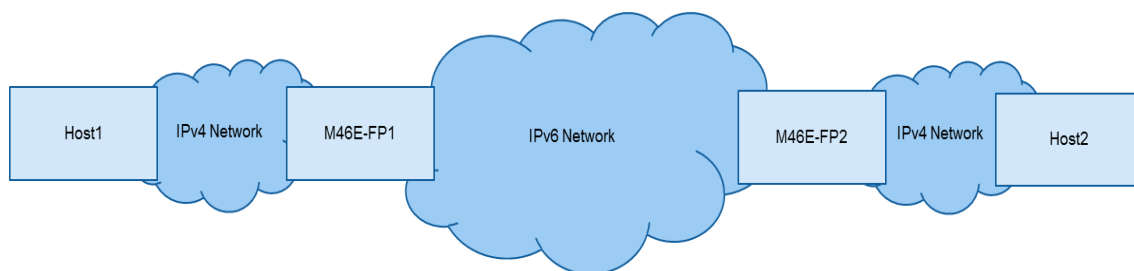


Backbone ネットワークで使用する M46E プロトコルの IPv6 アドレスは以下の図の様な構成になっている。M46E Prefix, PlaneID 部分の bit 長は可変である。

M46E Prefix	Plane ID	IPv4 address(32bit)
-------------	----------	---------------------

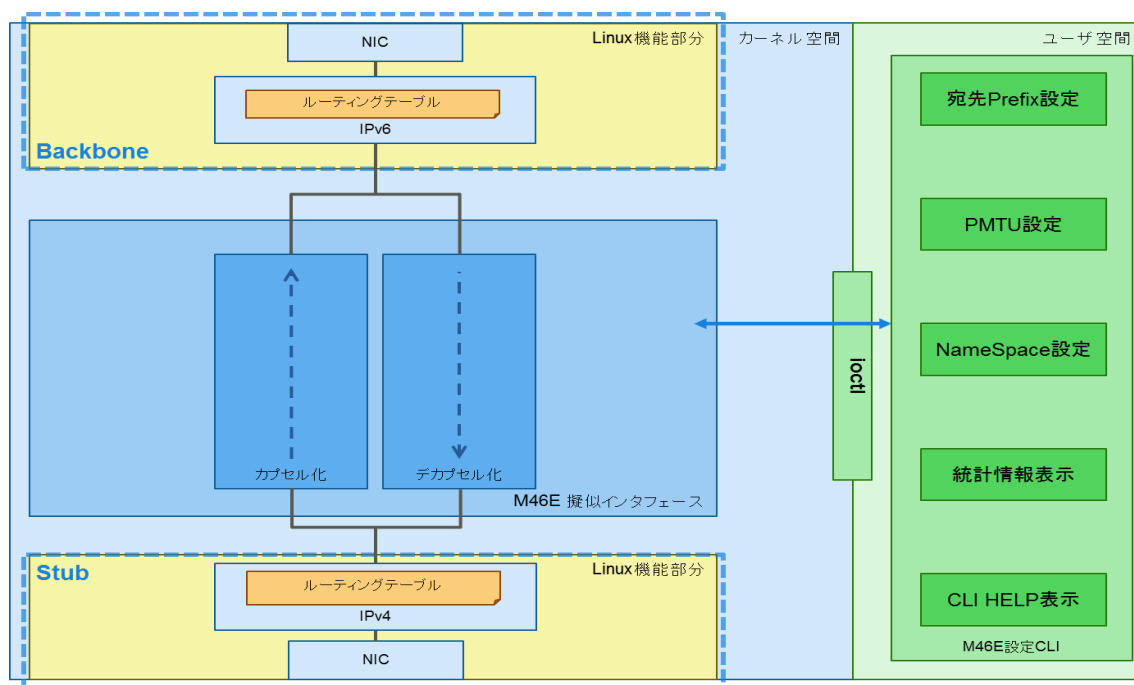
1.3. M46E-FP プロトコル機能概要

M46E-FP プロトコルの機能概要について記載する。M46E-FP は受信した IPv4 パケットを事前に設定した M46E 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 M46E Prefix は事前に制御アプリを使用して M46E モジュールに設定する。



1.4. M46E モジュール構成

M46E 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、M46E モジュールは主にカプセル化、デカプセル化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 M46E 設定 CLI)から ioctl を使用して M46E モジュールに設定することができる。



2. M46E コマンド(m46ecli)一覧

M46E 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	M46E-PR(Prefix Resolution)の設定用コマンド。ME6E-FP の宛先設定にも使用する。
pmtu	PMTU の設定用コマンド。静的な PMTU を指定できる。
ns	Network NameSpace 設定用コマンド。コマンドを実行した M46E アプリケーションに Network NameSpace を追加、設定するために使用する。
statistics	M46E 統計情報出力コマンド。M46E 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	M46E の設定に関するコマンド。現在の設定をファイルに出力したり、ファイルから設定を読み込むことができる。
show	M46E の設定や、M46E に関する linux の設定を出力するコマンド。
help	PR、PMTU、NS、CONFIG、SHOW の各コマンドの説明を参照できる。
exit	m46ecli 終了コマンド。

これらのコマンドは/root/m46e/kernel/m46ecli アプリから実行する。

m46ecli 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/m46e/kernel)

```
-sh-3.2# ./m46ecli
m46e >
```

3. M46E-PR(prefix resolution)設定コマンド

本項では M46E-PR の prefix 設定用コマンドについて記載する。

3.1. M46E-PR コマンド詳細

[コマンド名]

pr

[機能]

M46E-PR の宛先アドレスの設定に使用する。宛先の IPv4 アドレスとそれに関連する M46E-prefix,planeID を指定し、M46E アドレスを生成する。引数なしで実行した場合は、現在設定されている M46E-PR の宛先アドレステーブル一覧を表示する。

[入力形式]

pr -s pr-prefix <ipv4addr/mask> <m46e-prefix(64bit)> <planeID> pr -s default <m46e-prefix(64bit)> pr -d pr-prefix <ipv4addr/mask> <planeID> pr -d default pr -f <filepath>
--

[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。

-f

指定したファイルに記載されている宛先アドレスを全てテーブルに追加する。

[引数の説明]

- ・ pr-prefix

指定の宛先 IPv4 アドレスに対する設定を追加、削除する際に指定する。

- ・ default

デフォルト（該当する宛先なしの場合）の設定を追加、削除する際に指定する。
(M46E-FP の宛先アドレス設定時に使用する)

- ・ <ipv4addr/mask>

宛先 IPv4 ネットワークアドレス/マスク値。
IPv4 アドレス表記で指定する。

- ・ <m46e-prefix(64bit)>

指定した宛先 IPv4 アドレスに付加する M46E prefix。

IPv6 アドレス表記で指定する。

※IPv6 アドレス表記で指定するにあたり、64bit 目以降は 0 を入れて設定すること。

prefix の値で使用されるのは 64bit までであり、他の値を入れても、64bit 目以降は plane ID と ipv4 アドレスで上書きされる。

- ・ <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)。

- ・ <filepath>

登録したい内容の一覧を記載したファイルのフルパス。

ファイルには、登録したい宛先 IPv4 アドレス、prefix、plane ID の組み合わせを各行にカンマ区切りで記載する。

<ipv4addr/mask>,<m46e-prefix(64bit)>,<planeID>

[備考]

M46E 疑似インタフェースが IPv4 パケットを受け取ると、M46E-PR の宛先アドレステーブルから、自分の M46E アドレスの plane ID に一致し、かつ、宛先の IPv4 アドレスに一致するエントリを検索して、カプセル化を行う。

宛先が登録されていないパケットを受け取った場合は、デフォルトで設定した prefix と、自分の M46E アドレスの plane ID を使用してカプセル化する。デフォルトが未設定の状態で、一致するエントリがないパケットが転送されてきた場合は、パケットを破棄する。

[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「192.168.1.0/24」、plane ID 「1」、M46E prefix 「3ffe:1::」、のエントリを追加する場合。

```
> pr -s pr-prefix 192.168.1.0/24 3ffe:1:: 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「192.168.1.0/24」、plane ID 「1」、のエントリを削除する場合。

```
> pr -d pr-prefix 192.168.1.0/24 1
```

- 宛先アドレスをテーブルにファイルで一括登録する

```
> pr -f pr.txt
```

<ファイルの記載例 (pr. txt)>

192.168.1.0/24, 3ffe:1::, 1

192.168.1.0/24, 3ffe:1::, 2

192.168.2.0/24, 3ffe:2::, 1

- テーブル一覧を表示する

```
> pr
```

<出力例>

PlaneID	IPv4addr	Mask	M46E-PR Prefix
1	192.168.2.0	/120	3ffe:2::1:c0a8:200
1	192.168.1.0	/120	3ffe:1::1:c0a8:100
2	192.168.1.0	/120	3ffe:1::2:c0a8:100

4. PMTU 設定コマンド

本項では M46E-PMTU の設定用コマンドについて記載する。

4.1. PMTU コマンド詳細

[コマンド名]

pmtu

[機能]

指定した経路に MTU を設定する。また、MTU のキャッシュのクリア間隔や、強制フラグメント機能の ON、OFF を設定する。引数なしで実行した場合は、現在設定されている経路毎の MTU の一覧と強制フラグメント機能の ON、OFF を表示する。

[入力形式]

pmtu -s <ip_addr> <mtu_value> <plane ID> pmtu -d <ip_addr> <plane ID> pmtu -t <timeout_value> pmtu -f <on-off>

[オプション]

-s

指定した経路に MTU を設定する。

-d

指定した経路の設定を削除する。

-t

MTU のキャッシュがクリアされる間隔を指定する。

-f

強制フラグメント機能の ON、OFF を指定する。

[引数の説明]

- ・ <ip_addr>

MTU を設定する宛先の IPv4 アドレス。IPv4 アドレス表記で指定する。

- ・ <mtu_value>

設定する MTU サイズ。

設定可能な値は、1280-1500 (10 進数で指定)。単位は byte。

- ・ <plane ID>

MTU を設定する宛先の plane ID。

設定可能な値は、0-4294967295 (10 進数で指定)。

- ・ <timeout_value>

MTU のキャッシュがクリアされる間隔。

pmtu -s コマンドで静的に設定した MTU はクリアされない。

設定可能な値は、300-86400（10 進数で指定）。単位は秒。デフォルトは 600。

- ・ <on-off>

強制フラグメント機能（備考参照）の ON、OFF の指定。

設定可能な値は、"on"または"off"。デフォルトは OFF。

[備考]

M46E 疑似インタフェースでは、カプセル化によって IPv6 ヘッダ（40byte）分パケットサイズが増加する。そのため、カプセル化後の IPv6 パケットが送信エラーになる場合は、送信できる IPv4 パケットのサイズを ICMP エラーで返すか、フラグメントを行いパケットを分割して送信する。各経路の MTU サイズは ICMP エラーを受信した際に保存され、キャッシュクリア間隔の間保持される。

特定の経路に対する MTU サイズが事前に判明している場合は、pmtu -s コマンドで MTU サイズを設定しておくことができる。

強制フラグメント機能は、フラグメント禁止フラグ（DFbit）が有効になっているパケットに対しても、強制的にフラグメントを行う機能である。この機能が ON の状態で、フラグメント禁止パケットを受信すると、フラグメント禁止フラグを無効化してフラグメントを実施する。

[コマンド入力例]

○経路に MTU を設定する

宛先「192.168.1.1」、PlaneID 1 への経路の MTU 「1280 byte」を設定する場合。

> pmtu -s 192.168.1.1 1280 1

○指定した経路に設定した MTU を削除する

宛先「192.168.1.1」、PlaneID 1 への経路の MTU を削除する場合。

> pmtu -d 192.168.1.1 1

○MTU のキャッシュがクリアされる間隔を指定する

MTU のキャッシュクリアの間隔を「400 秒」に設定する場合。

> pmtu -t 400

○強制フラグメント機能を ON にする

> pmtu -f on

○MTU を設定した経路の一覧を表示する

> pmtu

＜出力例＞

force fragment = ON

Address	planeID	MTU	Life(sec)	initial value = 400
---------	---------	-----	-----------	---------------------

-----	-----	----	-----	
-------	-------	------	-------	--

192.168.1.1	1	1280	static	
-------------	---	------	--------	--

192.168.133.123	1	1300	static	
-----------------	---	------	--------	--

192.168.100.123	1	1300	static	
-----------------	---	------	--------	--

Total entries : 3

5. Network NameSpace 設定コマンド

本項では Network NameSpace 設定用コマンドについて記載する。

5.1. NS(Network NameSpace)コマンド詳細

[コマンド名]

ns

[機能]

コマンドを実行した M46E アプライアンスに Network NameSpace を追加、設定するために使用する。コマンドを実行した M46E アプライアンスに Network NameSpace を作成し、作成した Network NameSpace に m46e インタフェースを追加する。また、物理デバイスを指定した Network NameSpace に移動する。引数なしで実行した場合は、現在設定されている Network NameSpace の一覧を表示する。

[入力形式]

ns -s <NameSpace name> <planeid> <backbone v6 IF> <NameSpace v6 IF> ns -s <NameSpace name> <planeid> ns -d <NameSpace name> ns -m <IF> <NameSpace name> ns -i

[オプション]

-s

指定した名前の Network NameSpace を作成する。
<backbone v6 IF> <NameSpace v6 IF>が省略されていた場合は、自動で名前を設定する。

-d

指定した NameSpace を削除する。

-m

インタフェースを指定した NameSpace に移動する。

-i

インタフェースと ifindex の一覧を表示する。

[引数の説明]

- ・ <NameSpace name>

Network NameSpace の名前。

設定可能な文字数は 65 文字まで。

設定可能な文字の種類は、以下の通り。

アルファベット大文字("A"~"Z")

アルファベット小文字("a"~"z")、

数字("0"~"9")

ハイフン("-")※1

ピリオド(".")※1、2

※1 ハイフン、ピリオドは先頭の文字として使用できない。

※2 ピリオドは<NameSpace name>の最後の文字として使用できない。

- ・ <planeID>

追加する Network NameSpace の plane ID。

設定可能な値は、0-4294967295 (10 進数で指定)。

- ・ <backbone v6 IF>

Network NameSpace の Backbone 側のペア仮想インタフェースの名前。

設定可能な文字数は 15 文字。システムに存在しない IF 名のみ設定可能。

省略された場合は、自動で設定する。

- ・ <NameSpace v6 IF>

Network NameSpace の NameSpace 側のペア仮想インタフェースの名前。

設定可能な文字数は 15 文字。システムに存在しない IF 名のみ設定可能。

省略された場合は、自動で設定する。

- ・ <IF>

Network NameSpace に移動させる物理インタフェース。

システムに存在する IF 名から、移動させたいインタフェースを指定する。

[備考]

Network NameSpace を作成すると、Backbone 側と NameSpace 側で通信するためのペア仮想インタフェースと、各 Network NameSpace 内の M46E 疑似インタフェースがそれぞれ作成される。Network NameSpace の Backbone 側と NameSpace 側は独立したネットワークであり、仮想的に上記ペア仮想インタフェースで直接接続された状態となる。Backbone 側と NameSpace 側のパケットのやり取りは、このペア仮想インタフェースに転送することで行う。

ペア仮想インタフェース名を自動で設定した場合は、ns コマンドで一覧表示を行うことで、Backbone 側のペア仮想インタフェース名を確認できる(表示結果の“IF”列に記載される)。また、NameSpace 側のペア仮想インタフェース名、および、M46E 疑似インタフェース名は、Network NameSpace 内で ifconfig コマンドを実行して確認できる。

ペア仮想インタフェースには、以下のアドレスがそれぞれ自動で設定される。

Backbone 側	fe80::1
NameSpace 側	fe80::2

Network NameSpace 内の設定を行うためには、上記アドレスを使用して ssh でログインする。パスワードは OS に設定されているパスワードと同一である。なお、設定されるアドレスはすべての Network NameSpace で同一のため、Backbone 側のペア仮想インタフェース名を合わせて指定することで、接続先を切り替える。

```
-sh-3.2# ssh root@fe80::2%(Backbone 側のペア仮想インタフェース名)
```

各 NetworkNameSpace を IPv4 ネットワークと接続するには、IPv4 ネットワークに繋がっている物理インタフェースを Network NameSpace 内に移動させる。これにより、Network NameSpace と IPv4 ネットワークが直接接続された状態になる。

M46E の設定を行う m46ecli アプリは、Backbone 側で使用する。

先述した設定は、すべての Network NameSpace で共通となる。Backbone 側の m46ecli アプリで pr コマンドを実行し、それぞれの Network NameSpace の設定を追加する。各 Network NameSpace 内の M46E 疑似インタフェースは、M46E アドレスの plane ID と転送するパケットの宛先 IPv4 アドレスから、適用すべきエントリを検索する。

[コマンド入力例]

- NameSpace を作成する。

「space1」という名前で、plane ID「1」の Network NameSpace を、ペア仮想インタフェース名を指定（Backbone 側「veth1」、NameSpace 側「veth2」）して作成する場合。

```
> ns -s space1 1 veth1 veth2
```

- NameSpace を削除する。

Network NameSpace「space1」を削除する場合。

```
> ns -d space1
```

- インタフェースを NameSpace に移動する。

物理インタフェース「eth1」を Network NameSpace「space1」に移動する場合。

```
> ns -m eth1 space1
```

- インタフェースの index 番号一覧を表示する。

```
> ns -i
```

＜出力例＞

index	name
1	lo
2	eth0
3	eth1
4	eth2

※Network NameSpace 外部のインタフェースのみ表示。

- Network NameSpace の一覧を表示する

```
> ns
```

＜出力例＞

PlaneID	address	IF	NameSpace name
1	fe80::2	veth1	space1
2	fe80::2	veth3	space2

6. M46E-FP 設定

M46E-FP の設定について記載する

6.1. 機器毎の設定

ActiveAssist PF1000、もしくは CentOS7.2 で M46E 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。

6.1.1. 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。

M46E 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/m46e/kernel

M46E 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
m46e.ko	M46E カーネルモジュール。M46E を実行する際にカーネルに組み込むモジュール。
sample-init	M46E 設定サンプル。実際の設定の例を記載してあるファイル。
m46ecli	M46E の設定をするアプリ。本アプリで M46E の使用する設定を追加できる。

6.1.2. 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
Proxy NDP	0	1

6.2. M46E モジュール起動設定

M46E 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 M46E 設定ファイル配置場所に移動する。

```
-sh-3.2# cd /root/m46e/kernel
```

- M46E モジュールを insmod する。

```
-sh-3.2# insmod m46e.ko
```

insmod を実行すると、M46E の疑似インタフェース m46e0 が作成される。

- フォワーディングを有効にする。

```
-sh-3.2# echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
-sh-3.2# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

6.3. M46E モジュール終了設定

M46E 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。

- M46E モジュールを rmmod する。

```
-sh-3.2# rmmod m46e.ko
```

6.4. インタフェース、ルーティングテーブルの設定の考え方

本項では M46E を実行するために必要なインタフェース、ルーティングテーブルの設定の考え方について記載する。設定には、Linux の iptables、route コマンドを使用する。

- M46E 疑似インタフェースのアドレスの考え方

M46E 疑似インタフェースには、以下の形式の M46E アドレスを設定する。

例)	<u>3ffe:3::</u>	<u>1:</u>	<u>c0a8:0300</u>	<u>/120</u>	※192.168.3.0/24 のネットワークを接続する場合
	①	②	③	④	

- ① M46E の prefix (64bit)
- ② plane ID(32bit)。同じ IPv4 アドレスの Stub ネットワーク同士で値が重ならないように、かつ、通信したい相手の plane ID と一致するように設定。
- ③ 接続する IPv4 Stub ネットワークのアドレス(32bit)。16 進数に変換して設定。
- ④ 接続する IPv4 Stub ネットワークのマスク値。以下の値を 10 進数で設定。
[IPv4 アドレスのマスク値] + 64(=prefix の bit 数) + 32(=plane ID の bit 数)

- 物理インタフェースのアドレスの考え方

物理インタフェースには、それぞれが属するネットワークに合わせて、適切なアドレスを設定する。

- ルーティングテーブルの設定の考え方

- ・ IPv4 ルーティングテーブル

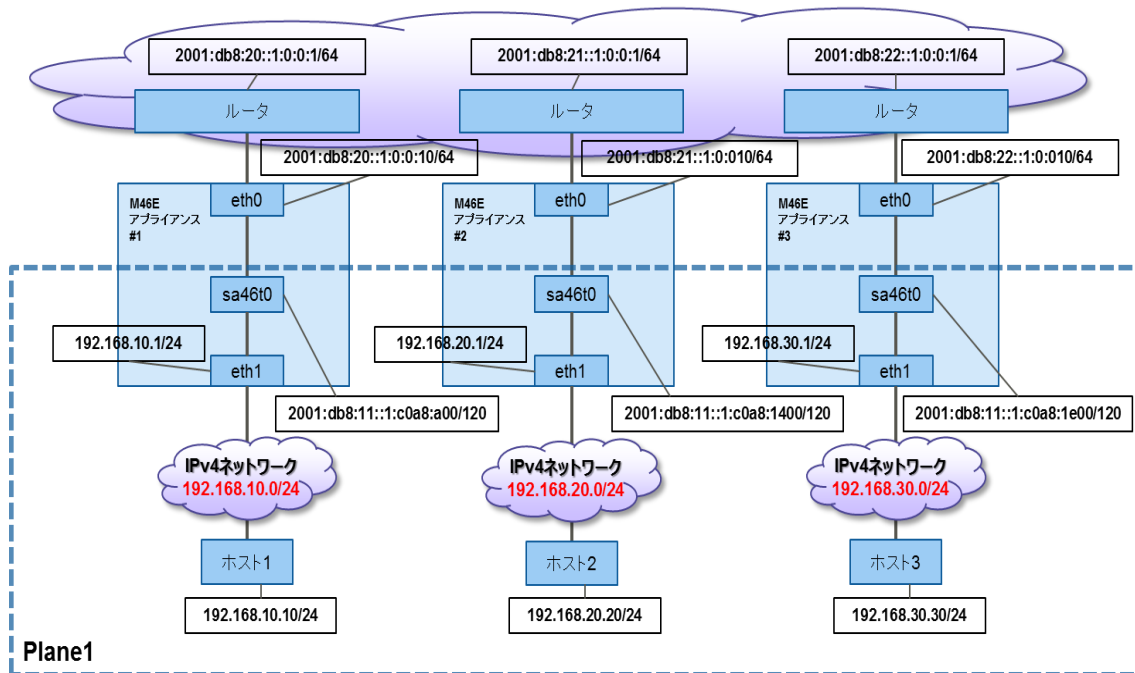
カプセル化したいパケットのルートは、M46E 疑似インタフェースに向ける。すべてのパケットをカプセル化する場合は、デフォルトルートに設定する。

- ・ IPv6 ルーティングテーブル

IPv6 パケットのデフォルトルートは、IPv6 Backbone ネットワークの入り口のアドレスを設定する。個別のルート設定が必要な場合は、static route を追加すること。

6.5. M46E-FP 設定例

ネットワーク構成の例を用いて M46E-FP の設定方法を記載する。以下に M46E-FP の構成例を示す。



M46E-FP を用いて、図中のホスト 1、ホスト 2、ホスト 3 で通信する場合、M46E アプライアンス#1,#2,#3 に必要な設定を以下に示す。

◆ M46E アプライアンス#1 の設定

- ① M46E モジュール配置ディレクトリに移動し、M46E モジュールを insmod する。

```
cd /root/m46e/kernel
insmod m46e.ko
```

- ② インタフェースの設定をする。

```
ifconfig eth0 up
ifconfig eth0 add 2001:db8:20::1:0:0:10/64
ifconfig eth1 192.168.10.1/24 up
```

- ③ M46E 疑似インタフェース(m46e0)にアドレスを設定する。

```
ifconfig m46e0 up
ifconfig m46e0 add 2001:db8:11::1:c0a8:a00/120
```

- ④ M46E でカプセル化する IPv4 パケットが m46e0 へ転送されるように、経路情報を追加する。

```
route add default dev m46e0
```

- ⑤ カプセル化パケットの経路情報を設定する。

```
route -A inet6 add default gw 2001:db8:20::1:0:0:1 dev eth0
```

- ⑥ m46ecli で M46E Prefix の設定をする。

```
m46e >pr -s default 2001:db8:11::
```

以降、M46E アプライアンス#2、M46E アプライアンス#3 について、同様の設定を行う。

◆ M46E アプライアンス#2 の設定

- ① インタフェースの設定をする。

```
ifconfig eth0 up  
ifconfig eth0 add 2001:db8:21::1:0:0:10/64  
ifconfig eth1 192.168.20.1/24 up
```

- ② M46E 疑似インタフェース(m46e0)にアドレスを設定する。

```
ifconfig m46e0 up  
ifconfig m46e0 add 2001:db8:11::1:c0a8:1400/120
```

- ③ M46E でカプセル化する IPv4 パケットが m46e0 へ転送されるように、経路情報を追加する。

```
route add default dev m46e0
```

- ④ カプセル化パケットの経路情報を設定する。

```
route -A inet6 add default gw 2001:db8:21::1:0:0:1 dev eth0
```

- ⑤ m46ecli で M46E Prefix の設定をする。

```
m46e >pr -s default 2001:db8:11::
```

◆ M46E アプライアンス#3 の設定

① インタフェースの設定をする。

```
ifconfig eth0 up  
ifconfig eth0 add 2001:db8:22::1:0:0:10/64  
ifconfig eth1 192.168.30.1/24 up
```

② M46E 疑似インタフェース(m46e0)にアドレスを設定する。

```
ifconfig m46e0 up  
ifconfig m46e0 add 2001:db8:22::1:c0a8:1e00/120
```

③ M46E でカプセル化する IPv4 パケットが m46e0 へ転送されるように、経路情報を追加する。

```
route add default dev m46e0
```

④ カプセル化パケットの経路情報を設定する。

```
route -A inet6 add default gw 2001:db8:22::1:0:0:1 dev eth0
```

⑤ m46ecli で宛先毎の M46E Prefix の設定をする。

```
m46e >pr -s default 2001:db8:11::
```

7. 付録

7.1. M46E の設定ファイル出力

m46ecli で config コマンドに save オプションとファイルパスを指定することで、現在設定されている M46E の設定をファイルに出力することができる。

```
m46e >config save <filepath>
```

config コマンドによって保存される設定、保存されない設定を以下に記載する。

保存される設定は、以下の通り。

コマンド名	保存される設定
pr	・ pr -s pr-prefix コマンドで追加した、各 M46E-PR の設定。 ・ pr -s default コマンドで追加したデフォルトの M46E-FP の設定。
pmtu	・ pmtu -s コマンドで設定した、各経路の MTU。 ・ pmtu -t コマンドで設定した、MTU キャッシュのクリア間隔。 ・ pmtu -f コマンドで設定した、強制フラグメントの ON、OFF。
ns	・ ns -s コマンドで設定した、各 Network Namespace の作成。 (仮想ペアインタフェース名も保存される)

保存されない設定は以下の通り。

コマンド名	保存されない設定
ns	・ ns -m コマンドでの物理インタフェースの移動

その他、m46ecli アプリ以外で設定する以下の内容も保存の対象外となる。

- フォワーディングの設定。
- インタフェース、ルーティングテーブルの設定。
- Network NameSpace 内の設定。

7.2. ファイルから M46E の設定読み込み

m46ecli に -f オプションと m46ecli で出力したコンフィグファイルのファイルパスを指定することで、コンフィグファイルに記載されているコンフィグがすべて設定される。

```
-sh-3.2# ./m46ecli -f <filepath>
```

コンフィグファイル（付録 10.1 のコマンドで出力）の例を以下に記載する。

```
pr -s pr-prefix 192.168.2.0/24 3ffe:2::1:c0a8:200 1
pr -s pr-prefix 192.168.3.0/24 3ffe:3::2:c0a8:300 2
pr -s default 3ffe:1::
pmtu -t 400
pmtu -s 192.168.2.0 1400 1
pmtu -s 192.168.3.0 1400 2
ns -s space1 1 peth0 peth1
ns -s space2 2 peth2 peth3
```

なお、-f オプションで読み込んだコンフィグファイルの設定は、m46ecli ですでに設定済みの内容がある場合、それに追加する形で設定される。設定のエラーや矛盾が発生する可能性があるため、m46ecli での設定を行っていない状態で読み込むこと。

7.3. sample-init

M46E 起動時の設定例である sample-init を記載する。

```
insmod /root/m46e/kernel/m46e.ko

ifconfig ethx up
ifconfig ethx up
ifconfig ethx 0
ifconfig ethx add xxxx:xxxx::xxxx:xxxx/64
ifconfig ethx xxx.xxx.xxx.xxx/24
ifconfig ethx xxx.xxx.xxx.xxx/24
ifconfig m46e0 up
ifconfig m46e0 add xxxx:xxxx::xxxx:xxxx:xxxx/120
ifconfig m46e0 add xxxx:xxxx::xxxx:xxxx:xxxx/120

route add default dev m46ex
route -A inet6 add default gw xxxx:xxxx::xxxx:xxxx ethx

echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding

#/root/m46e/kernel/m46ecli -f /root/m46e/kernel/config
```

上記設定例では、基本設定（M46E モジュールの insmod とフォワーディングの有効化）、各インタフェースの起動、IP アドレスの設定、ルーティングテーブルの設定を実施している。最後のコメントアウトされた行では、付録 10.2 の M46E 設定の読み込みを実施している。（設定ファイルのパスは/root/m46e/kernel/config を想定）

7.4. M46E の自動起動

M46E の自動起動について記載する。ActiveAssist PF1000 起動時に M46E を自動で起動したい場合は、IP アドレス等を適切に設定した sample-init を以下に記載することで実現する。M46E 固有の設定も自動で設定したい場合は sample-init の最後の行のコメントアウトを外すこと。

```
/etc/rc.local
```

rc.local 記載例

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
/root/m46e/kernel/sample-init
```

※自動起動が可能なのは、Network NameSpace を使用しない場合のみ。Network NameSpace を使用する場合は、物理インタフェースの移動や、Network NameSpace 内の設定が必要なため、手動での起動となる。

以上