Tunnel L3 - IPIP

1. Introduction : Pourquoi utiliser IPIP ?

1.1. À quoi sert un tunnel IPIP ?

Un tunnel **IPIP (IP-in-IP)** permet de créer un lien direct entre deux réseaux distants en encapsulant des paquets IP dans d'autres paquets IP. C'est le tunnel le plus simple et le plus léger qui existe. **Exemple concret :**

- Vous avez un réseau local derrière une box opérateur standard.
- Vous voulez une IP publique sur une ou des machines sur votre réseau local

1.2. Avantages de IPIP

- Extrêmement simple à configurer.
- Overhead minimal (seulement 20 octets d'en-tête supplémentaire).
- Compatible avec la plupart des routeurs et systèmes d'exploitation.

1.3. Inconvénients

- Pas de chiffrement natif.
- Pas de support de clé (contrairement à GRE).
- Supporte uniquement IPv4-in-IPv4 (pour IPv6-in-IPv4, utilisez SIT).
- Nécessite une configuration manuelle du routage.

---

2. Prérequis

2.1. Ce dont vous avez besoin

- Une **IP publique** (ou une redirection de port si derrière une box).
- Un **routeur compatible IPIP** (Linux, MikroTik, Cisco, etc.).

- Un **service Tunnel-IP**.

2.2. Ports et NAT

- IPIP n'est ni UDP, ni TCP **(protocole 4)**.
- Si le routeur final est derrière une box / NAT, il faut **rediriger le protocole IPIP** vers le routeur final, certaines box ont des ALG automatiques, mais il est souvent préférable de configurer une DMZ.

---

3. Étape 1 : Créer le tunnel sur Tunnel-IP.com

3.1. Accéder au tableau de bord

1. Connectez-vous à panel.tunnel-ip.com
2. Allez dans **"Tunnels"** > **"IPIP"**.
3. Remplissez les informations :
   - **Nom du tunnel** (ex : `Tunnel_X`).
   - **Endpoint** (IP publique du routeur / box, ex : `203.0.113.1`).
4. Validez. La plateforme s'occupera de choisir les IPs et de configurer le tunnel côté Tunnel-IP.com.
5. Attendez que le tunnel soit créé
6. Cliquez sur "Accéder" pour récupérer les détails du tunnel

3.2. Créer le subnet (Route côté plateforme)

1. Allez dans "Subnets"

2. Copiez le bloc d'IP publique et collez le dans le formulaire

3. Cliquez sur Créer

4. Une fois son statut à "Actif", la route est correctement installée sur les routeurs de la plateforme, il ne vous reste plus qu'à configurer le tunnel

---

4. Étape 2 : Configurer le tunnel

4.1. Exemple pour Linux (Ubuntu/Debian)

Étapes :

1. **Activer le forwarding IP** (pour permettre le routage) :

   ```
   echo 1 > /proc/sys/net/ipv4/ip_forward
   ```

   (Pour rendre permanent, ajoutez `net.ipv4.ip_forward=1` dans `/etc/sysctl.conf`.)
2. **Créer l'interface tunnel** :

   ```
   ip tunnel add tunnel0 mode ipip remote 172.16.126.33
   ip link set tunnel0 up
   ```

   - `tunnel0` : Nom de l'interface tunnel.
   - `remote 172.16.126.33` : IP distante (Tunnel-IP.com).
3. **Assigner une IP au tunnel** :

   ```
   ip addr add 100.64.0.6/30 dev tunnel0
   ```

   - `100.64.0.6/30` : IP locale du tunnel (ex : `100.64.0.5/30` est côté plateforme).

4.2. Exemple pour MikroTik

Étapes :

1. **Créer l'interface IPIP** :

   ```
   /interface/ipip/add remote-address=172.16.126.33 name=tun4
   ```

2. **Assigner une IP au tunnel** :

   ```
   /ip/address/add address=100.64.0.6/30 interface=tun4
   ```

4.3. Exemple pour Cisco (IOS/XE)

Étapes :

1. **Accéder au mode configuration** :

   ```
   enable
   configure terminal
   ```

2. **Créer l'interface tunnel** :

```
interface Tunnel0
 tunnel source 192.168.1.1  # IP locale du routeur
 tunnel destination 172.16.126.33  # IP distante
 tunnel mode ipip
 ip address 100.64.0.6 255.255.255.252  # IP locale du tunnel
```

   - `Tunnel0` : Nom de l'interface tunnel.
   - `tunnel mode ipip` : Active le mode IPIP.
3. **Activer l'interface** :

```
no shutdown
exit
```

4. **Sauvegarder la configuration** :

```
write memory
```

4.4. Exemple pour Arista (EOS)

Étapes :

1. **Accéder au mode configuration** :

```
enable
configure terminal
```

2. **Créer l'interface tunnel** :

```
interface Tunnel0
 tunnel source 192.168.1.1  # IP locale du routeur
 tunnel destination 172.16.126.33  # IP distante
 tunnel mode ipip
 ip address 100.64.0.6/30  # IP locale du tunnel
```

   - La syntaxe est très proche de Cisco, mais EOS est plus réactif pour les changements dynamiques.
3. **Activer l'interface** :

```
no shutdown
exit
```

4. **Sauvegarder la configuration** :

```
write memory
```

4.5. Exemple pour Juniper (JunOS)

Étapes :

1. **Accéder au mode configuration** :

```
edit
```

2. **Créer l'interface tunnel** :

```
set interfaces ip-0/0/0 tunnel source 192.168.1.1
set interfaces ip-0/0/0 tunnel destination 172.16.126.33
set interfaces ip-0/0/0 family inet address 100.64.0.6/30
```

   - `ip-0/0/0` : Nom de l'interface IPIP (peut varier selon le modèle).
   - Juniper utilise une syntaxe hiérarchique et des "commit" pour appliquer les changements.
3. **Activer l'interface** :

```
commit
```

---

5. Étape 3 : Router les IP publiques vers le tunnel

5.1. Pourquoi ?

La plateforme vous route un bloc d'IP publiques (ex : `198.51.100.0/30`) sur votre IP interne du tunnel, afin que ces IPs fonctionnent correctement, vous devez router ce bloc d'IP sur votre réseau et le trafic sortant par le tunnel.

Pour cela, 3 principales méthodes s'offrent à vous :

- **NAT 1:1** (1 IP publique → 1 IP privée)
- **LAN public** (utilisation directe des IPs publiques)
- **Routage en /32** (1 IP publique par machine)

5.2. Rappel : Architecture du Tunnel

```
Internet → [Infrastructure Tunnel-IP.com] → (Tunnel IPIP) → [Votre Routeur] → [Votre Réseau]
```

- La plateforme vous route un bloc public (ex: `198.51.100.0/30`).
- Votre routeur doit gérer ce bloc pour exposer vos services.

5.3. Cas d'Usage du Bloc /30

Un `/30` contient **4 adresses IP** :

- `198.51.100.0` : Adresse réseau (inutilisable en LAN).
- `198.51.100.1` : IP Utilisable
- `198.51.100.2` : IP Utilisable
- `198.51.100.3` : Adresse broadcast (inutilisable en LAN).

5.4. Méthode 1 : NAT 1:1 (Translation 1 IP publique ? 1 IP privée)

Avantages principaux : Permet de ne pas avoir à modifier l'adressage privé de votre réseau, et permet d'utiliser l'IP de réseau et broadcast.

Inconvénients : Adressage moins clair (Conversion IP publique -> privée), charge plus lourde sur le routeur (Le routeur est en charge de la translation NAT)

Schéma :

```
Internet → Infrastructure Tunnel-IP.com → [Votre Routeur] 198.51.100.0 → 192.168.1.100
(Privée)
```

Configurations :

Linux :

```
# Activer l'IP Forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
# Ajouter l'IP sur une interface (on utilise la loopback par exemple)
ip a add 198.51.100.0/32 dev lo

# 1:1 Static NAT
iptables -t nat -A PREROUTING -d 198.51.100.0 -j DNAT --to-destination 192.168.1.100
iptables -t nat -A POSTROUTING -s 192.168.1.100 -j SNAT --to-source 198.51.100.0
```

MikroTik :

```
/ip address add address=198.51.100.0/32 interface=eth0
/ip firewall nat add chain=dstnat dst-address=198.51.100.0 action=dst-nat to-
addresses=192.168.1.100
/ip firewall nat add chain=srcnat src-address=192.168.1.100 action=src-nat to-
addresses=198.51.100.0
```

Cisco :

```
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside

interface GigabitEthernet0/1
 ip address 198.51.100.0 255.255.255.255
 ip nat outside

ip nat inside source static 192.168.1.100 198.51.100.0
```

Arista :

```
interface Ethernet1
 ip address 192.168.1.1/24
 ip nat inside

interface Ethernet2
 ip address 198.51.100.0/32
 ip nat outside

ip nat inside source static 192.168.1.100 198.51.100.0
```

Juniper :

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.0/32

set security nat source rule-set OUTBOUND rule NAT1 match source-address 192.168.1.100/32
set security nat source rule-set OUTBOUND rule NAT1 then source-nat address 198.51.100.0
set security nat destination pool DNAT_POOL address 192.168.1.100/32
set security nat destination rule-set INBOUND rule NAT1 match destination-address
198.51.100.0/32
set security nat destination rule-set INBOUND rule NAT1 then destination-nat pool DNAT_POOL
```

5.5 Méthode 2 : LAN Public (Utilisation directe des IPs publiques)

Avantages principaux : Adressage propre, explicite

Inconvénients : Perte de 2 IPs utilisables (IP de réseau et IP de broadcast) + 1 IP est forcément assignée au routeur.

> **(Attention : Avec un /30 en LAN, vous n'avez qu'une seule IP utilisable. Pour plus d'IPs, prenez un bloc plus grand, ex:** `/29`**. ou faites du NAT 1:1 ou attribution en /32)**

Configurations :

Linux :

```
# Assigner l'IP de la passerelle (votre routeur)
ip addr add 198.51.100.1/30 dev eth1
```

MikroTik :

```
/ip address add address=198.51.100.1/30 interface=ether1
```

Cisco :

```
interface GigabitEthernet0/1
 ip address 198.51.100.1 255.255.255.252
 no shutdown
```

Juniper :

```
set interfaces ge-0/0/1 unit 0 family inet address 198.51.100.1/30
```

5.6. Méthode 3 : Routage en /32 (1 IP publique par machine)

Avantages principaux : Adressage propre, explicite, aucune perte d'IP

Inconvénients : Non supporté par Windows, mal supporté par certains routeurs / OS, 100% du trafic passe par le routeur

Configurations :

Linux :

```
# Sur le routeur :
ip route add 198.51.100.0/30 dev eth0  # eth0 = interface vers la Machine A

# Sur la Machine A :
ip addr add 203.0.113.1/32 dev eth0
ip route add default via 192.168.1.100 dev eth0 # IP de votre routeur local
# Si error lors de l'ajout de la route, il faut ajouter une règle :
# ip route add 192.168.1.100 dev eth0
```

MikroTik :

```
/ip route add dst-address=198.51.100.0/30 interface=LAN

# Sur la Machine A :
# Pareil que pour Linux, IP : 198.51.100.1 Gateway 192.168.1.100
```

Cisco :

```
ip route 198.51.100.0 255.255.255.252 GigabitEthernet0/0/1
```

Juniper :

```
set routing-options static route 198.51.100.0/30 interface ge-0/0/1
```

5.7. Router le trafic sortant par le tunnel

> Si vous ne faites pas cette étape, les IPs ne fonctionneront pas

Linux (ip route + policy routing)

```
# Création d'une table de routage
echo "143 tunnel" >> /etc/iproute2/rt_tables

# Routage vers l'extérieur via l'IP locale de la plateforme
ip route add default via 100.64.0.5 table tunnel
# Tout le trafic en provenance de nos IPs publiques, sortent via le tunnel
ip rule add from 198.51.100.0/30 table tunnel
```

Effet : Tout hôte du réseau `198.51.100.0/30` sort par le tunnel, les autres via la gateway par défaut.

> Si vous êtes à l'aise avec les VRF, il est aussi possible de faire avec :

```
ip link add vrf-IPIP type vrf table 143 # On créer la VRF
ip link set dev tunnel0 master vrf-IPIP # On ajoute le tunnel dans la VRF
ip route add default via 100.64.0.5 table 143 # On ajoute la route par défaut
ip link set dev eth0 master vrf-IPIP # On ajoute notre interface LAN dans la VRF (Attention si elle est partagée avec votre réseau local, le réseau local ne marchera plus, pour cela vous pouvez créer une interface dummy ou utiliser une interface séparée)
```

MikroTik

```
/routing/table/add name=IPIP-OUT fib
/routing/rule/add action=lookup-only-in-table table=IPIP-OUT src-address=198.51.100.0/30
/ip route add dst-address=0.0.0.0/0 gateway=100.64.0.5 routing-table=IPIP-OUT
```

> Si vous êtes à l'aise avec les VRF, il est aussi possible de faire avec :

```
/ip/vrf add name=IPIP-VRF interfaces=ipip1,ether1
/ip route add routing-table=IPIP-VRF dst-address=0.0.0.0/0 gateway=100.64.0.5
```

Cisco

```
ip access-list extended IPIP-OUT
 permit ip 198.51.100.0 0.0.0.3 any

route-map IPIP-ROUTE permit 10
 match ip address IPIP-OUT
 set ip next-hop 100.64.0.5

interface GigabitEthernet0/0
 ip policy route-map IPIP-ROUTE
```

On applique l'ACL sur l'interface LAN → tout ce bloc sort vers Tunnel0.

> Si vous êtes à l'aise avec les VRF, il est aussi possible de faire avec :

```
vrf definition OUT-IPIP
 rd 100:1
 !
 address-family ipv4
 exit-address-family
!
interface Tunnel0
 vrf forwarding OUT-IPIP
!
interface GigabitEthernet0/0/1
 vrf forwarding OUT-IPIP
!

! Default route de la VRF vers le vrai Next-Hop
ip route vrf OUT-IPIP 0.0.0.0 0.0.0.0 100.64.0.5
```

> La syntaxe Cisco peux varier selon les versions de l'OS

Arista

```
ip access-list IPIP-OUT
 10 permit ip 198.51.100.0/30 any

route-map IPIP-PBR permit 10
```

```
  match ip address IPIP-OUT
  set ip next-hop 100.64.0.5    # IP tunnel remote

interface Ethernet1
 ip policy route-map IPIP-PBR
```

> Si vous êtes à l'aise avec les VRF, il est aussi possible de faire avec :

```
vrf instance IPIP-VRF

interface Tunnel0
  vrf IPIP-VRF

interface Eth1
  vrf IPIP-VRF

ip route vrf IPIP-VRF 0.0.0.0/0 100.64.0.5
```

Juniper

```
set routing-instances IPIP-VRF instance-type virtual-router
set routing-instances IPIP-VRF interface ip-0/0/0.0
set routing-instances IPIP-VRF routing-options static route 0.0.0.0/0 next-hop 100.64.0.5
set firewall family inet filter PBR term IPIP from source-address 198.51.100.0/30
set firewall family inet filter PBR term IPIP then routing-instance IPIP-VRF
set firewall family inet filter PBR term DEFAULT then accept

set interfaces ge-0/0/1 unit 0 family inet filter input PBR
```

□□ Le trafic matché bascule dans la VRF = sortie via IPIP.

6. Commandes spécifiques par constructeur

Constructeur	Commande pour vérifier le tunnel	Commande pour vérifier le routage

Cisco	`show interface Tunnel0`	`show ip route`
Arista	`show interface Tunnel0`	`show ip route`
Juniper	`show interfaces ip-0/0/0`	`show route`
Linux	`ip tunnel show`	`ip route`
MikroTik	`/interface ipip print`	`/ip route print`

7. Vérification et Dépannage

7.1. Tester la connectivité

- **Depuis votre routeur** :

```
ping 100.64.0.5 # Ping de l'ip distante du tunnel
```

- **Depuis une machine locale** :

```
ping 8.8.8.8 # Vérification internet
curl -4 ifconfig.me # Vérification de l'IP sortante
```

7.2. Problèmes courants

Symptôme	Cause Possible	Solution
Le tunnel ne répond pas	Protocole 4 bloqué	Vérifiez la redirection sur votre box / configurez une DMZ
Les IPs publiques ne pingent pas	Route manquante ou NAT mal configuré	Vérifiez `ip route` ou `show ip route`
Latence élevée / Pertes de paquets	MTU trop grande	Réduisez la MTU : `ip link set mtu 1480 dev tunnel0`

---

8. Bonnes Pratiques

1. **Sécurité** :
   - Filtrez le trafic entrant avec un pare-feu (ex: `iptables` ou ACL Cisco).
   - IPIP n'offre aucun chiffrement. Envisagez IPsec si le chiffrement est nécessaire.
2. **Performance** :
   - Réduisez la MTU à `1480` pour éviter la fragmentation (overhead IPIP = 20 octets).
   - Activez le **keepalive** sur le tunnel (ex: `keepalive 10 3` sur Cisco).

3. **Redondance** :
    - Configurez un deuxième tunnel IPIP pour le failover.

Revision #1
Created 2026-03-30 15:02:44 UTC by Landry JUGE
Updated 2026-03-30 15:03:36 UTC by Landry JUGE