Tunnel L2 - L2TPv3

1. Introduction : Pourquoi utiliser L2TPv3 ?

1.1. À quoi sert un tunnel L2TPv3 ?

Un tunnel **L2TPv3 (Layer 2 Tunneling Protocol version 3)** permet de créer un lien Ethernet virtuel (Layer 2) entre deux sites distants, transportant des **trames Ethernet complètes**. L2TPv3 est une évolution de L2TP, spécifiquement conçue pour le transport L2 avec de nombreuses options de configuration (encapsulation UDP ou IP, IDs de session/tunnel configurables). **Exemple concret :**

- Vous avez un réseau local derrière une box opérateur standard.
- Vous voulez une ou des IPs publiques directement accessibles sur vos machines, comme si elles étaient sur le même réseau que la plateforme.

1.2. Avantages de L2TPv3

- Transport Layer 2 complet (trames Ethernet).
- Supporte l'encapsulation **UDP** (facilement NAT-able) ou **IP** (protocole 115, plus léger).
- IDs de session et tunnel configurables, permettant plusieurs tunnels vers le même endpoint.
- Compatible avec de nombreux équipements réseau (Linux, MikroTik, Cisco, Juniper).
- Peut être combiné avec d'autres tunnels L2 via un bridge.

1.3. Inconvénients

- Pas de chiffrement natif.
- Configuration plus complexe que VXLAN ou GRETAP (IDs de session/tunnel à configurer).
- En mode IP (protocole 115), nécessite une DMZ si derrière un NAT.

---

2. Prérequis

2.1. Ce dont vous avez besoin

- Une **IP publique** (ou une redirection de port si encapsulation UDP, ou une DMZ si encapsulation IP).
- Un **routeur ou serveur compatible L2TPv3** (Linux, MikroTik, Cisco, Juniper, etc.).
- Un **service Tunnel-IP**.

2.2. Ports et NAT

- **Encapsulation UDP** : L2TPv3 utilise un **port UDP configurable**. Si vous êtes derrière un NAT, il suffit de rediriger ce port.
- **Encapsulation IP** : L2TPv3 utilise le **protocole 115**. Ce n'est ni UDP ni TCP, il faut donc configurer une DMZ si vous êtes derrière un NAT.

---

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

> Les tunnels L2 fonctionnent avec des **bridges** sur la plateforme. Un bridge est un switch virtuel qui regroupe vos tunnels L2 et vos subnets. Vous devez d'abord créer un bridge, puis créer le tunnel en l'associant à ce bridge.

3.1. Créer le bridge

1. Connectez-vous à panel.tunnel-ip.com
2. Allez dans **"Bridges"**.
3. Cliquez sur **"Ajouter un Bridge"** et donnez-lui un nom.
4. Validez. Le bridge sera créé sur la plateforme.

3.2. Créer le tunnel L2TPv3

1. Allez dans **"Tunnels"** > **"L2TPv3"**.

2. Remplissez les informations :
   - **Endpoint** (IP publique du routeur / box, ex : `<span class="editor-theme-code">203.0.113.1</span>`).
   - **Zone** : Zone géographique du serveur.
   - **Nom** (ex : `<span class="editor-theme-code">L2TPv3_Tunnel</span>`).
   - **MTU** : Taille maximale des paquets (-1 pour laisser la plateforme choisir).
   - **Bridge** : Sélectionnez le bridge créé précédemment.
   - **Encapsulation** : Choisissez **UDP** ou **IP** selon votre configuration réseau.
   - **Peer session id** : ID de session du côté distant (votre côté). Doit correspondre au `session_id` que vous configurerez sur votre routeur.
   - **Peer tunnel id** : ID de tunnel du côté distant (votre côté). Doit correspondre au `tunnel_id` que vous configurerez sur votre routeur.
   - **Peer udp port** : Port UDP de l'autre côté du tunnel. Laisser vide si encapsulation IP.
3. Validez. La plateforme s'occupera de configurer le tunnel côté Tunnel-IP.com.
4. Attendez que le tunnel soit créé
5. Cliquez sur "Accéder" pour récupérer les détails du tunnel (vous y trouverez les IDs côté plateforme)

> **Important :** Les IDs de session et de tunnel doivent être **croisés** : le `peer_session_id` de la plateforme doit correspondre à votre `session_id` local, et inversement. Vérifiez bien les valeurs sur la page détails du tunnel.

3.3. Créer le subnet

1. Allez dans **"Subnets"**

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

3. Sélectionnez le **bridge** (et non un tunnel directement) comme destination

4. Cliquez sur Créer

5. Une fois son statut à "Actif", la route est correctement installée sur les routeurs de la plateforme

> **Important :** La première IP du subnet sera utilisée comme **gateway** sur le bridge côté plateforme. Par exemple, pour le subnet `198.51.100.0/29`, la gateway sera `198.51.100.1`.

---

4. Étape 2 : Configurer le tunnel

> L2TPv3 transporte des trames Ethernet (Layer 2). Côté client, vous devez créer l'interface L2TPv3 puis la **bridger** avec votre interface LAN (ou lui assigner directement une IP du subnet).

> Dans les exemples ci-dessous, les valeurs suivantes sont utilisées. **Adaptez-les** avec les paramètres fournis par la plateforme :
> - IP distante (plateforme) : `172.16.126.33`
> - Session ID local : `1000` / Peer session ID : `2000`
> - Tunnel ID local : `1000` / Peer tunnel ID : `2000`
> - Port UDP (si encapsulation UDP) : `1701`

4.1. Exemple pour Linux (Ubuntu/Debian) — Encapsulation UDP

Étapes :

1. **Créer le tunnel L2TPv3** :

```
ip l2tp add tunnel tunnel_id 1000 peer_tunnel_id 2000 \
  encap udp local 192.168.1.1 remote 172.16.126.33 \
  udp_sport 1701 udp_dport 1701
```

   - `tunnel_id 1000` : ID du tunnel local.
   - `peer_tunnel_id 2000` : ID du tunnel côté plateforme.
   - `local 192.168.1.1` : IP locale du routeur.
   - `remote 172.16.126.33` : IP distante (Tunnel-IP.com).
   - `udp_sport` / `udp_dport` : Ports UDP source et destination.

2. **Créer la session L2TPv3** :

```
ip l2tp add session tunnel_id 1000 session_id 1000 \
  peer_session_id 2000
```

3. **Activer l'interface** :

   ```
   ip link set l2tpeth0 up
   ```

   L'interface `l2tpeth0` est créée automatiquement.
4. **Option A : Assigner directement une IP publique** :

   ```
   ip addr add 198.51.100.2/29 dev l2tpeth0
   ip route add default via 198.51.100.1 dev l2tpeth0
   ```

5. **Option B : Bridger avec une interface LAN** :

   ```
   ip link add br0 type bridge
   ip link set br0 up
   ip link set l2tpeth0 master br0
   ip link set eth1 master br0  # Interface LAN vers vos machines
   ```

   Les machines connectées à `eth1` pourront alors utiliser les IPs du subnet directement, avec `198.51.100.1` comme gateway.

4.2. Exemple pour Linux — Encapsulation IP

La procédure est similaire, mais avec `encap ip` :

```
ip l2tp add tunnel tunnel_id 1000 peer_tunnel_id 2000 \
  encap ip local 192.168.1.1 remote 172.16.126.33

ip l2tp add session tunnel_id 1000 session_id 1000 \
  peer_session_id 2000

ip link set l2tpeth0 up
```

> En encapsulation IP, aucun port UDP n'est utilisé. Le protocole 115 est utilisé directement. Assurez-vous que votre box / NAT le laisse passer (DMZ recommandée).

4.3. Exemple pour MikroTik

Étapes :

1. **Créer l'interface L2TP client** :

> MikroTik supporte L2TPv3 à partir de **RouterOS v7**. Sur les versions antérieures, seul L2TP (v2) est disponible.

```
```bash
/interface/l2tp-ether/add name=l2tp-ether1 \
  local-address=192.168.1.1 remote-address=172.16.126.33 \
  tunnel-id=1000 peer-tunnel-id=2000 \
  session-id=1000 peer-session-id=2000
```
```

2. **Option A : Assigner une IP publique directement** :

```
/ip/address/add address=198.51.100.2/29 interface=l2tp-ether1
/ip/route/add dst-address=0.0.0.0/0 gateway=198.51.100.1
```

3. **Option B : Bridger avec une interface LAN** :

```
/interface/bridge/add name=br-l2tp
/interface/bridge/port/add bridge=br-l2tp interface=l2tp-ether1
/interface/bridge/port/add bridge=br-l2tp interface=ether2  # Interface LAN
```

4.4. Exemple pour Cisco (IOS/XE)

Étapes :

1. **Accéder au mode configuration** :

```
enable
configure terminal
```

2. **Créer la pseudowire L2TPv3** :

```
pseudowire-class L2TPv3-PW
 encapsulation l2tpv3
 protocol none
 ip local interface GigabitEthernet0/0
```

3. **Créer l'interface xconnect** :

```
interface GigabitEthernet0/1
 xconnect 172.16.126.33 1000 encapsulation l2tpv3 pw-class L2TPv3-PW
```

   - `172.16.126.33` : IP distante (Tunnel-IP.com).
   - `1000` : VC ID (doit correspondre aux IDs configurés sur la plateforme).
4. **Sauvegarder la configuration** :

```
write memory
```

> La configuration L2TPv3 sur Cisco varie selon la plateforme (IOS, IOS-XE, NX-OS). La syntaxe `xconnect` est disponible sur IOS et IOS-XE. Consultez la documentation de votre modèle.

4.5. Exemple pour Juniper (JunOS)

Étapes :

1. **Accéder au mode configuration** :

```
edit
```

2. **Créer le tunnel L2TPv3** :

```
set protocols l2circuit neighbor 172.16.126.33 interface ge-0/0/1.0 \
  virtual-circuit-id 1000 encapsulation-type ethernet

set interfaces ge-0/0/1 encapsulation ethernet-bridge
set interfaces ge-0/0/1 unit 0 family bridge
```

   - `172.16.126.33` : IP distante (Tunnel-IP.com).
   - `virtual-circuit-id 1000` : ID du circuit (doit correspondre aux IDs de la plateforme).
3. **Créer le bridge domain** :

```
set bridge-domains L2TPv3-BD interface ge-0/0/1.0
set bridge-domains L2TPv3-BD routing-interface irb.100

set interfaces irb unit 100 family inet address 198.51.100.2/29
```

4. **Appliquer la configuration** :

```
commit
```

4.6. Note pour Arista (EOS)

> Arista EOS ne supporte pas nativement L2TPv3. Si vous devez utiliser un switch Arista, vous pouvez utiliser VXLAN comme alternative L2, ou placer un serveur Linux comme point de terminaison L2TPv3 devant votre switch Arista.

---

5. Étape 3 : Configurer les IPs publiques

5.1. Architecture L2

```
Internet → [Infrastructure Tunnel-IP.com] → [Bridge plateforme] → (Tunnel L2TPv3) → [Votre Bridge/Interface] → [Vos Machines]
```

- La plateforme met la première IP du subnet comme gateway sur le bridge (ex: `198.51.100.1`).
- Vos machines utilisent les autres IPs du subnet.
- Comme le tunnel est L2, les trames Ethernet passent directement, **pas besoin de NAT ni de routage complexe**.

5.2. Attribution des IPs

Un `/29` contient **8 adresses IP** :

- `198.51.100.0` : Adresse réseau (inutilisable).
- `198.51.100.1` : **Gateway plateforme** (réservée automatiquement).
- `198.51.100.2` à `198.51.100.6` : IPs utilisables pour vos machines.
- `198.51.100.7` : Adresse broadcast (inutilisable).

5.3. Configuration sur les machines

Sur chaque machine qui doit utiliser une IP publique :

Linux :

```
ip addr add 198.51.100.2/29 dev eth0
ip route add default via 198.51.100.1
```

MikroTik :

```
/ip/address/add address=198.51.100.2/29 interface=ether1
/ip/route/add dst-address=0.0.0.0/0 gateway=198.51.100.1
```

> Si vous avez bridgé l'interface L2TPv3 avec votre LAN, les machines peuvent être configurées directement avec les IPs publiques et la gateway `198.51.100.1`, comme si elles étaient sur un réseau local classique.

---

6. Commandes spécifiques par constructeur

Constructeur	Commande pour vérifier le tunnel	Commande pour vérifier le bridge
Linux	`ip l2tp show tunnel` / `ip l2tp show session`	`bridge link show`
MikroTik	`/interface l2tp-ether print`	`/interface bridge port print`
Cisco	`show l2tun tunnel` / `show xconnect all`	`show bridge-domain`
Juniper	`show l2circuit connections`	`show bridge-domain`

7. Vérification et Dépannage

7.1. Tester la connectivité

- **Depuis votre routeur / machine** :

```
ping 198.51.100.1  # Ping de la gateway plateforme
```

- **Vérifier le tunnel L2TPv3 (Linux)** :

```
ip l2tp show tunnel  # Vérifier les tunnels
ip l2tp show session  # Vérifier les sessions
ip -d link show l2tpeth0  # Vérifier l'interface
```

- **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 (encap UDP)	Port UDP bloqué	Vérifiez la redirection du port sur votre box
Le tunnel ne répond pas (encap IP)	Protocole 115 bloqué	Configurez une DMZ sur votre box
Pas de connectivité L2	IDs session/tunnel inversés	Vérifiez que les IDs sont bien croisés : votre `session_id` = `peer_session_id` de la plateforme, et inversement
La gateway ne répond pas	Subnet non associé au bridge	Vérifiez que le subnet est bien associé au bridge sur la plateforme
Latence élevée / Pertes de paquets	MTU trop grande	Réduisez la MTU (overhead L2TPv3 UDP ≈ 36 octets, IP ≈ 12 octets)
Les IPs publiques ne fonctionnent pas	Mauvaise gateway	Utilisez la première IP du subnet comme gateway (ex: `198.51.100.1`)

---

8. Bonnes Pratiques

1. **Sécurité** :
   - Filtrez le trafic entrant avec un pare-feu (ex: `iptables` ou ACL).
   - L2TPv3 n'offre aucun chiffrement natif. Envisagez IPsec en complément si le chiffrement est nécessaire.
2. **Performance** :
   - Ajustez la MTU en fonction de l'encapsulation choisie.
   - L'encapsulation **IP** est plus performante (moins d'overhead) mais moins compatible avec le NAT.
   - L'encapsulation **UDP** est recommandée si vous êtes derrière un NAT.
3. **Bridging** :

- Vous pouvez combiner plusieurs tunnels L2 (VXLAN, EoIP, GRETAP, L2TPv3) sur le même bridge, aussi bien sur la plateforme que côté client.
- Attention aux boucles de bridge : activez le STP si nécessaire.

Revision #1
Created 2026-03-30 14:42:32 UTC by Landry JUGE
Updated 2026-03-30 14:42:59 UTC by Landry JUGE