FutureNet NXR 設定例集

GRE・IPinIP 編

Ver 1.0.0

センチュリー・システムズ株式会社

目次

目次 2

はじめに 3

改版履歴 4

NXR シリーズの GRE・IPinIP 機能 5

1. GRE・IPinIP 基本設定 6

1-1. GRE（IPinIP）での LAN 間接続設定 7

1-2. PPPoE を利用した GRE（IPinIP）接続設定 12

2. GRE・IPinIP 応用設定 24

2-1. GRE（IPinIP）経由でのインターネット接続設定 25

2-2. GRE（IPinIP）over IPsec 設定 36

付録 50

GRE・IPinIP 状態確認方法 51

サポートデスクへのお問い合わせ 52

サポートデスクへのお問い合わせに関して 53

サポートデスクのご利用に関して 54

はじめに

- FutureNet はセンチュリー・システムズ株式会社の登録商標です。
- その他本書に記載されている会社名，製品名は、各社の商標または登録商標です。
- 本ガイドは、以下の FutureNet NXR 製品に対応しております。
  - NXR-120/C, NXR-125/CX, NXR-130/C, NXR-155/C-WM, NXR-1200
- 本書の内容の一部または全部を無断で転載することを禁止しています。
- 本書の内容については、将来予告なしに変更することがあります。
- 本書の内容については万全を期しておりますが、ご不審な点や誤り、記載漏れ等お気づきの点がありましたらお手数ですが、ご一報下さいますようお願い致します。
- 本書は FutureNet NXR-120/C の以下のバージョンをベースに作成しております。

  FutureNet NXR シリーズ NXR-120/C Ver5.14.2

  各種機能において、ご使用されている製品およびファームウェアのバージョンによっては一部機能，コマンドおよび設定画面が異なっている場合もありますので、その場合は各製品のユーザーズガイドを参考に適宜読みかえてご参照および設定を行って下さい。
- 本バージョンでは IPv4 のみを対象とし、IPv6 の設定に関しては本バージョンでは記載しておりません。
- 設定した内容の復帰（流し込み）を行う場合は、CLI では「copy」コマンド，GUI では設定の復帰を行う必要があります。
- モバイル通信端末をご利用頂く場合で契約内容が従量制またはそれに準ずる場合、大量のデータ通信を行うと利用料が高額になりますので、ご注意下さい。
- 本書を利用し運用した結果発生した問題に関しましては、責任を負いかねますのでご了承下さい。

改版履歴

Version	更新内容
1.0.0	初版

NXR シリーズの GRE・IPinIP 機能

■ GRE 機能

GRE(Generic Routing Encapsulation)はトンネリングプロトコルの一つです。

GRE トンネルは暗号化を行いませんが、IPsec トンネル内に GRE トンネルを生成することも可能ですので、GRE を使用する場合でもセキュアな通信路を構築することができます。

NXR シリーズでは GRE 機能のオプションとして ID キー, チェックサムをサポートしています。

なおこのオプション機能を使用しない場合の GRE ヘッダは 4 バイトとなり、ID キーを使用した場合は 4 バイト、チェックサムを使用した場合はさらに 4 バイト付与されます。よって最大 32 バイトのヘッダが付与されます。

■ IPinIP 機能

IPinIP は GRE と同様トンネリングプロトコルの一つです。

GRE と同様暗号化を行いませんが、IPsec トンネル内に IPinIP トンネルを生成しセキュアな通信路を構築することができます。

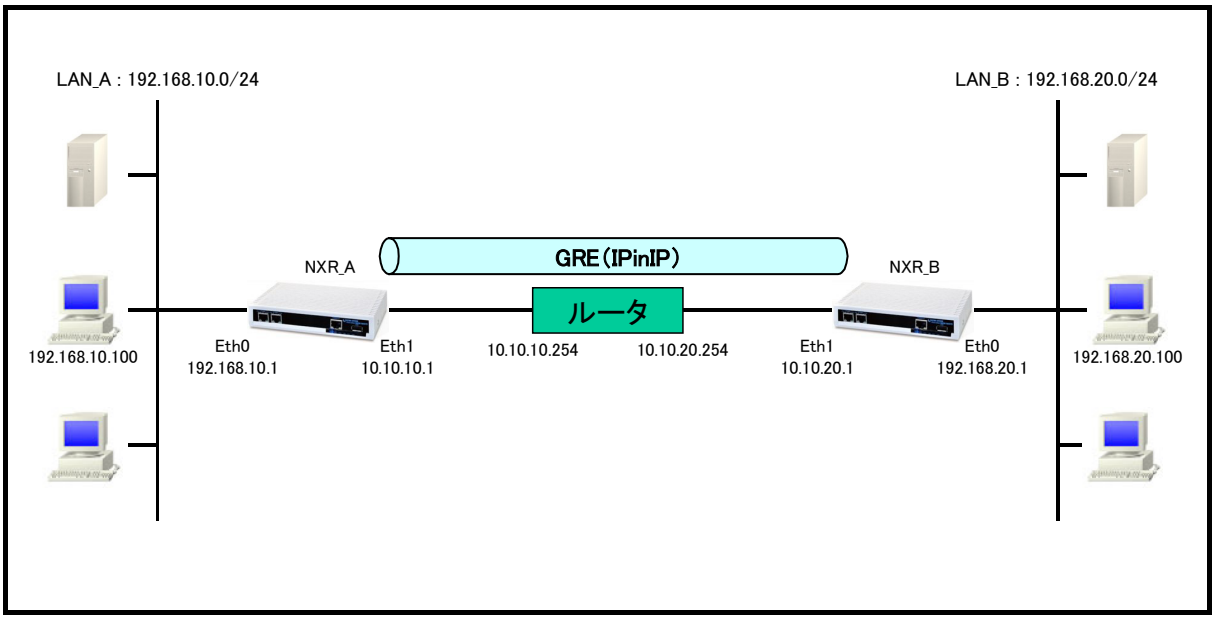
なお IPinIP では IPinIP 用のヘッダはなく、トンネル IP ヘッダの 20 バイトが付与されるのみとなります。

1. GRE・IPinIP 基本設定

1-1. GRE(IPinIP)での LAN 間接続設定

この設定例では LAN_A 192.168.10.0/24 と LAN_B 192.168.20.0/24 のネットワークにある NXR_A, NXR_B 間で GRE(IPinIP)トンネルを構築し、LAN 間通信を可能にします。なお GRE(IPinIP)を使用するルータの WAN 側 IP アドレスはともに固定 IP アドレスが必要となります。
※設定例は GRE ですが、設定例解説の中で IPinIP に関する補足を記載しています。

【 構成図 】



- GRE(IPinIP)ではトンネルインタフェースを使用します。
- GRE(IPinIP)を利用して対向ルータ配下のネットワークと通信する場合でかつそのルートをスタティックルートで設定する場合は、ゲートウェイとしてトンネルインタフェースを指定します。

【 設定例 】

〔NXR_A の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_A
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
NXR_A(config-if)#exit
NXR_A(config)#interface ethernet 1
NXR_A(config-if)#ip address 10.10.10.1/24
NXR_A(config-if)#exit
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
NXR_A(config)#ip route 0.0.0.0/0 10.10.10.254
NXR_A(config)#interface tunnel 1
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
NXR_A(config-tunnel)#exit
NXR_A(config)#exit
NXR_A#save config
```

〔NXR_B の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_B
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
NXR_B(config-if)#exit
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#ip address 10.10.20.1/24
NXR_B(config-if)#exit
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1
NXR_B(config)#ip route 0.0.0.0/0 10.10.20.254
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode gre
NXR_B(config-tunnel)#tunnel source 10.10.20.1
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#tunnel ttl 255
NXR_B(config-tunnel)#exit
NXR_B(config)#exit
NXR_B#save config
```

【 設定例解説 】

〔NXR_A の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_A
```

ホスト名を NXR_A と設定します。

2. ＜LAN 側（ethernet0）インタフェース設定＞

```
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
```

LAN 側（ethernet0）インタフェースの IP アドレスに 192.168.10.1/24 を設定します。

3. ＜WAN 側（ethernet1）インタフェース設定＞

```
NXR_A(config)#interface ethernet 1
NXR_A(config-if)#ip address 10.10.10.1/24
```

WAN 側（ethernet1）インタフェースの IP アドレスとして 10.10.10.1/24 を設定します。

4. ＜スタティックルート設定＞

```
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは LAN_B 向けのルートのゲートウェイインタフェースとして tunnel 1 を設定します。

```
NXR_A(config)#ip route 0.0.0.0/0 10.10.10.254
```

デフォルトルートを設定します。ゲートウェイアドレスは上位ルータの IP アドレスを設定します。

ここではゲートウェイアドレスとして 10.10.10.254 を設定します。

5. ＜tunnel1 インタフェース設定＞

```
NXR_A(config)#interface tunnel 1
```

tunnel1 インタフェースを設定します。

```
NXR_A(config-tunnel)#tunnel mode gre
```

トンネルインタフェースで使用するトンネルモードを設定します。

トンネルインタフェースを GRE で使用する場合は、gre と設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

```
NXR_A(config-tunnel)#tunnel source 10.10.10.1
```

トンネルの送信元 IP アドレスとして機器の WAN 側（ethernet1）IP アドレス 10.10.10.1 を設定します。

```
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
```

トンネルのあて先 IP アドレスとして対向機器の WAN 側 IP アドレス 10.10.20.1 を設定します。

```
NXR_A(config-tunnel)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

(☞) ここでは MSS 値の算出にはトンネルインタフェースの MTU 値を使用します。

```
NXR_A(config-tunnel)#tunnel ttl 255
```

TTL 値を 255 に設定します。

〔NXR_B の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_B
```

ホスト名を NXR_B と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.20.1/24 を設定します。

3. ＜WAN 側(ethernet1)インタフェース設定＞

```
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#ip address 10.10.20.1/24
```

WAN 側(ethernet1)インタフェースの IP アドレスとして 10.10.20.1/24 を設定します。

4. ＜スタティックルート設定＞

```
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは LAN_A 向けのルートのゲートウェイインタフェースとして tunnel 1 を設定します。

```
NXR_B(config)#ip route 0.0.0.0/0 10.10.20.254
```

デフォルトルートを設定します。ゲートウェイアドレスは上位ルータの IP アドレスを設定します。

ここではゲートウェイアドレスとして 10.10.20.254 を設定します。

5. ＜tunnel1 インタフェース設定＞

```
NXR_B(config)#interface tunnel 1
```

tunnel1 インタフェースを設定します。

NXR_B(config-tunnel)#**tunnel mode gre**

トンネルインタフェースで使用するトンネルモードを設定します。

トンネルインタフェースを GRE で使用する場合は、gre と設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

NXR_B(config-tunnel)#**tunnel source 10.10.20.1**

トンネルの送信元 IP アドレスとして機器の WAN 側(ethernet1)IP アドレス 10.10.20.1 を設定します。

NXR_B(config-tunnel)#**tunnel destination 10.10.10.1**

トンネルのあて先 IP アドレスとして対向機器の WAN 側 IP アドレス 10.10.10.1 を設定します。

NXR_B(config-tunnel)#**ip tcp adjust-mss auto**

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

(☞) ここでは MSS 値の算出にはトンネルインタフェースの MTU 値を使用します。

NXR_B(config-tunnel)#**tunnel ttl 255**

TTL 値を 255 に設定します。

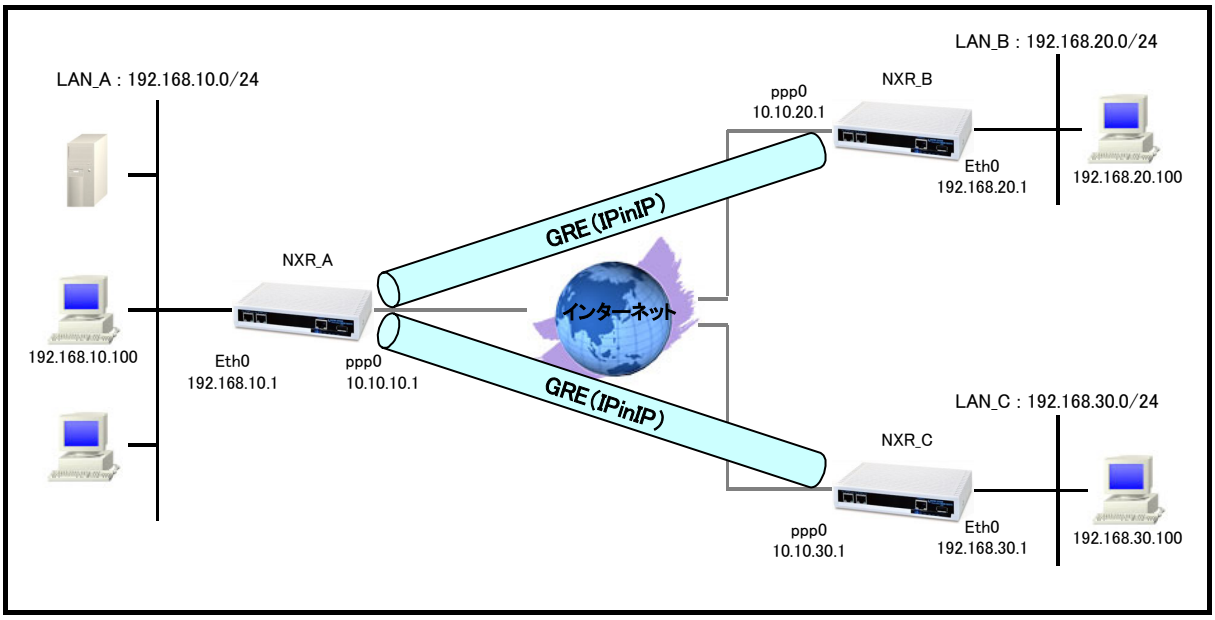
【 パソコンの設定例 】

	LAN A のパソコン	LAN B のパソコン
IP アドレス	192.168.10.100	192.168.20.100
サブネットマスク	255.255.255.0	255.255.255.0
デフォルトゲートウェイ	192.168.10.1	192.168.20.1

1-2. PPPoE を利用した GRE（IPinIP）接続設定

PPPoE 環境でも GRE（IPinIP）機能を利用することは可能です。
なおこの設定例では LAN B <-> LAN C 間では通信することができませんが、ルート設定により拠点間通信も可能です。
※設定例は GRE ですが、設定例解説の中で IPinIP に関する補足を記載しています。

【 構成図 】



- GRE（IPinIP）で複数の拠点と接続する場合は、トンネルインタフェースを複数設定します。この例ではセンタルータの NXR_A で NXR_B 向けにトンネル1，NXR_C 向けにトンネル2インタフェースを設定します。
- IP マスカレードを設定し ppp0 インタフェースから出力されるパケットの送信元 IP アドレスを変換します。これにより各拠点の NXR 配下の複数台の端末からインターネットアクセスが可能になります。
- ppp0 インタフェースでステートフルパケットインスペクションを利用しインターネット側からのアクセスを破棄しながらも各拠点の NXR 配下の端末からのアクセスは自由に行えるようにします。
- この例ではインターネット環境での GRE の利用を想定していますので、IP アクセスリストで GRE（IPinIP）に関するパケットを許可する設定を行っています。
- DNS 機能を有効にすることにより各拠点の NXR 配下の端末からの名前解決要求(クエリ要求)を ISP より取得した DNS サーバに転送します。

【 設定例 】

〔NXR_A の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_A
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
NXR_A(config-if)#exit
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
NXR_A(config)#ip route 192.168.30.0/24 tunnel 2
NXR_A(config)#ip route 0.0.0.0/0 ppp 0
NXR_A(config)#ip access-list ppp0_in permit 10.10.20.1 10.10.10.1 47
NXR_A(config)#ip access-list ppp0_in permit 10.10.30.1 10.10.10.1 47
NXR_A(config)#interface ppp 0
NXR_A(config-ppp)#ip address 10.10.10.1/32
NXR_A(config-ppp)#ip masquerade
NXR_A(config-ppp)#ip access-group in ppp0_in
NXR_A(config-ppp)#ip spi-filter
NXR_A(config-ppp)#ip tcp adjust-mss auto
NXR_A(config-ppp)#no ip redirects
NXR_A(config-ppp)#ppp username test1@centurysys password test1pass
NXR_A(config-ppp)#exit
NXR_A(config)#interface ethernet 1
NXR_A(config-if)#no ip address
NXR_A(config-if)#pppoe-client ppp 0
NXR_A(config-if)#exit
NXR_A(config)#interface tunnel 1
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_B
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
NXR_A(config-tunnel)#mtu 1430
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
NXR_A(config-tunnel)#exit
NXR_A(config)#interface tunnel 2
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_C
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.30.1
NXR_A(config-tunnel)#mtu 1430
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
NXR_A(config-tunnel)#exit
NXR_A(config)#dns
NXR_A(config-dns)#service enable
NXR_A(config-dns)#exit
NXR_A(config)#exit
NXR_A#save config
```

〔NXR_B の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_B
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
NXR_B(config-if)#exit
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1
NXR_B(config)#ip route 0.0.0.0/0 ppp 0
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.20.1 47
NXR_B(config)#interface ppp 0
NXR_B(config-ppp)#ip address 10.10.20.1/32
NXR_B(config-ppp)#ip masquerade
NXR_B(config-ppp)#ip access-group in ppp0_in
NXR_B(config-ppp)#ip spi-filter
NXR_B(config-ppp)#ip tcp adjust-mss auto
NXR_B(config-ppp)#no ip redirects
NXR_B(config-ppp)#ppp username test2@centurysys password test2pass
NXR_B(config-ppp)#exit
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
NXR_B(config-if)#exit
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode gre
NXR_B(config-tunnel)#description NXR_A
NXR_B(config-tunnel)#tunnel source 10.10.20.1
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
NXR_B(config-tunnel)#mtu 1430
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#tunnel ttl 255
NXR_B(config-tunnel)#exit
NXR_B(config)#dns
NXR_B(config-dns)#service enable
NXR_B(config-dns)#exit
NXR_B(config)#exit
NXR_B#save config
```

〔NXR_C の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_C
NXR_C(config)#interface ethernet 0
NXR_C(config-if)#ip address 192.168.30.1/24
NXR_C(config-if)#exit
NXR_C(config)#ip route 192.168.10.0/24 tunnel 1
NXR_C(config)#ip route 0.0.0.0/0 ppp 0
NXR_C(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.30.1 47
NXR_C(config)#interface ppp 0
NXR_C(config-ppp)#ip address 10.10.30.1/32
NXR_C(config-ppp)#ip masquerade
NXR_C(config-ppp)#ip access-group in ppp0_in
NXR_C(config-ppp)#ip spi-filter
NXR_C(config-ppp)#ip tcp adjust-mss auto
NXR_C(config-ppp)#no ip redirects
NXR_C(config-ppp)#ppp username test3@centurysys password test3pass
NXR_C(config-ppp)#exit
NXR_C(config)#interface ethernet 1
NXR_C(config-if)#no ip address
NXR_C(config-if)#pppoe-client ppp 0
NXR_C(config-if)#exit
NXR_C(config)#interface tunnel 1
NXR_C(config-tunnel)#tunnel mode gre
NXR_C(config-tunnel)#description NXR_A
NXR_C(config-tunnel)#tunnel source 10.10.30.1
NXR_C(config-tunnel)#tunnel destination 10.10.10.1
NXR_C(config-tunnel)#mtu 1430
NXR_C(config-tunnel)#ip tcp adjust-mss auto
NXR_C(config-tunnel)#tunnel ttl 255
NXR_C(config-tunnel)#exit
NXR_C(config)#dns
NXR_C(config-dns)#service enable
NXR_C(config-dns)#exit
NXR_C(config)#exit
NXR_C#save config
```

【 設定例解説 】

〔NXR_A の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_A
```

ホスト名を NXR_A と設定します。

2. ＜LAN 側（ethernet0）インタフェース設定＞

```
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
```

LAN 側（ethernet0）インタフェースの IP アドレスに 192.168.10.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
```

GRE（IPinIP）で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE（IPinIP）のカプセル化対象となります。

ゲートウェイは GRE（IPinIP）で使用するトンネルインタフェースを設定します。

ここでは LAN_B(192.168.20.0/24)向けのルートのゲートウェイとして tunnel 1 を設定します。

```
NXR_A(config)#ip route 192.168.30.0/24 tunnel 2
```

GRE（IPinIP）で使用するスタティックルートを設定します。

ここでは LAN_C(192.168.30.0/24)向けのルートのゲートウェイとして tunnel 2 を設定します。

```
NXR_A(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。ゲートウェイとして ppp0 インタフェースを設定します。

4. ＜IP アクセスリスト設定＞

```
NXR_A(config)#ip access-list ppp0_in permit 10.10.20.1 10.10.10.1 47
NXR_A(config)#ip access-list ppp0_in permit 10.10.30.1 10.10.10.1 47
```

フィルタの動作を規定するルールリストを作成します。

ここでは IP アクセスリスト名を ppp0_in とします。

一行目の設定は送信元 IP アドレス 10.10.20.1 あて先 IP アドレス 10.10.10.1 プロトコル番号 47（GRE）のパケットを許可するための設定です。

二行目の設定は送信元 IP アドレス 10.10.30.1 あて先 IP アドレス 10.10.10.1 プロトコル番号 47（GRE）のパケットを許可するための設定です。

この IP アクセスリスト設定は、WAN 側（ppp0）インタフェース設定で登録します。

(☞) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングを行うインタフェースでの登録が必要になります。

(☞) IPinIP を利用する場合はプロトコル番号 4 を設定します。

5. ＜WAN 側（ppp0）インタフェース設定＞

```
NXR_A(config)#interface ppp 0
```

WAN 側（ppp0）インタフェースを設定します。

```
NXR_A(config-ppp)#ip address 10.10.10.1/32
```

IP アドレスを 10.10.10.1/32 に設定します。

```
NXR_A(config-ppp)#ip masquerade
```

IP マスカレードを設定します。

```
NXR_A(config-ppp)#ip access-group in ppp0_in
```

IP アクセスリスト設定で設定した ppp0_in を in フィルタに適用します。これにより ppp0 インタフェースで受信したパケット（NXR 自身宛）に対して IP アクセスリストによるチェックが行われます。

```
NXR_A(config-ppp)#ip spi-filter
```

ステートフルパケットインスペクションを設定します。
ステートフルパケットインスペクションはパケットを監視してパケットフィルタリング項目を随時変更する機能で、動的パケットフィルタリング機能として利用できます。
該当インタフェースでこの設定を有効にした場合、通常そのインタフェースで受信したパケットは全て破棄されますが、そのインタフェースから送信されたパケットに対応する戻りパケットに対してはアクセスを許可します。
これにより自動的に WAN からの不要なアクセスを制御することが可能です。

```
NXR_A(config-ppp)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。
TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_A(config-ppp)#no ip redirects
```

ICMP リダイレクト機能を無効に設定します。

```
NXR_A(config-ppp)#ppp username test1@centurysys password test1pass
```

PPPoE 接続で使用するユーザ ID とパスワードを設定します。
ここではユーザ ID を test1@centurysys, パスワードを test1pass とします。

6. ＜ethernet1 インタフェース設定＞

```
NXR_A(config)#interface ethernet 1
```

ethernet1 インタフェースを設定します。

```
NXR_A(config-if)#no ip address
```

ethernet1 インタフェースに IP アドレスを割り当てない設定をします。
PPPoE 接続でプロバイダ等から割り当てられる IP アドレスはイーサネットインタフェースではなく PPP インタフェースに割り当てられますので、PPPoE のみで使用する場合は IP アドレスの設定は不要です。

```
NXR_A(config-if)#pppoe-client ppp 0
```

ethernet1 インタフェース上で ppp0 インタフェースを使用するための設定をします。

PPPoE で PPP インタフェースを使用する場合は、pppoe-client コマンドによるインタフェース設定での登録が必要になります。

7. ＜tunnel1 インタフェース設定＞

```
NXR_A(config)#interface tunnel 1
```

tunnel1 インタフェースを設定します。

```
NXR_A(config-tunnel)#tunnel mode gre
```

トンネルインタフェースで使用するトンネルモードを設定します。

トンネルインタフェースを GRE で使用する場合は、gre と設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

```
NXR_A(config-tunnel)#description NXR_B
```

このトンネルの説明として、ここでは NXR_B と設定します。

```
NXR_A(config-tunnel)#tunnel source 10.10.10.1
```

トンネルの送信元 IP アドレスとして WAN 側(ppp0)IP アドレス 10.10.10.1 を設定します。

```
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
```

トンネルのあて先 IP アドレスとして対向機器(NXR_B)の WAN 側 IP アドレス 10.10.20.1 を設定します。

```
NXR_A(config-tunnel)#mtu 1430
```

トンネルインタフェースの MTU 値として 1430 を設定します。

(☞) この設定例では PPPoE 接続時の WAN 側インタフェースの MTU 値を 1454 とし、かつ GRE のオプションは使用していません。

```
NXR_A(config-tunnel)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

(☞) ここでは MSS 値の算出にはトンネルインタフェースの MTU 値を使用します。

```
NXR_A(config-tunnel)#tunnel ttl 255
```

TTL 値を 255 に設定します。

8. ＜tunnel2 インタフェース設定＞

```
NXR_A(config)#interface tunnel 2
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_C
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.30.1
NXR_A(config-tunnel)#mtu 1430
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
```

tunnel2 インタフェースを設定します。

このトンネルの説明として、ここでは NXR_C と設定します。

トンネルの送信元 IP アドレスとして機器の WAN 側(ppp0)IP アドレス 10.10.10.1、あて先 IP アドレスとして対向機器(NXR_C)の WAN 側 IP アドレス 10.10.30.1 を設定します。

その他の設定は 7. ＜tunnel1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

9. ＜DNS 設定＞

```
NXR_A(config)#dns
NXR_A(config-dns)#service enable
```

DNS サービスを有効に設定します。

〔NXR_B の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_B
```

ホスト名を NXR_B と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.20.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは LAN_A(192.168.10.0/24)向けのルートのゲートウェイインタフェースとして tunnel 1 を設定します。

```
NXR_B(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。ゲートウェイとして ppp0 インタフェースを指定します。

4. ＜IP アクセスリスト設定＞

```
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.20.1 47
```

フィルタの動作を規定するルールリストを作成します。

ここでは IP アクセスリスト名を ppp0_in とします。

この設定は送信元 IP アドレス 10.10.10.1 あて先 IP アドレス 10.10.20.1 プロトコル番号 47(GRE)のパケットを許

可するための設定です。

この IP アクセスリスト設定は、WAN 側(ppp0)インタフェース設定で登録します。

(☞) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングを行うインタフェースでの登録が必要になります。

(☞) IPinIP を利用する場合はプロトコル番号 4 を設定します。

5. ＜WAN 側(ppp0)インタフェース設定＞

```
NXR_B(config)#interface ppp 0
NXR_B(config-ppp)#ip address 10.10.20.1/32
NXR_B(config-ppp)#ip masquerade
NXR_B(config-ppp)#ip access-group in ppp0_in
NXR_B(config-ppp)#ip spi-filter
NXR_B(config-ppp)#ip tcp adjust-mss auto
NXR_B(config-ppp)#no ip redirects
NXR_B(config-ppp)#ppp username test2@centurysys password test2pass
```

WAN 側(ppp0)インタフェースを設定します。

IP アドレスを 10.10.20.1/32 に設定します。

ここでは PPPoE 接続で使用するユーザ ID とパスワードとしてユーザ ID を test2@centurysys, パスワードを test2pass と設定します。

その他の設定は NXR_A の **5. ＜WAN 側(ppp0)インタフェース設定＞**が参考になりますので、そちらをご参照下さい。

6. ＜ethernet1 インタフェース設定＞

```
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
```

ethernet1 インタフェースを設定します。

この設定は NXR_A の **6. ＜ethernet1 インタフェース設定＞**が参考になりますので、そちらをご参照下さい。

7. ＜tunnel1 インタフェース設定＞

```
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode gre
NXR_B(config-tunnel)#description NXR_A
NXR_B(config-tunnel)#tunnel source 10.10.20.1
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
NXR_B(config-tunnel)#mtu 1430
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#tunnel ttl 255
```

tunnel1 インタフェースを設定します。

トンネルインタフェースで使用するトンネルモードに gre を設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

このトンネルの説明として、ここでは NXR_A と設定します。

トンネルの送信元 IP アドレスとして機器の WAN 側(ppp0)IP アドレス 10.10.20.1 を設定します。

トンネルのあて先 IP アドレスとして対向機器(NXR_A)の WAN 側 IP アドレス 10.10.10.1 を設定します。

その他の設定は NXR_A の **7. ＜tunnel1 インタフェース設定＞**が参考になりますので、そちらをご参照下さい。

8. ＜DNS 設定＞

```
NXR_C(config)#dns
NXR_C(config-dns)#service enable
```

DNS サービスを有効に設定します。

〔NXR_C の設定〕

1. ＜ホスト名の設定＞

```
nxr120(config)#hostname NXR_C
```

ホスト名を NXR_C と設定します。

2. ＜LAN 側（ethernet0）インタフェース設定＞

```
NXR_C(config)#interface ・thernet 0
NXR_C(config-if)#ip address 192.168.30.1/24
```

LAN 側（ethernet0）インタフェースの IP アドレスに 192.168.30.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_C(config)#ip route 192.168.10.0/24 tunnel 1
```

GRE で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE（IPinIP）のカプセル化対象となります。

ゲートウェイは GRE（IPinIP）で使用するトンネルインタフェースを設定します。

ここでは LAN_A(192.168.10.0/24)向けのルートのゲートウェイインタフェースとして tunnel 1 を設定します。

```
NXR_C(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。ゲートウェイとして ppp0 インタフェースを指定します。

4. ＜IP アクセスリスト設定＞

```
NXR_C(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.30.1 47
```

フィルタの動作を規定するルールリストを作成します。

ここでは IP アクセスリスト名を ppp0_in とします。

この設定は送信元 IP アドレス 10.10.10.1 あて先 IP アドレス 10.10.30.1 プロトコル番号 47（GRE）のパケットを許可するための設定です。

この IP アクセスリスト設定は、WAN 側（ppp0）インタフェース設定で登録します。

(☞) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングを行うインタフェースでの登録が必要になります。

(☞) IPinIP を利用する場合はプロトコル番号 4 を設定します。

5. ＜WAN 側(ppp0)インタフェース設定＞

```
NXR_C(config)#interface ppp 0
NXR_C(config-ppp)#ip address 10.10.30.1/32
NXR_C(config-ppp)#ip masquerade
NXR_C(config-ppp)#ip access-group in ppp0_in
NXR_C(config-ppp)#ip spi-filter
NXR_C(config-ppp)#ip tcp adjust-mss auto
NXR_C(config-ppp)#no ip redirects
NXR_C(config-ppp)#ppp username test3@centurysys password test3pass
```

WAN 側(ppp0)インタフェースを設定します。

IP アドレスを 10.10.30.1/32 に設定します。

ここでは PPPoE 接続で使用するユーザ ID とパスワードとしてユーザ ID を test3@centurysys, パスワードを test3pass と設定します。

その他の設定は NXR_A の 5. ＜WAN 側(ppp0)インタフェース設定＞が参考になりますので、そちらをご参照下さい。

6. ＜ethernet1 インタフェース設定＞

```
NXR_C(config)#interface ethernet 1
NXR_C(config-if)#no ip address
NXR_C(config-if)#pppoe-client ppp 0
```

ethernet1 インタフェースを設定します。

この設定は NXR_A の 6. ＜ethernet1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

7. ＜tunnel1 インタフェース設定＞

```
NXR_C(config)#interface tunnel 1
NXR_C(config-tunnel)#tunnel mode gre
NXR_C(config-tunnel)#description NXR_A
NXR_C(config-tunnel)#tunnel source 10.10.30.1
NXR_C(config-tunnel)#tunnel destination 10.10.10.1
NXR_C(config-tunnel)#mtu 1430
NXR_C(config-tunnel)#ip tcp adjust-mss auto
NXR_C(config-tunnel)#tunnel ttl 255
```

tunnel1 インタフェースを設定します。

トンネルインタフェースで使用するトンネルモードに gre を設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

このトンネルの説明として、ここでは NXR_A と設定します。

トンネルの送信元 IP アドレスとして機器の WAN 側(ppp0)IP アドレス 10.10.30.1 を設定します。

トンネルのあて先 IP アドレスとして対向機器(NXR_A)の WAN 側 IP アドレス 10.10.10.1 を設定します。

その他の設定は NXR_A の 7. ＜tunnel1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

8. ＜DNS 設定＞

```
NXR_C(config)#dns
NXR_C(config-dns)#service enable
```

DNS サービスを有効に設定します。

【 パソコンの設定例 】

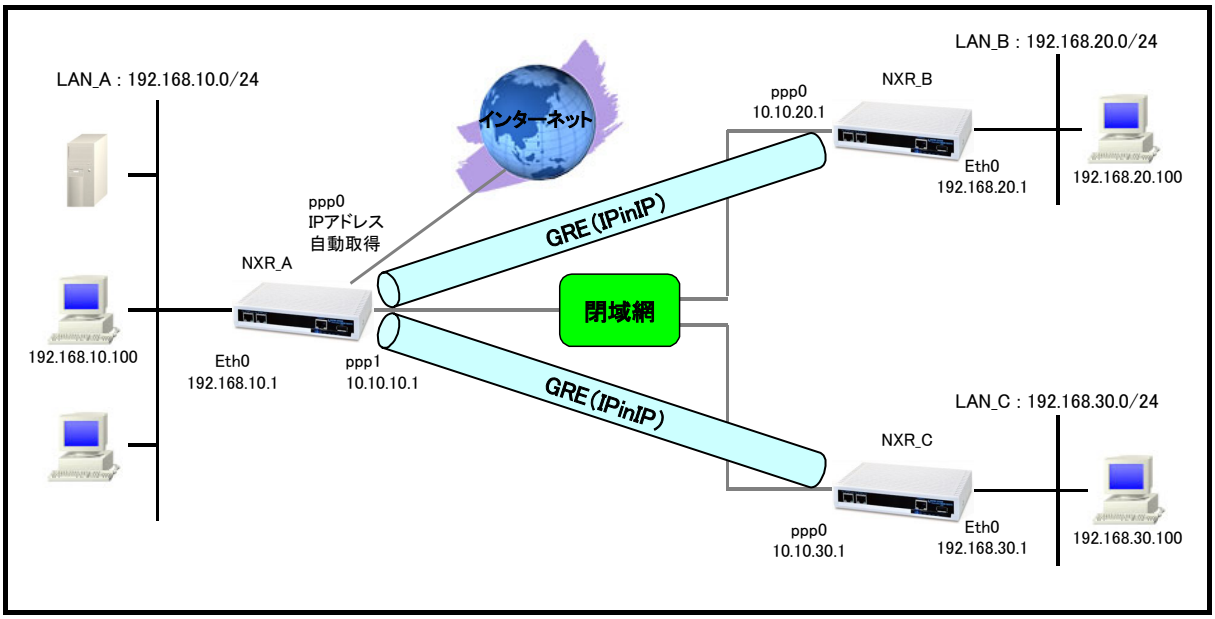
	LAN A のパソコン	LAN B のパソコン	LAN C のパソコン
IP アドレス	192.168.10.100	192.168.20.100	192.168.30.100
サブネットマスク	255.255.255.0	255.255.255.0	255.255.255.0
デフォルトゲートウェイ	192.168.10.1	192.168.20.1	192.168.30.1
DNS サーバの IP アドレス	192.168.10.1	192.168.20.1	192.168.30.1

2. GRE・IPinIP 応用設定

2-1. GRE(IPinIP)経由でのインターネット接続設定

通常閉域網にのみ接続しているルータからはインターネットアクセスすることができませんが、この設定例では GRE(IPinIP)を利用することで拠点間通信だけでなく、センタ経由でインターネットアクセスも可能にします。
※設定例は GRE ですが、設定例解説の中で IPinIP に関する補足を記載しています。

【 構成図 】



- センタルータとなる NXR_A では ppp0 インタフェースでインターネット接続，ppp1 インタフェースでフレッツ VPN ワイドなどの閉域網への接続を行います。
- NXR_A では ppp0 インタフェースで IP マスカレード，ステートフルパケットインスペクションを設定します。
- 拠点ルータとなる NXR_B，C では ppp0 インタフェースでフレッツ VPN ワイドなどの閉域網への接続を行います。
- LAN_B,C 内の端末は GRE(IPinIP)を経由してインターネットアクセスを行い、この例では DNS サーバの IP アドレスとして NXR_A の IP アドレスを指定します。

  (☞) この際 NXR_A の LAN 側(ethernet0)インタフェースがリンクダウンした場合、LAN 側(ethernet0)インタフェースの IP アドレス宛の通信ができなくなりますので、リンクダウン時でも通信できるよう設定をします。

【 設定例 】

〔NXR_A の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_A
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
NXR_A(config-if)#ip access-linkdown
NXR_A(config-if)#exit
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
NXR_A(config)#ip route 192.168.30.0/24 tunnel 2
NXR_A(config)#ip route 0.0.0.0/0 ppp 0
NXR_A(config)#interface ppp 0
NXR_A(config-ppp)#ip address negotiated
NXR_A(config-ppp)#ip masquerade
NXR_A(config-ppp)#ip spi-filter
NXR_A(config-ppp)#ip tcp adjust-mss auto
NXR_A(config-ppp)#no ip redirects
NXR_A(config-ppp)#ppp username test@centurysys password testpass
NXR_A(config-ppp)#exit
NXR_A(config)#interface ppp 1
NXR_A(config-ppp)#ip address 10.10.10.1/32
NXR_A(config-ppp)#ip send-source
NXR_A(config-ppp)#ip tcp adjust-mss auto
NXR_A(config-ppp)#no ip redirects
NXR_A(config-ppp)#ppp username test1@centurysys password test1pass
NXR_A(config-ppp)#exit
NXR_A(config)#interface ethernet 1
NXR_A(config-if)#no ip address
NXR_A(config-if)#pppoe-client ppp 0
NXR_A(config-if)#pppoe-client ppp 1
NXR_A(config-if)#exit
NXR_A(config)#interface tunnel 1
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_B
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
NXR_A(config-tunnel)#mtu 1430
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
NXR_A(config-tunnel)#exit
NXR_A(config)#interface tunnel 2
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_C
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.30.1
NXR_A(config-tunnel)#mtu 1430
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
NXR_A(config-tunnel)#exit
NXR_A(config)#dns
NXR_A(config-dns)#service enable
NXR_A(config-dns)#exit
NXR_A(config)#exit
NXR_A#save config
```

〔NXR_B の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_B
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
NXR_B(config-if)#exit
NXR_B(config)#ip route 0.0.0.0/0 tunnel 1
NXR_B(config)#interface ppp 0
NXR_B(config-ppp)#ip address 10.10.20.1/32
NXR_B(config-ppp)#ip send-source
NXR_B(config-ppp)#ip tcp adjust-mss auto
NXR_B(config-ppp)#no ip redirects
NXR_B(config-ppp)#ppp username test2@centurysys password test2pass
NXR_B(config-ppp)#exit
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
NXR_B(config-if)#exit
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode gre
NXR_B(config-tunnel)#description NXR_A
NXR_B(config-tunnel)#tunnel source 10.10.20.1
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
NXR_B(config-tunnel)#mtu 1430
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#tunnel ttl 255
NXR_B(config-tunnel)#exit
NXR_B(config)#dns
NXR_B(config-dns)#service enable
NXR_B(config-dns)#exit
NXR_B(config)#exit
NXR_B#save config
```

〔NXR_C の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_C
NXR_C(config)#interface ethernet 0
NXR_C(config-if)#ip address 192.168.30.1/24
NXR_C(config-if)#exit
NXR_C(config)#ip route 0.0.0.0/0 tunnel 1
NXR_C(config)#interface ppp 0
NXR_C(config-ppp)#ip address 10.10.30.1/32
NXR_C(config-ppp)#ip send-source
NXR_C(config-ppp)#ip tcp adjust-mss auto
NXR_C(config-ppp)#no ip redirects
NXR_C(config-ppp)#ppp username test3@centurysys password test3pass
NXR_C(config-ppp)#exit
NXR_C(config)#interface ethernet 1
NXR_C(config-if)#no ip address
NXR_C(config-if)#pppoe-client ppp 0
NXR_C(config-if)#exit
NXR_C(config)#interface tunnel 1
NXR_C(config-tunnel)#tunnel mode gre
NXR_C(config-tunnel)#description NXR_A
NXR_C(config-tunnel)#tunnel source 10.10.30.1
NXR_C(config-tunnel)#tunnel destination 10.10.10.1
NXR_C(config-tunnel)#mtu 1430
NXR_C(config-tunnel)#ip tcp adjust-mss auto
NXR_C(config-tunnel)#tunnel ttl 255
NXR_C(config-tunnel)#exit
NXR_C(config)#dns
NXR_C(config-dns)#service enable
NXR_C(config-dns)#exit
NXR_C(config)#exit
NXR_C#save config
```

【 設定例解説 】

〔NXR_A の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_A
```

ホスト名を NXR_A と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.10.1/24 を設定します。

```
NXR_A(config-if)#ip access-linkdown
```

リンクダウン状態でも LAN 側(ethernet0)インタフェースの IP アドレスに通信することができるように設定します。

3. ＜スタティックルート設定＞

```
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。
ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。
ここでは LAN_B(192.168.20.0/24)向けのルートのゲートウェイとして tunnel 1 を設定します。

```
NXR_A(config)#ip route 192.168.30.0/24 tunnel 2
```

GRE(IPinIP)で使用するスタティックルートを設定します。
ここでは LAN_C(192.168.30.0/24)向けのルートのゲートウェイとして tunnel 2 を設定します。

```
NXR_A(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。ゲートウェイとして ppp0 インタフェースを設定します。

4. ＜WAN1(インターネット)側(ppp0)インタフェース設定＞

```
NXR_A(config)#interface ppp 0
```

WAN1側(ppp0)インタフェースを設定します。
ppp0 インタフェースがインターネット接続用のインタフェースとなります。

```
NXR_A(config-ppp)#ip address negotiated
```

本設定例では動的 IP アドレスが割り当てられるため、IP アドレスとして negotiated を設定します。

```
NXR_A(config-ppp)#ip masquerade
```

IP マスカレードを設定します。

```
NXR_A(config-ppp)#ip spi-filter
```

ステートフルパケットインスペクションを設定します。
ステートフルパケットインスペクションはパケットを監視してパケットフィルタリング項目を随時変更する機能で、動的パケットフィルタリング機能として利用できます。
該当インタフェースでこの設定を有効にした場合、通常そのインタフェースで受信したパケットは全て破棄されますが、そのインタフェースから送信されたパケットに対応する戻りパケットに対してはアクセスを許可します。
これにより自動的に WAN からの不要なアクセスを制御することが可能です。

```
NXR_A(config-ppp)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。
TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_A(config-ppp)#no ip redirects
```

ICMP リダイレクト機能を無効に設定します。

```
NXR_A(config-ppp)#ppp username test@centurysys password testpass
```

PPPoE 接続で使用するユーザ ID とパスワードを設定します。
ここではユーザ ID を test@centurysys, パスワードを testpass とします。

5. ＜WAN2(閉域網)側(ppp1)インタフェース設定＞

```
NXR_A(config)#interface ppp 1
```

WAN2側(ppp1)インタフェースを設定します。

ppp1 インタフェースが閉域網接続用のインタフェースとなります。

```
NXR_A(config-ppp)#ip address 10.10.10.1/32
```

IP アドレスを 10.10.10.1/32 に設定します。

```
NXR_A(config-ppp)#ip send-source
```

ip send-source を設定します。

ip send-source は PPP インタフェースに設定されている IP アドレスを送信元 IP アドレスとするパケットを出力する際、ルーティングテーブルで指定されたインタフェースではなく、必ず IP アドレスの所有者である PPP インタフェースから出力する設定です。

```
NXR_A(config-ppp)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_A(config-ppp)#no ip redirects
```

ICMP リダイレクト機能を無効に設定します。

```
NXR_A(config-ppp)#ppp username test1@centurysys password test1pass
```

PPPoE 接続で使用するユーザ ID とパスワードを設定します。

ここではユーザ ID を test1@centurysys, パスワードを test1pass とします。

6. ＜ethernet1 インタフェース設定＞

```
NXR_A(config)#interface ethernet 1
```

ethernet1 インタフェースを設定します。

```
NXR_A(config-if)#no ip address
```

ethernet1 インタフェースに IP アドレスを割り当てない設定をします。

PPPoE 接続でプロバイダ等から割り当てられる IP アドレスはイーサネットインタフェースではなく PPP インタフェースに割り当てられますので、PPPoE のみで使用する場合は IP アドレスの設定は不要です。

```
NXR_A(config-if)#pppoe-client ppp 0
```

ethernet1 インタフェース上で ppp0 インタフェースを使用するための設定をします。

PPPoE で PPP インタフェースを使用する場合は、pppoe-client コマンドによるインタフェース設定での登録が必要になります。

```
NXR_A(config-if)#pppoe-client ppp 1
```

ethernet1 インタフェース上で ppp1 インタフェースを使用するための設定をします。

7. ＜tunnel1 インタフェース設定＞

```
NXR_A(config)#interface tunnel 1
```

tunnel1 インタフェースを設定します。

```
NXR_A(config-tunnel)#tunnel mode gre
```

トンネルインタフェースで使用するトンネルモードを設定します。

トンネルインタフェースを GRE で使用する場合は、gre と設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

```
NXR_A(config-tunnel)#description NXR_B
```

このトンネルの説明として、ここでは NXR_B と設定します。

```
NXR_A(config-tunnel)#tunnel source 10.10.10.1
```

トンネルの送信元 IP アドレスとして WAN2側(ppp1)IP アドレス 10.10.10.1 を設定します。

```
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
```

トンネルのあて先 IP アドレスとして NXR_B の WAN 側 IP アドレス 10.10.20.1 を設定します。

```
NXR_A(config-tunnel)#mtu 1430
```

トンネルインタフェースの MTU 値として 1430 を設定します。

(☞) この設定例では PPPoE 接続時の WAN 側インタフェースの MTU 値を 1454 とし、かつ GRE のオプションは使用していないことを想定しています。

```
NXR_A(config-tunnel)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

(☞) ここでは MSS 値の算出にはトンネルインタフェースの MTU 値を使用します。

```
NXR_A(config-tunnel)#tunnel ttl 255
```

TTL 値を 255 に設定します。

8. ＜tunnel2 インタフェース設定＞

```
NXR_A(config)#interface tunnel 2
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_C
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.30.1
NXR_A(config-tunnel)#mtu 1430
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
```

tunnel2 インタフェースを設定します。

このトンネルの説明として、ここでは NXR_C と設定します。

トンネルの送信元 IP アドレスとして WAN2側(ppp1)IP アドレス 10.10.10.1、あて先 IP アドレスとして NXR_C の WAN 側 IP アドレス 10.10.30.1 を設定します。

その他の設定は 7. ＜tunnel1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

9. ＜DNS 設定＞

```
NXR_A(config)#dns
NXR_A(config-dns)#service enable
```

DNS サービスを有効に設定します。

〔NXR_B の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_B
```

ホスト名を NXR_B と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.20.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_B(config)#ip route 0.0.0.0/0 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは拠点間通信だけでなくインターネット向けのパケットも GRE(IPinIP)トンネルを経由するよう、デフォルトルートを設定しゲートウェイとして tunnel 1 を設定します。

4. ＜WAN(閉域網)側(ppp0)インタフェース設定＞

```
NXR_B(config)#interface ppp 0
```

WAN(閉域網)側(ppp0)インタフェースを設定します。

```
NXR_B(config-ppp)#ip address 10.10.20.1/32
```

IP アドレスを 10.10.20.1/32 に設定します。

```
NXR_B(config-ppp)#ip send-source
```

ip send-source を設定します。

ip send-source は PPP インタフェースに設定されている IP アドレスを送信元 IP アドレスとするパケットを出力する際、ルーティングテーブルで指定されたインタフェースではなく、必ず IP アドレスの所有者である PPP インタフェースから出力する設定です。

```
NXR_B(config-ppp)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_B(config-ppp)#no ip redirects
```

ICMP リダイレクト機能を無効に設定します。

```
NXR_B(config-ppp)#ppp username test2@centurysys password test2pass
```

PPPoE 接続で使用するユーザ ID とパスワードを設定します。

ここではユーザ ID を test2@centurysys, パスワードを test2pass とします。

5. ＜ethernet1 インタフェース設定＞

```
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
```

ethernet1 インタフェースを設定します。

この設定は NXR_A の 6. ＜ethernet1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

6. ＜tunnel1 インタフェース設定＞

```
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode gre
NXR_B(config-tunnel)#description NXR_A
NXR_B(config-tunnel)#tunnel source 10.10.20.1
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
NXR_B(config-tunnel)#mtu 1430
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#tunnel ttl 255
```

tunnel1 インタフェースを設定します。

このトンネルの説明として、ここでは NXR_A と設定します。

トンネルの送信元 IP アドレスとして WAN 側(ppp0)インタフェースの IP アドレス 10.10.20.1、あて先 IP アドレスとして NXR_A の ppp1 インタフェースの IP アドレス 10.10.10.1 を設定します。

その他の設定は NXR_A の 7. ＜tunnel1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

7. ＜DNS 設定＞

```
NXR_B(config)#dns
NXR_B(config-dns)#service enable
```

DNS サービスを有効に設定します。

〔NXR_C の設定〕

1.<ホスト名の設定>

```
nxr120(config)#hostname NXR_C
```

ホスト名を NXR_C と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_C(config)#interface ethernet 0
NXR_C(config-if)#ip address 192.168.30.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.30.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_C(config)#ip route 0.0.0.0/0 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは拠点間通信だけでなくインターネット向けのパケットも GRE(IPinIP)トンネルを経由するよう、デフォルトルートを設定しゲートウェイとして tunnel 1 を設定します。

4. ＜WAN(閉域網)側(ppp0)インタフェース設定＞

```
NXR_C(config)#interface ppp 0
NXR_C(config-ppp)#ip address 10.10.30.1/32
NXR_C(config-ppp)#ip send-source
NXR_C(config-ppp)#ip tcp adjust-mss auto
NXR_C(config-ppp)#no ip redirects
NXR_C(config-ppp)#ppp username test3@centurysys password test3pass
```

WAN(閉域網)側(ppp0)インタフェースを設定します。

IP アドレスを 10.10.30.1/32 に設定します。

PPPoE 接続で使用するユーザ ID とパスワードとしてユーザ ID を test3@centurysys, パスワードを test3pass と設定します。

その他の設定は NXR_B の **4. ＜WAN(閉域網)側(ppp0)インタフェース設定＞**が参考になりますので、そちらをご参照下さい。

5. ＜ethernet1 インタフェース設定＞

```
NXR_C(config)#interface ethernet 1
NXR_C(config-if)#no ip address
NXR_C(config-if)#pppoe-client ppp 0
```

Ethernet1 インタフェースを設定します。

この設定は NXR_A の **6. ＜ethernet1 インタフェース設定＞**が参考になりますので、そちらをご参照下さい。

6. ＜tunnel1 インタフェース設定＞

```
NXR_C(config)#interface tunnel 1
NXR_C(config-tunnel)#tunnel mode gre
NXR_C(config-tunnel)#description NXR_A
NXR_C(config-tunnel)#tunnel source 10.10.30.1
NXR_C(config-tunnel)#tunnel destination 10.10.10.1
NXR_C(config-tunnel)#mtu 1430
NXR_C(config-tunnel)#ip tcp adjust-mss auto
NXR_C(config-tunnel)#tunnel ttl 255
```

tunnel1 インタフェースを設定します。

このトンネルの説明として、ここでは NXR_A と設定します。

トンネルの送信元 IP アドレスとして WAN 側(ppp0)インタフェースの IP アドレス 10.10.30.1、あて先 IP アドレスとして NXR_A の ppp1 インタフェースの IP アドレス 10.10.10.1 を設定します。

その他の設定は NXR_A の 7. ＜tunnel1 インタフェース設定＞が参考になりますので、そちらをご参照下さい。

7. ＜DNS 設定＞

```
NXR_C(config)#dns
NXR_C(config-dns)#service enable
```

DNS サービスを有効に設定します。

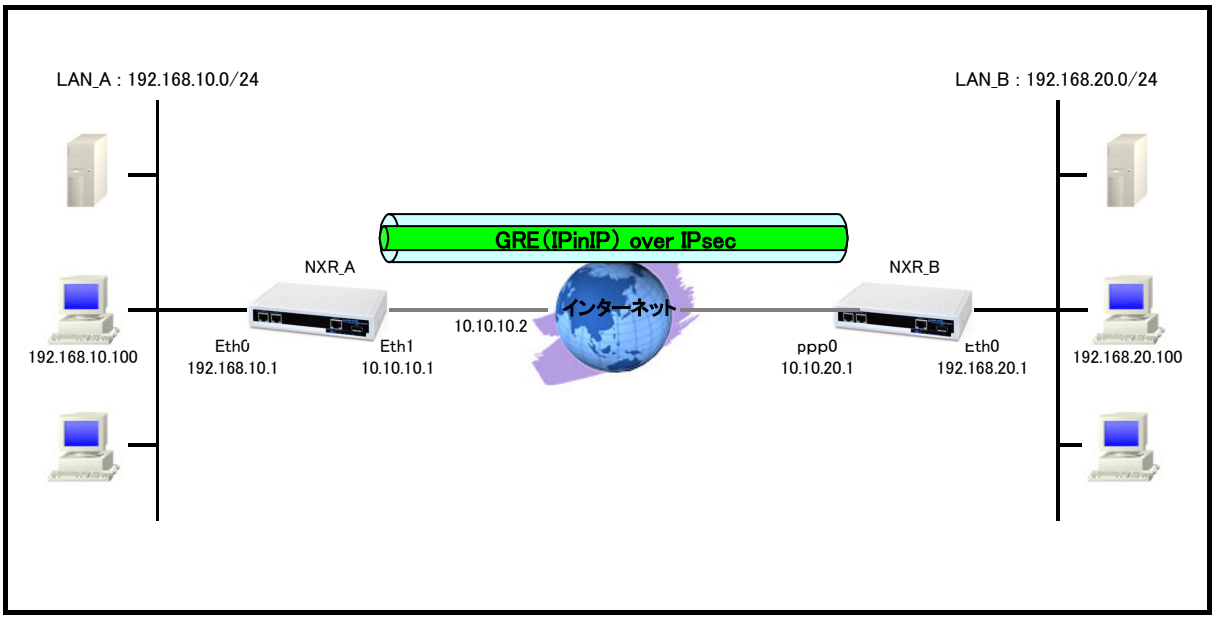
【 パソコンの設定例 】

	LAN A のパソコン	LAN B のパソコン	LAN C のパソコン
IP アドレス	192.168.10.100	192.168.20.100	192.168.30.100
サブネットマスク	255.255.255.0	255.255.255.0	255.255.255.0
デフォルトゲートウェイ	192.168.10.1	192.168.20.1	192.168.30.1
DNS サーバの IP アドレス	192.168.10.1	192.168.10.1	192.168.10.1

2-2. GRE(IPinIP)over IPsec 設定

GRE, IPinIP トンネルは暗号化を行いませんが、IPsec トンネル内に GRE, IPinIP トンネルを生成することも可能ですので、GRE, IPinIP を使用する場合でもセキュアな通信路を構築することができます。
※設定例は GRE ですが、設定例解説の中で IPinIP に関する補足を記載しています。

【 構成図 】



- IPsec は Policy Based IPsec の形式で設定します。
  Policy Based IPsec に関しては FutureNet NXR 設定例集 IPsec 編をご参照ください。
- IPsec はそれぞれの NXR が WAN 側固定 IP アドレスとなっていますので、フェーズ1ではメインモードで設定します。
- この例では WAN 側インタフェースで GRE パケットの出力をフィルタすることで、GRE パケットが IPsec トンネルがない場合に WAN 側インタフェースから出力されることを防止する設定を行っています。
- ルータ配下の端末からのインターネットアクセス用に NXR_A では ethernet1 インタフェース、NXR_B では ppp0 インタフェースで IP マスカレード, ステートフルパケットインスペクションを設定します。

【 設定例 】

〔NXR_A の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_A
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
NXR_A(config-if)#exit
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
NXR_A(config)#ip route 0.0.0.0/0 10.10.10.2
NXR_A(config)#ip access-list eth1_in permit 10.10.20.1 10.10.10.1 udp 500 500
NXR_A(config)#ip access-list eth1_in permit 10.10.20.1 10.10.10.1 50
NXR_A(config)#ip access-list eth1_out deny 10.10.10.1 any 47
NXR_A(config)#ipsec access-list NXR_B ip host host
NXR_A(config)#ipsec local policy 1
NXR_A(config-ipsec-local)#address ip
NXR_A(config-ipsec-local)#exit
NXR_A(config)#ipsec isakmp policy 1
NXR_A(config-ipsec-isakmp)#description NXR_B
NXR_A(config-ipsec-isakmp)#authentication pre-share ipseckey
NXR_A(config-ipsec-isakmp)#hash sha1
NXR_A(config-ipsec-isakmp)#encryption aes128
NXR_A(config-ipsec-isakmp)#group 5
NXR_A(config-ipsec-isakmp)#lifetime 10800
NXR_A(config-ipsec-isakmp)#isakmp-mode main
NXR_A(config-ipsec-isakmp)#remote address ip 10.10.20.1
NXR_A(config-ipsec-isakmp)#keepalive 10 3 periodic restart
NXR_A(config-ipsec-isakmp)#local policy 1
NXR_A(config-ipsec-isakmp)#exit
NXR_A(config)#ipsec tunnel policy 1
NXR_A(config-ipsec-tunnel)#description NXR_B
NXR_A(config-ipsec-tunnel)#negotiation-mode auto
NXR_A(config-ipsec-tunnel)#set transform esp-aes128 esp-sha1-hmac
NXR_A(config-ipsec-tunnel)#set pfs group5
NXR_A(config-ipsec-tunnel)#set sa lifetime 3600
NXR_A(config-ipsec-tunnel)#set key-exchange isakmp 1
NXR_A(config-ipsec-tunnel)#match address NXR_B
NXR_A(config-ipsec-tunnel)#exit
NXR_A(config)#interface ethernet 1
NXR_A(config-if)#ip address 10.10.10.1/29
NXR_A(config-if)#ip masquerade
NXR_A(config-if)#ip access-group in eth1_in
NXR_A(config-if)#ip access-group out eth1_out
NXR_A(config-if)#ip spi-filter
NXR_A(config-if)#ip tcp adjust-mss auto
NXR_A(config-if)#no ip redirects
NXR_A(config-if)#ipsec policy 1
NXR_A(config-if)#exit
NXR_A(config)#interface tunnel 1
NXR_A(config-tunnel)#tunnel mode gre
NXR_A(config-tunnel)#description NXR_B
NXR_A(config-tunnel)#tunnel source 10.10.10.1
NXR_A(config-tunnel)#tunnel destination 10.10.20.1
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#tunnel ttl 255
NXR_A(config-tunnel)#exit
NXR_A(config)#dns
NXR_A(config-dns)#service enable

----- 次のページに続きがあります -----
```

----- 前のページからの続きです -----

```
NXR_A(config-dns)#address 10.10.10.3
NXR_A(config-dns)#exit
NXR_A(config)#exit
NXR_A#save config
```

〔NXR_B の設定〕

```
nxr120#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
nxr120(config)#hostname NXR_B
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
NXR_B(config-if)#exit
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1
NXR_B(config)#ip route 0.0.0.0/0 ppp 0
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.20.1 udp 500 500
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.20.1 50
NXR_B(config)#ip access-list ppp0_out deny 10.10.20.1 10.10.10.1 47
NXR_B(config)#ipsec access-list NXR_A ip host host
NXR_B(config)#ipsec local policy 1
NXR_B(config-ipsec-local)#address ip
NXR_B(config-ipsec-local)#exit
NXR_B(config)#ipsec isakmp policy 1
NXR_B(config-ipsec-isakmp)#description NXR_A
NXR_B(config-ipsec-isakmp)#authentication pre-share ipseckey
NXR_B(config-ipsec-isakmp)#hash sha1
NXR_B(config-ipsec-isakmp)#encryption aes128
NXR_B(config-ipsec-isakmp)#group 5
NXR_B(config-ipsec-isakmp)#lifetime 10800
NXR_B(config-ipsec-isakmp)#isakmp-mode main
NXR_B(config-ipsec-isakmp)#remote address ip 10.10.10.1
NXR_B(config-ipsec-isakmp)#keepalive 10 3 periodic restart
NXR_B(config-ipsec-isakmp)#local policy 1
NXR_B(config-ipsec-isakmp)#exit
NXR_B(config)#ipsec tunnel policy 1
NXR_B(config-ipsec-tunnel)#description NXR_A
NXR_B(config-ipsec-tunnel)#negotiation-mode auto
NXR_B(config-ipsec-tunnel)#set transform esp-aes128 esp-sha1-hmac
NXR_B(config-ipsec-tunnel)#set pfs group5
NXR_B(config-ipsec-tunnel)#set sa lifetime 3600
NXR_B(config-ipsec-tunnel)#set key-exchange isakmp 1
NXR_B(config-ipsec-tunnel)#match address NXR_A
NXR_B(config-ipsec-tunnel)#exit
NXR_B(config)#interface ppp 0
NXR_B(config-ppp)#ip address 10.10.20.1/32
NXR_B(config-ppp)#ip masquerade
NXR_B(config-ppp)#ip access-group in ppp0_in
NXR_B(config-ppp)#ip access-group out ppp0_out
NXR_B(config-ppp)#ip spi-filter
NXR_B(config-ppp)#ip tcp adjust-mss auto
NXR_B(config-ppp)#no ip redirects
NXR_B(config-ppp)#ppp username test2@centurysys password test2pass
NXR_B(config-ppp)#ipsec policy 1
NXR_B(config-ppp)#exit
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
NXR_B(config-if)#exit
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode gre
NXR_B(config-tunnel)#description NXR_A
NXR_B(config-tunnel)#tunnel source 10.10.20.1
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#tunnel ttl 255
```

----- 次のページに続きがあります -----

```
----- 前のページからの続きです -----

NXR_B(config-tunnel)#exit
NXR_B(config)#dns
NXR_B(config-dns)#service enable
NXR_B(config-dns)#exit
NXR_B(config)#exit
NXR_B#save config
```

【 設定例解説 】

〔NXR_A の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_A
```

ホスト名を NXR_A と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.10.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは LAN_B(192.168.20.0/24)向けのルートのゲートウェイとして tunnel 1 を設定します。

```
NXR_A(config)#ip route 0.0.0.0/0 10.10.10.2
```

デフォルトルートを設定します。ゲートウェイアドレスは上位ルータの IP アドレスを設定します。

ここではゲートウェイアドレスとして 10.10.10.2 を設定します。

4. ＜IP アクセスリスト設定＞

```
NXR_A(config)#ip access-list eth1_in permit 10.10.20.1 10.10.10.1 udp 500 500
NXR_A(config)#ip access-list eth1_in permit 10.10.20.1 10.10.10.1 50
```

フィルタの動作を規定するルールリストを作成します。

ここでは eth1_in と eth1_out の 2 つの IP アクセスリスト名を設定します。

まず IP アクセスリスト名 eth1_in を設定します。

一行目の設定は送信元 IP アドレス 10.10.20.1 あて先 IP アドレス 10.10.10.1 送信元 UDP ポート番号 500 あて先 UDP ポート番号 500 のパケットを許可するための設定です。

二行目の設定は送信元 IP アドレス 10.10.20.1 あて先 IP アドレス 10.10.10.1 プロトコル番号 50(ESP)のパケットを許可するための設定です。

(☞) この設定は IPsec のネゴシエーションおよび IPsec SA 確立後の ESP パケットによる通信を許可するための設定となります。

```
NXR_A(config)#ip access-list eth1_out deny 10.10.10.1 any 47
```

次に IP アクセスリスト名 eth1_out を設定します。

この設定は送信元 IP アドレス 10.10.10.1 プロトコル番号 47(GRE)のパケットを破棄するための設定です。

(☞) この設定は IPsec 未確立時などに WAN 側(ethernet1)インタフェースから直接 GRE パケットを出力しないようにするための設定となります。

(☞) IPinIP を利用する場合はプロトコル番号 4 を設定します。

上記 IP アクセスリスト設定は全て WAN 側(ethernet1)インタフェース設定で登録します。

(☞) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングを行うインタフェースでの登録が必要になります。

5. ＜IPsec アクセスリスト設定＞

```
NXR_A(config)#ipsec access-list NXR_B ip host host
```

IPsec アクセスリスト名を NXR_B とし、送信元 IP アドレス host, あて先 IP アドレス host を設定します。

(☞) 送信元 IP アドレスに host を設定した場合は、IPsec のエンドポイントの IP アドレス(通常 WAN 側 IP アドレス)を設定したのと同じ意味になります。またあて先 IP アドレス host と設定した場合は対向ルータの IPsec のエンドポイントの IP アドレス(通常対向ルータの WAN 側 IP アドレス)を設定したのと同じ意味になります。

6. ＜IPsec ローカルポリシー設定＞

```
NXR_A(config)#ipsec local policy 1
```

IPsec ローカルポリシー1を設定します。

```
NXR_A(config-ipsec-local)#address ip
```

IPsec トンネルの送信元 IP アドレスを指定します。

この IP アドレスは、インタフェース設定で ipsec policy 1 と指定したインタフェースの IP アドレスが自動的に設定されます。

7. ＜IPsec ISAKMP ポリシー設定1＞

```
NXR_A(config)#ipsec isakmp policy 1
```

NXR_B との IPsec 接続で使用する ISAKMP ポリシー1を設定します。

```
NXR_A(config-ipsec-isakmp)#description NXR_B
```

ISAKMP ポリシー1の説明として、ここでは NXR_B と設定します。

```
NXR_A(config-ipsec-isakmp)#authentication pre-share ipseckey
```

認証方式として pre-share(事前共有鍵)を選択し、事前共有鍵として ipseckey を設定します。

この設定は、対向の NXR_B と同じ値を設定する必要があります。

```
NXR_A(config-ipsec-isakmp)#hash sha1
```

認証アルゴリズムを設定します。ここでは sha1 を設定します。

NXR_A(config-ipsec-isakmp)#**encryption aes128**

暗号化アルゴリズムを設定します。ここでは aes128 を設定します。

NXR_A(config-ipsec-isakmp)#**group 5**

Diffie-Hellman(DH)グループを設定します。ここでは group5 を設定します。

NXR_A(config-ipsec-isakmp)#**lifetime 10800**

ISAKMP SA のライフタイムを設定します。ここでは 10800 秒を設定します。

NXR_A(config-ipsec-isakmp)#**isakmp-mode main**

フェーズ1のネゴシエーションモードを設定します。ここでは NXR_A, NXR_B ともに WAN 側 IP アドレスが固定 IP アドレスのため、メインモードを設定します。

NXR_A(config-ipsec-isakmp)#**remote address ip 10.10.20.1**

対向の NXR_B の WAN 側 IP アドレスを設定します。ここでは対向の NXR_B の WAN 側 IP アドレス 10.10.20.1 を設定します。

NXR_A(config-ipsec-isakmp)#**keepalive 10 3 periodic restart**

IKE KeepAlive(DPD)を設定します。DPD(Dead Peer Detection)は ISAKMP SA を監視する機能で、対向の NXR の WAN 側で障害が発生した場合などにそれを検知し、現在利用している SA を削除したり SA を削除して再ネゴシエーションを行ったりするなどの機能があります。
なお DPD は常に定期的に送信されるわけではなく、対向の NXR より IPsec パケットを受信している場合は、DPD パケットの送信は行われません。
ここでは監視を 10 秒間隔で 3 回リトライを行い、keepalive 失敗時に SA を削除し、IKE のネゴシエーションを開始するように設定します。

NXR_A(config-ipsec-isakmp)#**local policy 1**

関連づけを行う IPsec ローカルポリシーを指定します。
ここでは IPsec ローカルポリシー1と関連づけを行います。

8. ＜IPsec トンネルポリシー設定1＞

NXR_A(config)#**ipsec tunnel policy 1**

NXR_B との IPsec 接続で使用するトンネルポリシー1を設定します。

NXR_A(config-ipsec-tunnel)#**description NXR_B**

トンネルポリシー1の説明として、ここでは NXR_B と設定します。

```
NXR_A(config-ipsec-tunnel)#negotiation-mode auto
```

IPsec ポリシーのネゴシエーションモードを設定します。この設定によってネゴシエーションを自ら開始したり、逆にいかなる場合も自らネゴシエーションを開始しないという設定が可能です。
ここではネゴシエーションモードを auto に設定します。これによりこちらからネゴシエーションを開始することができます。

```
NXR_A(config-ipsec-tunnel)#set transform esp-aes128 esp-sha1-hmac
```

IPsec トンネルポリシーで使用するトランスフォーム(プロポーザル)を設定します。
ここでは、暗号化アルゴリズム esp-aes128, 認証アルゴリズム esp-sha1-hmac を設定します。

```
NXR_A(config-ipsec-tunnel)#set pfs group5
```

PFS(Perfect Forward Secrecy)の設定とそれに伴う DH グループを設定します。
ここでは PFS を使用し、DH グループとして group5 を設定します。

```
NXR_A(config-ipsec-tunnel)#set sa lifetime 3600
```

IPsec SA のライフタイムを設定します。ここでは 3600 秒を設定します。

```
NXR_A(config-ipsec-tunnel)#set key-exchange isakmp 1
```

関連づけを行う ISAKMP ポリシーを指定します。
ここでは ISAKMP ポリシー1と関連づけを行います。

```
NXR_A(config-ipsec-tunnel)#match address NXR_B
```

使用する IPsec アクセスリストを指定します。
ここでは IPsec アクセスリスト NXR_B を設定します。

9. ＜WAN 側(ethernet1)インタフェース設定＞

```
NXR_A(config)#interface ethernet 1
```

WAN 側(ethernet1)インタフェースを設定します。

```
NXR_A(config-if)#ip address 10.10.10.1/29
```

IP アドレスを 10.10.10.1/29 に設定します。

```
NXR_A(config-if)#ip masquerade
```

IP マスカレードを設定します。

```
NXR_A(config-if)#ip access-group in eth1_in
```

IP アクセスリスト設定で設定した eth1_in を in フィルタに適用します。これにより ethernet1 インタフェースで受信した NXR 自身宛のパケットに対して IP アクセスリストによるチェックが行われます。

```
NXR_A(config-if)#ip access-group out eth1_out
```

IP アクセスリスト設定で設定した eth1_out を out フィルタに適用します。これにより NXR 自身が ethernet1 インタフェースから送信するパケットに対して IP アクセスリストによるチェックが行われます。

NXR_A(config-if)#**ip spi-filter**

ステートフルパケットインスペクションを設定します。

ステートフルパケットインスペクションはパケットを監視してパケットフィルタリング項目を随時変更する機能で、動的パケットフィルタリング機能として利用できます。

該当インタフェースでこの設定を有効にした場合、通常そのインタフェースで受信したパケットは全て破棄されますが、そのインタフェースから送信されたパケットに対応する戻りパケットに対してはアクセスを許可します。

これにより自動的に WAN からの不要なアクセスを制御することが可能です。

NXR_A(config-if)#**ip tcp adjust-mss auto**

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

NXR_A(config-if)#**no ip redirects**

ICMP リダイレクト機能を無効に設定します。

NXR_A(config-if)#**ipsec policy 1**

IPsec ローカルポリシー1を適用します。これによりこのインタフェースが IPsec トンネルのエンドポイントとなります。

10.. ＜tunnel1 インタフェース設定＞

NXR_A(config)#**interface tunnel 1**

tunnel1 インタフェースを設定します。

NXR_A(config-tunnel)#**tunnel mode gre**

トンネルインタフェースで使用するトンネルモードを設定します。

トンネルインタフェースを GRE で使用する場合は、gre と設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

NXR_A(config-tunnel)#**description NXR_B**

このトンネルの説明として、ここでは NXR_B と設定します。

NXR_A(config-tunnel)#**tunnel source 10.10.10.1**

トンネルの送信元 IP アドレスとして WAN 側(ethernet1)IP アドレス 10.10.10.1 を設定します。

NXR_A(config-tunnel)#**tunnel destination 10.10.20.1**

トンネルのあて先 IP アドレスとして NXR_B の WAN 側 IP アドレス 10.10.20.1 を設定します。

NXR_A(config-tunnel)#**ip tcp adjust-mss auto**

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを

転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_A(config-tunnel)#tunnel ttl 255
```

TTL 値を 255 に設定します。

11. ＜DNS 設定＞

```
NXR_A(config)#dns
NXR_A(config-dns)#service enable
```

DNS サービスを有効に設定します。

```
NXR_A(config-dns)#address 10.10.10.3
```

DNS サーバの IP アドレスとしてこの例では外部の DNS サーバ 0.10.10.3 を設定します。

〔NXR_B の設定〕

1. <ホスト名の設定>

```
nxr120(config)#hostname NXR_B
```

ホスト名を NXR_B と設定します。

2. ＜LAN 側(ethernet0)インタフェース設定＞

```
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスに 192.168.20.1/24 を設定します。

3. ＜スタティックルート設定＞

```
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1
```

GRE(IPinIP)で使用するスタティックルートを設定します。

ここで設定したあて先 IP アドレスにマッチしたパケットが GRE(IPinIP)のカプセル化対象となります。

ゲートウェイは GRE(IPinIP)で使用するトンネルインタフェースを設定します。

ここでは LAN_A(192.168.10.0/24)向けのルートのゲートウェイとして tunnel 1 を設定します。

```
NXR_B(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。ゲートウェイとして ppp0 インタフェースを設定します。

4. ＜IP アクセスリスト設定＞

```
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.20.1 udp 500 500
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 10.10.20.1 50
```

フィルタの動作を規定するルールリストを作成します。

ここでは ppp0_in と ppp0_out の 2 つの IP アクセスリスト名を設定します。

まず IP アクセスリスト名 ppp0_in を設定します。

一行目の設定は送信元 IP アドレス 10.10.10.1 あて先 IP アドレス 10.10.20.1 送信元 UDP ポート番号 500 あて先 UDP ポート番号 500 のパケットを許可するための設定です。

二行目の設定は送信元 IP アドレス 10.10.10.1 あて先 IP アドレス 10.10.20.1 プロトコル番号 50(ESP)のパケットを許可するための設定です。

(☞) この設定は IPsec のネゴシエーションおよび IPsec SA 確立後の ESP パケットによる通信を許可するため

の設定となります。

```
NXR_B(config)#ip access-list ppp0_out deny 10.10.20.1 10.10.10.1 47
```

次に IP アクセスリスト名 ppp0_out を設定します。

この設定は送信元 IP アドレス 10.10.20.1 あて先 IP アドレス 10.10.10.1 プロトコル番号 47(GRE)のパケットを破棄するための設定です。

(☞) この設定は IPsec 未確立時などに WAN 側(ppp0)インタフェースから直接 GRE パケットを出力しないようにするための設定となります。

(☞) IPinIP を利用する場合はプロトコル番号 4 を設定します。

上記 IP アクセスリスト設定は全て WAN 側(ppp0)インタフェース設定で登録します。

(☞) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングを行うインタフェースでの登録が必要になります。

5. ＜IPsec アクセスリスト設定＞

```
NXR_B(config)#ipsec access-list NXR_A ip host host
```

IPsec アクセスリスト名を NXR_A とし、送信元 IP アドレス host, あて先 IP アドレス host を設定します。

(☞) 送信元 IP アドレスに host を設定した場合は、IPsec のエンドポイントの IP アドレス(通常 WAN 側 IP アドレス)を設定したのと同じ意味になります。またあて先 IP アドレス host と設定した場合は対向ルータの IPsec のエンドポイントの IP アドレス(通常対向ルータの WAN 側 IP アドレス) を設定したのと同じ意味になります。

6. ＜IPsec ローカルポリシー設定＞

```
NXR_B(config)#ipsec local policy 1
```

IPsec ローカルポリシー1を設定します。

```
NXR_B(config-ipsec-local)#address ip
```

IPsec トンネルの送信元 IP アドレスを指定します。

この IP アドレスは、インタフェース設定で ipsec policy 1 と指定したインタフェースの IP アドレスが自動的に設定されます。

7. ＜IPsec ISAKMP ポリシー設定1＞

```
NXR_B(config)#ipsec isakmp policy 1
NXR_B(config-ipsec-isakmp)#description NXR_A
NXR_B(config-ipsec-isakmp)#authentication pre-share ipseckey
NXR_B(config-ipsec-isakmp)#hash sha1
NXR_B(config-ipsec-isakmp)#encryption aes128
NXR_B(config-ipsec-isakmp)#group 5
NXR_B(config-ipsec-isakmp)#lifetime 10800
NXR_B(config-ipsec-isakmp)#isakmp-mode main
NXR_B(config-ipsec-isakmp)#remote address ip 10.10.10.1
NXR_B(config-ipsec-isakmp)#keepalive 10 3 periodic restart
NXR_B(config-ipsec-isakmp)#local policy 1
```

NXR_B との IPsec 接続で使用する ISAKMP ポリシー1を設定します。

ISAKMP ポリシー1の説明として、ここでは NXR_B と設定します。
対向の NXR_A の WAN 側 IP アドレス 10.10.20.1 を設定します。
その他の設定は NXR_A の 7. ＜IPsec ISAKMP ポリシー設定1＞が参考になりますので、そちらをご参照下さい。

8. ＜IPsec トンネルポリシー設定1＞

```
NXR_B(config)#ipsec tunnel policy 1
NXR_B(config-ipsec-tunnel)#description NXR_A
NXR_B(config-ipsec-tunnel)#negotiation-mode auto
NXR_B(config-ipsec-tunnel)#set transform esp-aes128 esp-sha1-hmac
NXR_B(config-ipsec-tunnel)#set pfs group5
NXR_B(config-ipsec-tunnel)#set sa lifetime 3600
NXR_B(config-ipsec-tunnel)#set key-exchange isakmp 1
NXR_B(config-ipsec-tunnel)#match address NXR_A
```

NXR_B との IPsec 接続で使用するトンネルポリシー1を設定します。
トンネルポリシー1の説明として、ここでは NXR_B と設定します。
使用する IPsec アクセスリストとして IPsec アクセスリスト NXR_A を設定します。
その他の設定は NXR_A の 8. ＜IPsec トンネルポリシー設定1＞が参考になりますので、そちらをご参照下さい。

9. ＜WAN 側(ppp0)インタフェース設定＞

```
NXR_B(config)#interface ppp 0
```

WAN 側(ppp0)インタフェースを設定します。

```
NXR_B(config-ppp)#ip address 10.10.20.1/32
```

IP アドレスを 10.10.20.1/32 に設定します。

```
NXR_B(config-ppp)#ip masquerade
```

IP マスカレードを設定します。

```
NXR_B(config-ppp)#ip access-group in ppp0_in
```

IP アクセスリスト設定で設定した ppp0_in を in フィルタに適用します。これにより ppp0 インタフェースで受信したパケット(NXR 自身宛)に対して IP アクセスリストによるチェックが行われます。

```
NXR_B(config-ppp)#ip access-group out ppp0_out
```

IP アクセスリスト設定で設定した ppp0_out を out フィルタに適用します。これにより NXR 自身が ppp0 インタフェースから送信するパケットに対して IP アクセスリストによるチェックが行われます。

```
NXR_B(config-ppp)#ip spi-filter
```

ステートフルパケットインスペクションを設定します。
ステートフルパケットインスペクションはパケットを監視してパケットフィルタリング項目を随時変更する機能で、動的パケットフィルタリング機能として利用できます。
該当インタフェースでこの設定を有効にした場合、通常そのインタフェースで受信したパケットは全て破棄されますが、そのインタフェースから送信されたパケットに対応する戻りパケットに対してはアクセスを許可します。

これにより自動的に WAN からの不要なアクセスを制御することが可能です。

```
NXR_B(config-ppp)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_B(config-ppp)#no ip redirects
```

ICMP リダイレクト機能を無効に設定します。

```
NXR_B(config-ppp)#ppp username test2@centurysys password test2pass
```

PPPoE 接続で使用するユーザ ID とパスワードを設定します。

ここではユーザ ID を test2@centurysys, パスワードを test2pass とします。

```
NXR_B(config-ppp)#ipsec policy 1
```

IPsec ローカルポリシー1を適用します。これによりこのインタフェースが IPsec トンネルのエンドポイントとなります。

10. ＜ethernet1 インタフェース設定＞

```
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
```

ethernet1 インタフェースを設定します。

11. ＜tunnel1 インタフェース設定＞

```
NXR_B(config)#interface tunnel 1
```

tunnel1 インタフェースを設定します。

```
NXR_B(config-tunnel)#tunnel mode gre
```

トンネルインタフェースで使用するトンネルモードを設定します。

トンネルインタフェースを GRE で使用する場合は、gre と設定します。

(☞) IPinIP を利用する場合はこのトンネルモードの設定で ipip と設定します。

```
NXR_B(config-tunnel)#description NXR_A
```

このトンネルの説明として、ここでは NXR_A と設定します。

```
NXR_B(config-tunnel)#tunnel source 10.10.20.1
```

トンネルの送信元 IP アドレスとして WAN 側(ppp0)10.10.20.1 を設定します。

```
NXR_B(config-tunnel)#tunnel destination 10.10.10.1
```

トンネルのあて先 IP アドレスとして NXR_A の WAN 側 IP アドレス 10.10.10.1 を設定します。

```
NXR_B(config-tunnel)#ip tcp adjust-mss auto
```

TCP MSS の調整機能をオートに設定します。

TCP MSS 調整機能は TCP のネゴシエーション時に MSS 値を調整することで、サイズの大きい TCP パケットを転送する際にフラグメントによるスループットの低下を抑制する場合に利用します。

```
NXR_B(config-tunnel)#tunnel ttl 255
```

TTL 値を 255 に設定します。

12. ＜DNS 設定＞

```
NXR_B(config)#dns
NXR_B(config-dns)#service enable
```

DNS サービスを有効に設定します。

【 パソコンの設定例 】

	LAN A のパソコン	LAN B のパソコン
IP アドレス	192.168.10.100	192.168.20.100
サブネットマスク	255.255.255.0	255.255.255.0
デフォルトゲートウェイ	192.168.10.1	192.168.20.1
DNS サーバの IP アドレス	192.168.10.1	192.168.20.1

付録

GRE・IPinIP 状態確認方法

show interface コマンドでどのトンネルインタフェースが GRE または IPinIP で動作しているかを確認することができます。

GRE を使用している場合の実行例

```
nxr120#show interface tunnel 1
tunnel1
          Link encap:GREIP Tunnel
          inet6 addr: fe80::xxx:xxxx:xxxx:xxx/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1476  Metric:1
          RX packets:50894 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72219 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:32969347 (31.4 Mb)  TX bytes:64757919 (61.7 Mb)


tunnel1: gre/ip  remote 10.10.20.1  local 10.10.10.1  ttl 255  tos inherit pmtudisc
RX: Packets     Bytes          Errors CsumErrs OutOfSeq Mcasts
     50894       32969347       0       0          0          0
TX: Packets     Bytes          Errors DeadLoop NoRoute  NoBufs
     72219       64757919       0       0          0          0
```

IPinIP を使用している場合の実行例

```
nxr120#show interface tunnel 1
tunnel1
          Link encap:IPIP Tunnel  HWaddr
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:14449 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30220 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:578078 (564.5 Kb)  TX bytes:32143136 (30.6 Mb)


tunnel1: ip/ip  remote 10.10.20.1  local 10.10.10.1  ttl 255  tos inherit pmtudisc
RX: Packets     Bytes          Errors CsumErrs OutOfSeq Mcasts
     14449       578078         0       0          0          0
TX: Packets     Bytes          Errors DeadLoop NoRoute  NoBufs
     30220       32143136       0       0          0          0
```

サポートデスクへのお問い合わせ

サポートデスクへのお問い合わせに関して

サポートデスクにお問い合わせ頂く際は、以下の情報をお知らせ頂けると効率よく対応させて頂くことが可能ですので、ご協力をお願い致します。

- ご利用頂いている NXR 製品の機種名,バージョン番号
- ご利用頂いている NXR 製品を含んだネットワーク構成
- 不具合の内容および不具合の再現手順（何を行った場合にどのような問題が発生したのかをできるだけ具体的にお知らせ下さい）
- ご利用頂いている NXR 製品での不具合発生時のログ（show syslog message）
- ご利用頂いている NXR 製品の設定ファイル，show tech-support コマンドの実行結果

サポートデスクのご利用に関して

電話サポート

電話番号:**0422－37－8926**

電話での対応は以下の時間帯で行います。

月曜日 ～ 金曜日 10:00 AM – 5:00 PM

ただし、国の定める祝祭日、弊社の定める年末年始は除きます。

電子メールサポート

E-mail: support@centurysys.co.jp

FAXサポート

FAX 番号:**0422－55－3373**

電子メール、FAX は 毎日 24 時間受け付けております。

ただし、システムのメンテナンスやビルの電源点検のため停止する場合があります。 その際は弊社ホームページ等にて事前にご連絡いたします。

FutureNet NXR 設定例集

GRE・IPinIP 編

Ver 1.0.0

2011 年 7 月

発行　センチュリー・システムズ株式会社

Copyright(c) 2009-2011 Century Systems Co., Ltd. All Rights Reserved.