Guide des Opérations OmniUPF

Table des Matières

1. Aperçu
2. Comprendre l'Architecture du Plan Utilisateur 5G
3. Composants UPF
4. Protocole PFCP et Intégration SMF
5. Opérations Courantes
6. Dépannage
7. Documentation Supplémentaire
8. Glossaire

Aperçu

OmniUPF (Fonction de Plan Utilisateur basée sur eBPF) est une fonction de plan utilisateur 5G/LTE haute performance qui fournit un transfert de paquets de qualité opérateur, l'application de la QoS et la gestion du trafic pour les réseaux mobiles. Basé sur la technologie eBPF de Linux (Extended Berkeley Packet Filter) et amélioré avec des capacités de gestion complètes, OmniUPF fournit l'infrastructure de traitement de paquets essentielle requise pour les réseaux 5G SA, 5G NSA et LTE.

Qu'est-ce qu'une Fonction de Plan Utilisateur ?

La Fonction de Plan Utilisateur (UPF) est l'élément de réseau standardisé par la 3GPP responsable du traitement et du transfert des paquets dans les réseaux 5G et LTE. Elle fournit :

- **Transfert de paquets à haute vitesse** entre les appareils mobiles et les réseaux de données
- **Application de la Qualité de Service (QoS)** pour différents types de trafic
- **Détection et routage du trafic** basé sur des filtres de paquets et des règles
- **Rapport d'utilisation** pour la facturation et l'analyse
- **Mise en mémoire tampon des paquets** pour les scénarios de gestion de mobilité et de session
- **Support d'interception légale** pour la conformité réglementaire

OmniUPF implémente l'intégralité de la fonctionnalité UPF définie dans la 3GPP TS 23.501 (5G) et TS 23.401 (LTE), fournissant une solution de plan utilisateur complète et prête pour la production utilisant la technologie eBPF du noyau Linux pour des performances maximales.

Capacités Clés d'OmniUPF

Traitement des Paquets :

- Traitement des paquets de plan utilisateur conforme à la 3GPP
- Chemin de données basé sur eBPF pour des performances au niveau du noyau
- Encapsulation et décapsulation GTP-U (GPRS Tunneling Protocol)
- Support IPv4 et IPv6 pour les réseaux d'accès et de données
- XDP (eXpress Data Path) pour un traitement à latence ultra-faible
- Traitement des paquets multi-threadé

QoS et Gestion du Trafic :

- Règles d'Application de la QoS (QER) pour la gestion de la bande passante
- Règles de Détection de Paquets (PDR) pour la classification du trafic
- Règles d'Action de Transfert (FAR) pour les décisions de routage
- Filtrage de Flux de Données de Service (SDF) pour le routage spécifique aux applications

- Règles de Rapport d'Utilisation (URR) pour le suivi des volumes et la facturation

Contrôle et Gestion :

- Interface PFCP (Packet Forwarding Control Protocol) vers SMF/PGW-C
- API RESTful pour la surveillance et le diagnostic
- Statistiques et métriques en temps réel
- Surveillance de la capacité des cartes eBPF
- Panneau de contrôle basé sur le web

Fonctionnalités de Performance :

- Traitement des paquets sans copie via eBPF
- Transfert de paquets au niveau du noyau (sans surcharge de l'espace utilisateur)
- Scalabilité multi-cœurs
- Capable de décharger pour l'accélération matérielle
- Optimisé pour les déploiements cloud-natifs

Pour des détails sur l'utilisation du panneau de contrôle, voir Opérations de l'Interface Web.

Comprendre l'Architecture du Plan Utilisateur

OmniUPF est une solution unifiée de plan utilisateur fournissant un transfert de paquets de qualité opérateur pour les réseaux 5G Standalone (SA), 5G NSA et 4G LTE/EPC. **OmniUPF est un produit unique** qui peut simultanément fonctionner comme :

- **UPF (Fonction de Plan Utilisateur)** - plan utilisateur 5G/NSA (contrôlé par OmniSMF via N4/PFCP)
- **PGW-U (Passerelle de Données Utilisateur)** - passerelle EPC 4G vers des réseaux externes (contrôlé par OmniPGW-C via Sxc/PFCP)

- **SGW-U (Passerelle de Service Utilisateur)** - passerelle de service EPC 4G (contrôlé par OmniSGW-C via Sxb/PFCP)

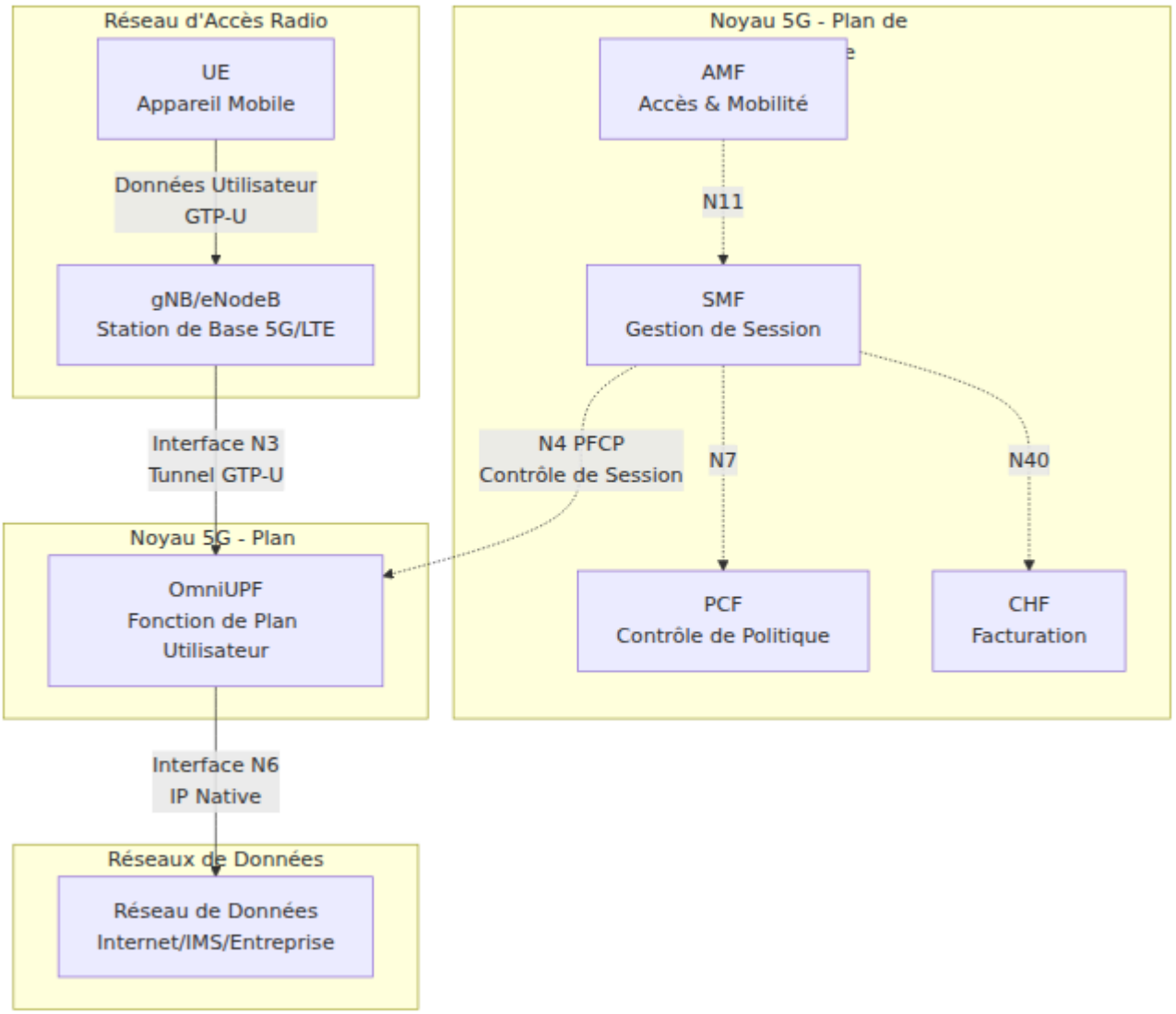
OmniUPF peut fonctionner dans **n'importe quelle combinaison** de ces modes :

- **UPF uniquement** : Déploiement 5G pur
- **PGW-U + SGW-U** : Passerelle 4G combinée (déploiement EPC typique)
- **UPF + PGW-U + SGW-U** : Support simultané 4G et 5G (scénario de migration)

Tous les modes utilisent le même moteur de traitement de paquets basé sur eBPF et le protocole PFCP, offrant des performances élevées constantes qu'il fonctionne comme UPF, PGW-U, SGW-U, ou les trois simultanément.

Architecture du Réseau 5G (Mode SA)

La solution OmniUPF se trouve dans le plan de données des réseaux 5G, fournissant la couche de transfert de paquets à haute vitesse qui connecte les appareils mobiles aux réseaux et services de données.



Architecture du Réseau 4G LTE/EPC

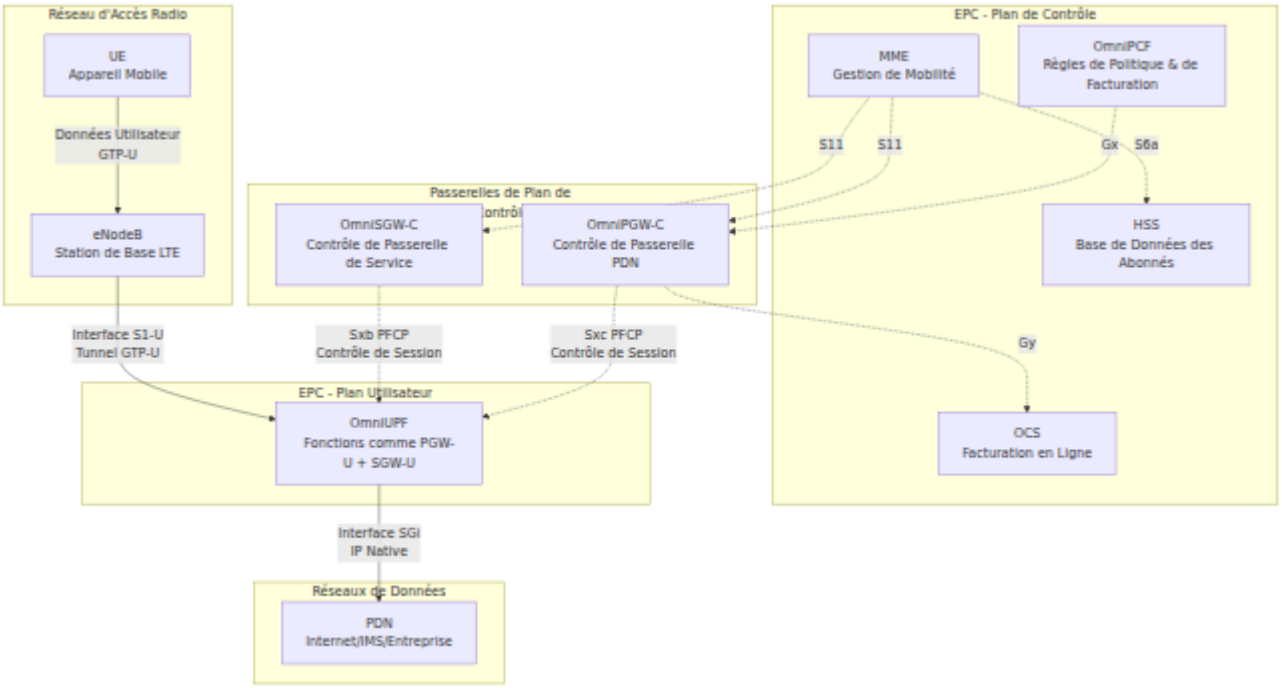
OmniUPF prend également en charge les déploiements 4G LTE et EPC (Evolved Packet Core), fonctionnant soit comme OmniPGW-U soit comme OmniSGW-U selon l'architecture du réseau.

Mode Combiné PGW-U/SGW-U (Déploiement 4G Typique)

Dans ce mode, OmniUPF agit à la fois comme SGW-U et PGW-U, contrôlé par des fonctions de plan de contrôle séparées.

Mode Séparé SGW-U et PGW-U (Roaming/Multi-Site)

Dans les déploiements de roaming ou multi-site, deux instances séparées d'OmniUPF peuvent être déployées - une comme SGW-U et une comme PGW-U.



Comment Fonctionnent les Fonctions de Plan Utilisateur dans le Réseau

La fonction de plan utilisateur (OmniUPF, OmniPGW-U ou OmniSGW-U) fonctionne comme le plan de transfert contrôlé par le plan de contrôle respectif :

1. **Établissement de Session**
   - **5G** : OmniSMF établit une association PFCP via l'interface N4 avec OmniUPF
   - **4G** : OmniPGW-C ou OmniSGW-C établit une association PFCP via Sxb/Sxc avec OmniPGW-U/OmniSGW-U
   - Le plan de contrôle crée des sessions PFCP pour chaque session PDU UE (5G) ou contexte PDP (4G)
   - Le plan utilisateur reçoit des règles PDR, FAR, QER et URR via PFCP
   - Les cartes eBPF sont peuplées avec des règles de transfert
2. **Traitement des Paquets Montants** (UE → Réseau de Données)
   - **5G** : Les paquets arrivent sur l'interface N3 depuis gNB avec encapsulation GTP-U
   - **4G** : Les paquets arrivent sur l'interface S1-U (SGW-U) ou l'interface S5/S8 (PGW-U) depuis eNodeB avec encapsulation GTP-U
   - Le plan utilisateur fait correspondre les paquets avec les PDR montants basés sur le TEID
   - Le programme eBPF applique QER (limitation de débit, marquage)
   - FAR détermine l'action de transfert (transférer, supprimer, mettre en mémoire tampon, dupliquer)
   - Tunnel GTP-U supprimé, paquets transférés vers l'interface N6 (5G) ou SGi (4G)
   - URR suit les comptes de paquets et d'octets pour la facturation
3. **Traitement des Paquets Descendants** (Réseau de Données → UE)
   - **5G** : Les paquets arrivent sur l'interface N6 comme IP native
   - **4G** : Les paquets arrivent sur l'interface SGi comme IP native

- Le plan utilisateur fait correspondre les paquets avec les PDR descendants basés sur l'adresse IP de l'UE
- Les filtres SDF peuvent classifier davantage le trafic par port, protocole ou application
- FAR détermine le tunnel GTP-U et les paramètres de transfert
- Encapsulation GTP-U ajoutée avec le TEID approprié
- **5G** : Paquets transférés vers l'interface N3 vers gNB
- **4G** : Paquets transférés vers S1-U (SGW-U) ou S5/S8 (PGW-U) vers eNodeB

4. **Mobilité et Transfert**

   - **5G** : OmniSMF met à jour les règles PDR/FAR pendant les scénarios de transfert
   - **4G** : OmniSGW-C/OmniPGW-C met à jour les règles pendant le transfert inter-eNodeB ou TAU (Mise à Jour de Zone de Suivi)
   - Le plan utilisateur peut mettre en mémoire tampon les paquets pendant le changement de chemin
   - Transition transparente entre les stations de base sans perte de paquets

Intégration avec le Plan de Contrôle (4G et 5G)

OmniUPF s'intègre avec les fonctions de plan de contrôle 5G et 4G via des interfaces standard 3GPP :

Interfaces 5G

Interface	De → À	But	Spécification 3GPP
N4	OmniSMF ↔ OmniUPF	Établissement, modification, suppression de session PFCP	TS 29.244
N3	gNB → OmniUPF	Trafic de plan utilisateur depuis RAN (GTP-U)	TS 29.281
N6	OmniUPF → Réseau de Données	Trafic de plan utilisateur vers DN (IP native)	TS 23.501
N9	OmniUPF ↔ OmniUPF	Communication inter-UPF pour roaming/edge	TS 23.501

Interfaces 4G/EPC

Interface	De → À	But	Spécification 3GPP
Sxb	OmniSGW-C ↔ OmniUPF (mode SGW-U)	Contrôle de session PFCP pour la passerelle de service	TS 29.244
Sxc	OmniPGW-C ↔ OmniUPF (mode PGW-U)	Contrôle de session PFCP pour la passerelle PDN	TS 29.244
S1-U	eNodeB → OmniUPF (mode SGW-U)	Trafic de plan utilisateur depuis RAN (GTP-U)	TS 29.281
S5/S8	OmniUPF (SGW-U) ↔ OmniUPF (PGW-U)	Plan utilisateur inter-passerelle (GTP-U)	TS 29.281
SGi	OmniUPF (mode PGW-U) → PDN	Trafic de plan utilisateur vers le réseau de données (IP native)	TS 23.401

Remarque : Toutes les interfaces PFCP (N4, Sxb, Sxc) utilisent le même protocole PFCP défini dans la TS 29.244. Les noms des interfaces diffèrent mais le protocole et les formats de message sont identiques.

Composants UPF

Chemin de Données eBPF

Le **chemin de données eBPF** est le moteur de traitement de paquets central qui fonctionne dans le noyau Linux pour des performances maximales.

Fonctions Principales :

- **Traitement GTP-U** : Encapsulation et décapsulation des tunnels GTP-U
- **Classification des Paquets** : Correspondance des paquets avec les règles PDR en utilisant TEID, IP de l'UE ou filtres SDF
- **Application de la QoS** : Appliquer la limitation de débit et le marquage des paquets par les règles QER
- **Décisions de Transfert** : Exécuter les actions FAR (transférer, supprimer, mettre en mémoire tampon, dupliquer, notifier)
- **Suivi de l'Utilisation** : Incrémenter les compteurs URR pour la facturation basée sur le volume

Cartes eBPF : Le chemin de données utilise des cartes eBPF (tables de hachage en mémoire du noyau) pour le stockage des règles :

Nom de la Carte	But	Clé	Valeur
`uplink_pdr_map`	PDRs Montants	TEID (32 bits)	Infos PDR (ID FAR, ID QER, IDs URR)
`downlink_pdr_map`	PDRs Descendants (IPv4)	Adresse IP de l'UE	Infos PDR
`downlink_pdr_map_ip6`	PDRs Descendants (IPv6)	Adresse IPv6 de l'UE	Infos PDR
`far_map`	Règles de Transfert	ID FAR	Paramètres de transfert (action, infos de tunnel)
`qer_map`	Règles de QoS	ID QER	Paramètres de QoS (MBR, GBR, marquage)
`urr_map`	Suivi de l'Utilisation	ID URR	Compteurs de volume (montant, descendant, total)
`sdf_filter_map`	Filtres SDF	ID PDR	Filtres d'application (ports, protocoles)

Caractéristiques de Performance :

- **Sans copie** : Paquets traités entièrement dans l'espace noyau
- **Support XDP** : Attacher au niveau du pilote réseau pour une latence sub-microseconde
- **Multi-cœurs** : Évolue à travers les cœurs CPU avec support de carte par CPU

- **Capacité** : Millions de PDRs/FARs dans les cartes eBPF (limité par la mémoire du noyau)

Pour la surveillance de la capacité, voir Gestion de la Capacité.

Gestionnaire d'Interface PFCP

L'**interface PFCP** implémente la 3GPP TS 29.244 pour la communication avec SMF ou PGW-C.

Fonctions Principales :

- **Gestion des Associations** : Cœur de PFCP et mise en place/libération d'association
- **Cycle de Vie de Session** : Créer, modifier et supprimer des sessions PFCP
- **Installation de Règles** : Traduire les IEs PFCP en entrées de carte eBPF
- **Rapport d'Événements** : Notifier SMF des seuils d'utilisation, erreurs ou événements de session

Support des Messages PFCP :

Type de Message	Direction	But
Établissement d'Association	SMF → UPF	Établir l'association de contrôle PFCP
Libération d'Association	SMF → UPF	Détruire l'association PFCP
Cœur de PFCP	Bidirectionnel	Maintenir l'association active
Établissement de Session	SMF → UPF	Créer une nouvelle session PDU avec PDR/FAR/QER/URR
Modification de Session	SMF → UPF	Mettre à jour les règles pour la mobilité, les changements de QoS
Suppression de Session	SMF → UPF	Supprimer la session et toutes les règles associées
Rapport de Session	UPF → SMF	Rapporter l'utilisation, les erreurs ou les événements

Éléments d'Information (IE) Supportés :

- Créer PDR, FAR, QER, URR
- Mettre à jour PDR, FAR, QER, URR
- Supprimer PDR, FAR, QER, URR
- Informations de Détection de Paquets (IP de l'UE, F-TEID, filtre SDF)
- Paramètres de Transfert (instance réseau, création d'en-tête externe)
- Paramètres de QoS (MBR, GBR, QFI)
- Déclencheurs de Rapport d'Utilisation (seuil de volume, seuil de temps)

Serveur API REST

L'**API REST** fournit un accès programmatique à l'état et aux opérations de l'UPF.

Fonctions Principales :

- **Surveillance des Sessions** : Interroger les sessions PFCP actives et les associations
- **Inspection des Règles** : Voir les configurations PDR, FAR, QER, URR
- **Statistiques** : Récupérer les compteurs de paquets, les statistiques de routage, les statistiques XDP
- **Gestion des Tampons** : Voir et contrôler les tampons de paquets
- **Informations sur les Cartes** : Surveiller l'utilisation et la capacité des cartes eBPF

Points de Terminaison API : (34 points de terminaison au total)

Catégorie	Points de Terminaison	Description
Santé	`/health`	Vérification de la santé et statut
Configuration	`/config`	Configuration de l'UPF
Sessions	`/pfcp_sessions`, `/pfcp_associations`	Données de session/association PFCP
PDRs	`/uplink_pdr_map`, `/downlink_pdr_map`, `/downlink_pdr_map_ip6`, `/uplink_pdr_map_ip6`	Règles de détection de paquets
FARs	`/far_map`	Règles d'action de transfert
QERs	`/qer_map`	Règles d'application de la QoS
URRs	`/urr_map`	Règles de rapport d'utilisation
Tampons	`/buffer`	État et contrôle des tampons de paquets
Statistiques	`/packet_stats`, `/route_stats`, `/xdp_stats`, `/n3n6_stats`	Métriques de performance
Capacité	`/map_info`	Capacité et utilisation de la carte eBPF
Plan de Données	`/dataplane_config`	Adresses des interfaces N3/N9

Pour des détails sur l'API et son utilisation, voir **Guide de Surveillance**.

Panneau de Contrôle Web

Le **Panneau de Contrôle Web** fournit un tableau de bord en temps réel pour la surveillance et la gestion de l'UPF.

Fonctionnalités :

- **Vue des Sessions** : Parcourir les sessions PFCP actives avec IP de l'UE, TEID et comptes de règles
- **Gestion des Règles** : Voir et gérer les PDRs, FARs, QERs et URRs à travers toutes les sessions
- **Surveillance des Tampons** : Suivre les paquets mis en mémoire tampon et contrôler la mise en mémoire tampon par FAR
- **Tableau de Bord Statistiques** : Statistiques en temps réel des paquets, des routes, XDP et des interfaces N3/N6
- **Surveillance de la Capacité** : Utilisation de la carte eBPF avec des indicateurs de capacité codés par couleur
- **Vue de Configuration** : Afficher la configuration de l'UPF et les adresses du plan de données
- **Visionneuse de Journaux** : Diffusion en direct des journaux pour le dépannage

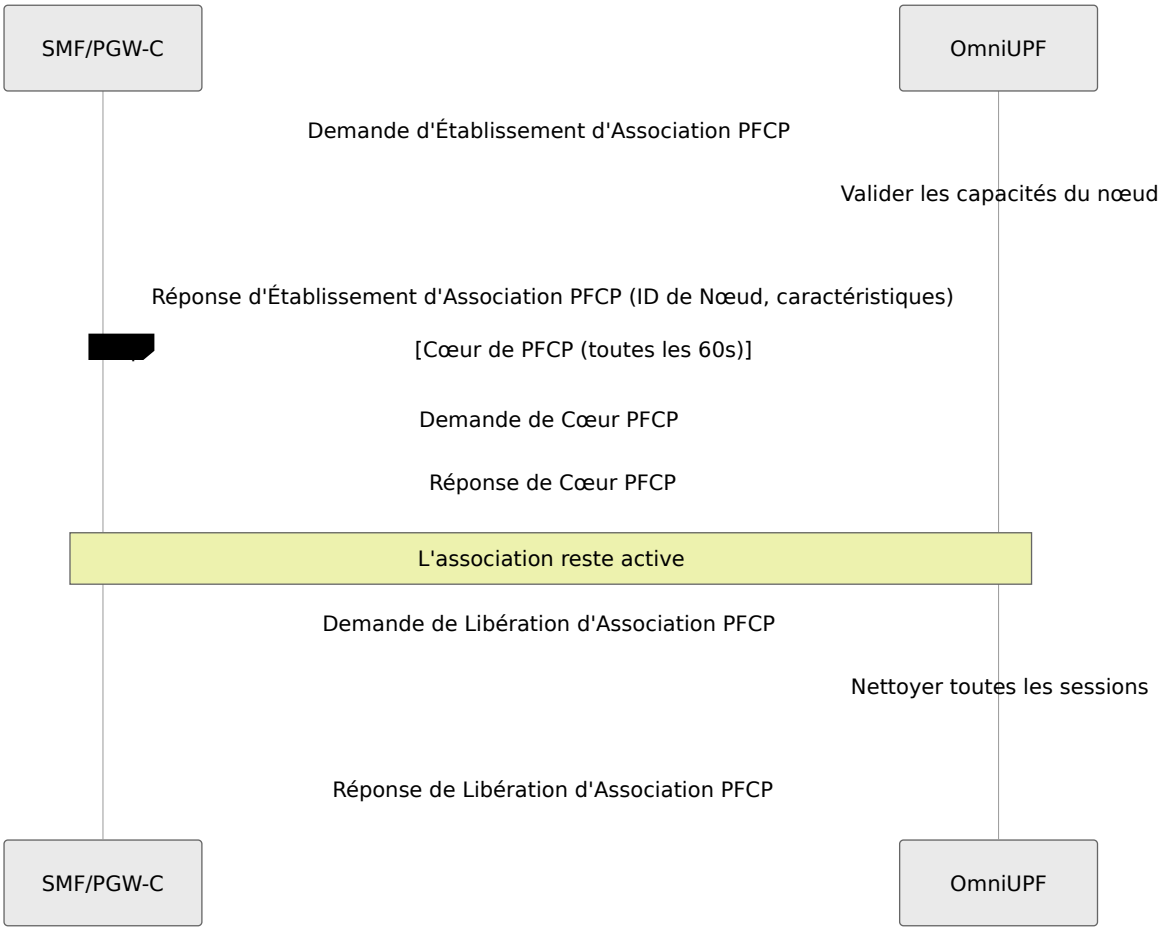
Pour des opérations détaillées de l'UI, voir **Guide des Opérations de l'Interface Web**.

Protocole PFCP et Intégration SMF

Association PFCP

Avant que des sessions puissent être créées, le SMF doit établir une association PFCP avec l'UPF.

Cycle de Vie de l'Association :



Points Clés :

- Chaque SMF établit une association avec l'UPF
- L'UPF suit l'association par ID de Nœud (FQDN ou adresse IP)
- Les messages de cœur maintiennent la vivacité de l'association
- Toutes les sessions sous une association sont supprimées si l'association est libérée

Pour voir les associations, voir Vue des Sessions.

Détection de Redémarrage SMF et Nettoyage des Sessions Orphelines

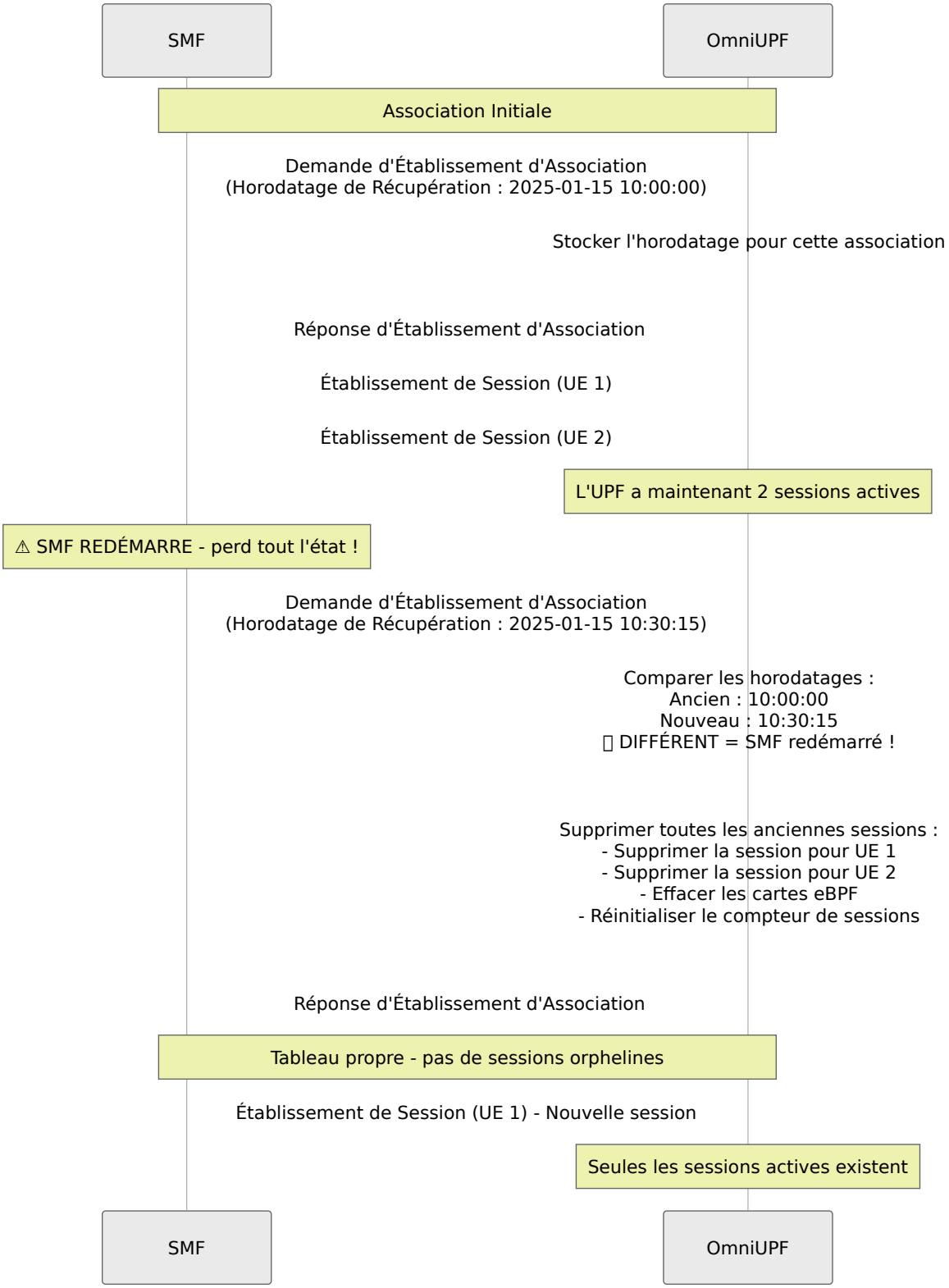
OmniUPF détecte automatiquement quand un SMF redémarre et nettoie les sessions orphelines selon les spécifications de la 3GPP TS 29.244.

Comment Cela Fonctionne :

Lorsqu'un SMF établit une association PFCP, il fournit un **Horodatage de Récupération** indiquant quand il a démarré. OmniUPF stocke cet horodatage pour chaque association. Si le SMF redémarre :

1. Le SMF perd tout l'état de session en mémoire
2. Le SMF rétablit l'association PFCP avec l'UPF
3. Le SMF envoie **un nouvel Horodatage de Récupération** (différent de l'avant)
4. L'UPF détecte le changement d'horodatage = SMF redémarré
5. L'UPF supprime automatiquement **toutes les sessions orphelines** de l'ancienne instance SMF
6. Le SMF crée de nouvelles sessions pour les abonnés actifs

Flux de Détection de Redémarrage :



Exemple de Journal :

Lorsqu'un SMF redémarre, vous verrez :

```
WARN: L'association avec NodeID: smf-1 et adresse: 192.168.1.10
existe déjà
WARN: L'Horodatage de Récupération du SMF a changé (ancien : 2025-
01-15T10:00:00Z, nouveau : 2025-01-15T10:30:15Z) - SMF redémarré,
suppression de 245 sessions orphelines
INFO: Suppression de la session orpheline 2 (LocalSEID) en raison
du redémarrage du SMF
INFO: Suppression de la session orpheline 3 (LocalSEID) en raison
du redémarrage du SMF
...
INFO: Suppression de la session orpheline 246 (LocalSEID) en
raison du redémarrage du SMF
```

Remarques Importantes :

1. **Isolation** : Seules les sessions du SMF redémarré sont supprimées. Les autres associations de SMF et leurs sessions ne sont **pas affectées**.

2. **Comparaison des Horodatages** : Si l'Horodatage de Récupération est **identique**, les sessions sont **conservées** (SMF reconnecté sans redémarrage).

3. **Conformité 3GPP** : Ce comportement est imposé par la section 5.22.2 de la norme 3GPP TS 29.244 :

   > "Si l'Horodatage de Récupération de la fonction CP a changé depuis la dernière configuration d'association, la fonction UP doit considérer que la fonction CP a redémarré et doit supprimer toutes les sessions PFCP associées à cette fonction CP."

Pour le dépannage des sessions orphelines, consultez le Guide de Dépannage.

Gestion des Indications d'Erreur GTP-U

OmniUPF gère les messages d'Indication d'Erreur GTP-U des pairs en aval (PGW-U, SGW-U, eNodeB, gNodeB) conformément aux spécifications de la norme 3GPP TS 29.281.

Qu'est-ce que les Indications d'Erreur :

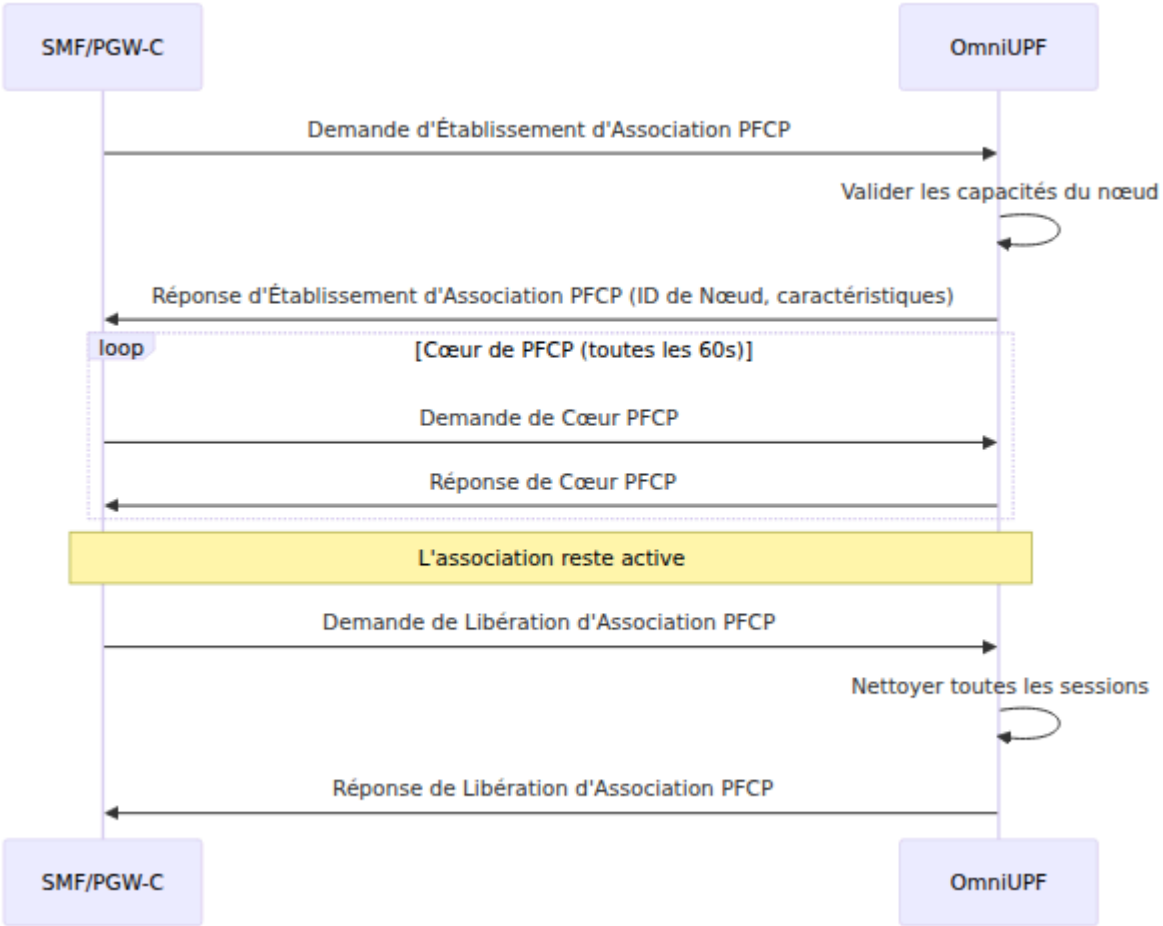
Lorsque OmniUPF transmet un paquet GTP-U à un pair distant (par exemple, PGW-U dans un déploiement SGW-U), le pair peut renvoyer une **Indication d'Erreur** s'il ne reconnaît pas le TEID (Identifiant de Point de Terminaison de Tunnel). Cela indique :

- Le pair distant a redémarré et a perdu l'état du tunnel
- Le tunnel n'a jamais été créé du côté distant (incompatibilité de configuration)
- Le tunnel a déjà été supprimé du côté distant

Comment ça marche :

1. **UPF transmet le paquet** → Envoie le paquet GTP-U avec TEID X au pair distant (port 2152)
2. **Le pair distant ne reconnaît pas le TEID X** → Recherche le TEID dans sa table de tunnels, non trouvé
3. **Le pair distant envoie une Indication d'Erreur** → Message GTP-U de type 26 avec IE contenant le TEID erroné
4. **UPF reçoit l'Indication d'Erreur** → Analyse le message pour extraire le TEID X
5. **UPF trouve les sessions affectées** → Recherche toutes les sessions pour les FARs transférant vers le TEID X
6. **UPF supprime les sessions** → Supprime les sessions des cartes eBPF et de l'état PFCP
7. **UPF met à jour les métriques** → Incrémente les compteurs Prometheus pour le suivi

Flux d'Indication d'Erreur :



Format du Paquet (section 7.3.1 de la norme 3GPP TS 29.281) :

```
Indication d'Erreur GTP-U :
┌──────────────────────────────────────────┐
│ En-tête GTP-U (12 octets)                │
├──────────────────────────────────────────┤
│ Version, PT, Drapeaux      │ 0x32        │
│ Type de Message            │ 26 (0x1A)   │
│ Longueur                   │ 9 octets     │
│ TEID                       │ 0 (toujours) │
│ Numéro de Séquence         │ varie        │
│ Numéro N-PDU              │ 0            │
│ Prochain En-tête d'Extension│ 0           │
├──────────────────────────────────────────┤
│ IE : Données TEID I (5 octets)           │
├──────────────────────────────────────────┤
│ Type                      │ 16 (0x10)    │
│ TEID erroné              │ 4 octets     │
└──────────────────────────────────────────┘
```

Quand cela a de l'importance :

Scénario 1 : Redémarrage de PGW-U dans l'architecture GTP S5/S8

- SGW-U (OmniUPF) transfère le trafic S5/S8 vers PGW-U
- PGW-U redémarre et perd tout l'état du tunnel S5/S8
- SGW-U continue de transférer vers les anciens TEIDs
- PGW-U envoie des Indications d'Erreur
- SGW-U **arrête automatiquement d'utiliser les tunnels morts**

Scénario 2 : Redémarrage du pair UPF dans l'architecture N9

- UPF-1 (OmniUPF) transfère le trafic N9 vers UPF-2
- UPF-2 redémarre
- UPF-1 reçoit des Indications d'Erreur
- UPF-1 nettoie les sessions

Exemple de Journal :

Lors de la réception d'une Indication d'Erreur :

```
WARN: Received GTP-U Error Indication from 192.168.50.10:2152 for
TEID 0x12345678 - remote peer doesn't recognize this TEID
WARN: Found session LocalSEID=42 with FAR GlobalId=1 forwarding to
erroneous TEID 0x12345678 from peer 192.168.50.10
INFO: Deleting session LocalSEID=42 due to GTP-U Error Indication
for TEID 0x12345678 from 192.168.50.10
WARN: Deleted 1 session(s) due to GTP-U Error Indication for TEID
0x12345678 from peer 192.168.50.10
```

Métriques Prometheus :

Surveillez l'activité des Indications d'Erreur avec une granularité par pair et par nœud :

```
# Total des Indications d'Erreur reçues des pairs
upf_buffer_listener_error_indications_received_total{node_id="pgw-u-
1",peer_address="192.168.50.10"}

# Sessions supprimées en raison des Indications d'Erreur
upf_buffer_listener_error_indication_sessions_deleted_total{node_id="
u-1",peer_address="192.168.50.10"}

# Indications d'Erreur envoyées (pour des TEIDs entrants inconnus)
upf_buffer_listener_error_indications_sent_total{node_id="enodeb-
1",peer_address="10.60.0.1"}
```

Étiquettes de Métriques :

- `node_id` : ID de nœud PFCP de l'association (ou "inconnu" si aucune association n'existe)
- `peer_address` : adresse IP du pair distant

Ces métriques aident à identifier les pairs problématiques et à suivre les modèles d'Indication d'Erreur par nœud de plan de contrôle.

Remarques Importantes :

1. **Nettoyage Automatique** : Pas besoin d'intervention de l'opérateur - les sessions sont supprimées automatiquement

2. **Correspondance de TEID** : Seules les sessions avec des FARs transférant vers le TEID erroné exact sont supprimées

3. **Isolation par Pair** : Les Indications d'Erreur d'un pair n'affectent que les sessions transférant vers ce pair

4. **Multiples Sessions** : Si plusieurs sessions transfèrent vers le même TEID mort, **toutes sont supprimées**

5. **Complémentaire à l'Horodatage de Récupération** :

    - Détection de l'Horodatage de Récupération = proactif (détecte le redémarrage lors de la configuration de l'association)

    - Gestion des Indications d'Erreur = réactif (détecte les tunnels morts lorsque le trafic circule)

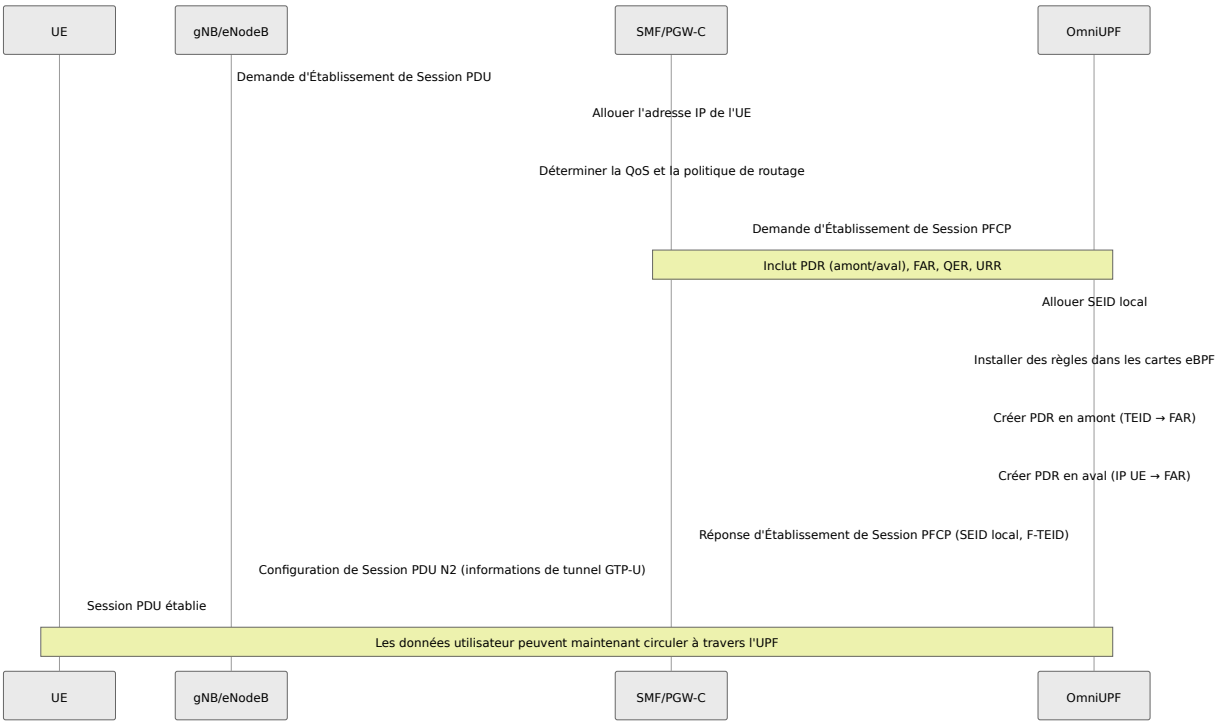
6. **Gestion des Paquets Malformés** : Les Indications d'Erreur invalides sont enregistrées et ignorées (aucune session supprimée)

Pour le dépannage des Indications d'Erreur, consultez le Guide de Dépannage.

Création de Session PFCP

Lorsqu'un UE établit une session PDU (5G) ou un contexte PDP (LTE), le SMF crée une session PFCP au niveau de l'UPF.

Flux d'Établissement de Session :



Contenu Typique de la Session :

- **PDR en amont** : Correspondre sur le TEID N3, transférer via FAR vers N6
- **PDR en aval** : Correspondre sur l'adresse IP de l'UE, transférer via FAR vers N3 avec encapsulation GTP-U
- **FAR** : Paramètres de transfert (création d'en-tête externe, instance réseau)
- **QER** : Limites de QoS (MBR, GBR) et marquage de paquets (QFI)
- **URR** : Rapport de volume pour la facturation (optionnel)

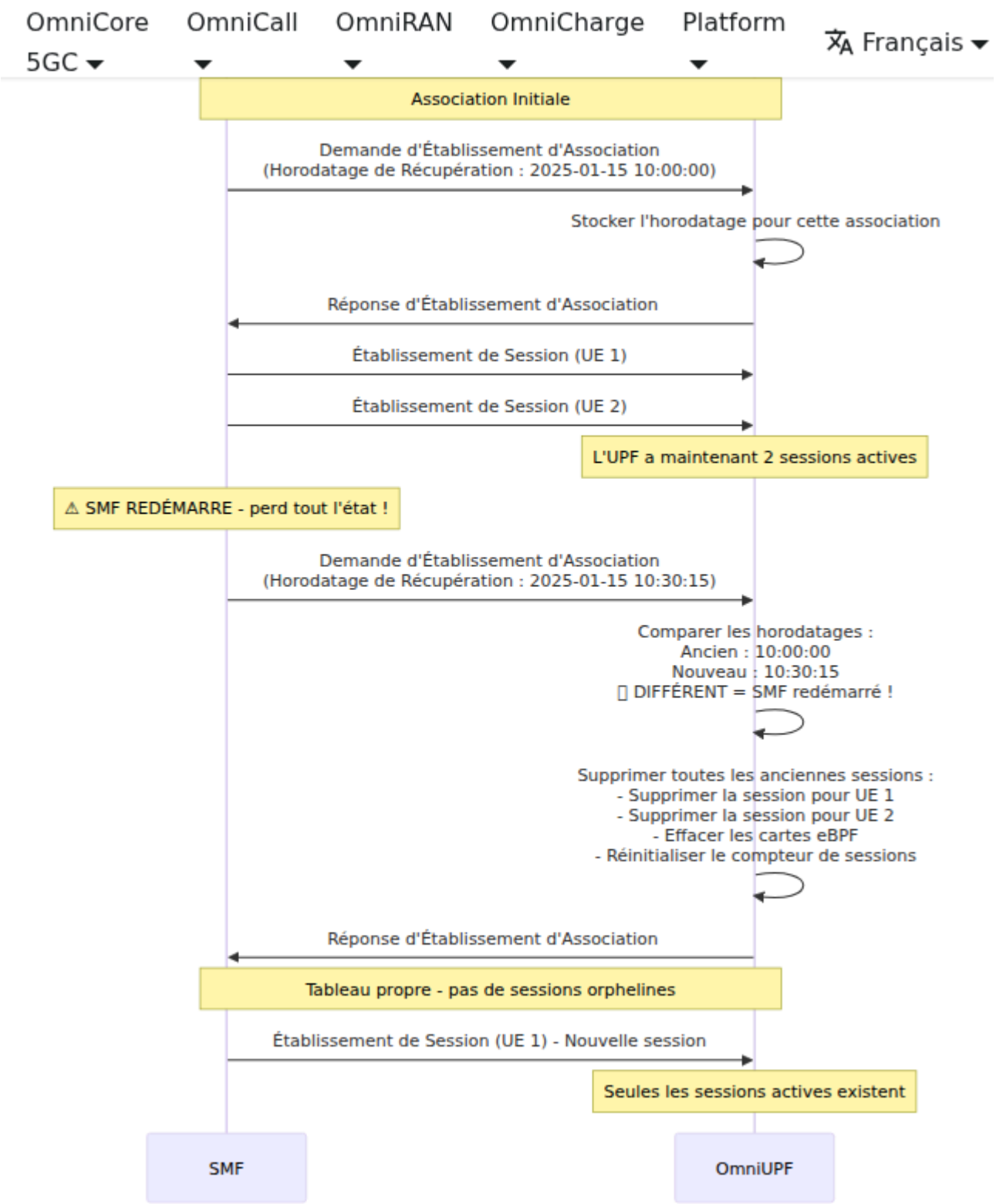
Modification de Session PFCP

Le SMF peut modifier les sessions pour des événements de mobilité (transfert), des changements de QoS ou des mises à jour de service.

Scénarios de Modification Courants :

1. **Transfert (basé sur N2)**
    - Mettre à jour le FAR en amont avec le nouveau point de terminaison de tunnel gNB (F-TEID)
    - Optionnellement, mettre en mémoire tampon les paquets pendant le changement de chemin
    - Vider le tampon vers le nouveau chemin lorsque prêt
2. **Changement de QoS**
    - Mettre à jour le QER avec de nouvelles valeurs MBR/GBR
    - Peut ajouter/supprimer des filtres SDF dans le PDR pour une QoS spécifique à l'application
3. **Mise à Jour de Service**
    - Ajouter de nouveaux PDR pour des flux de trafic supplémentaires
    - Modifier les FAR pour des changements de routage

Flux de Modification de Session :

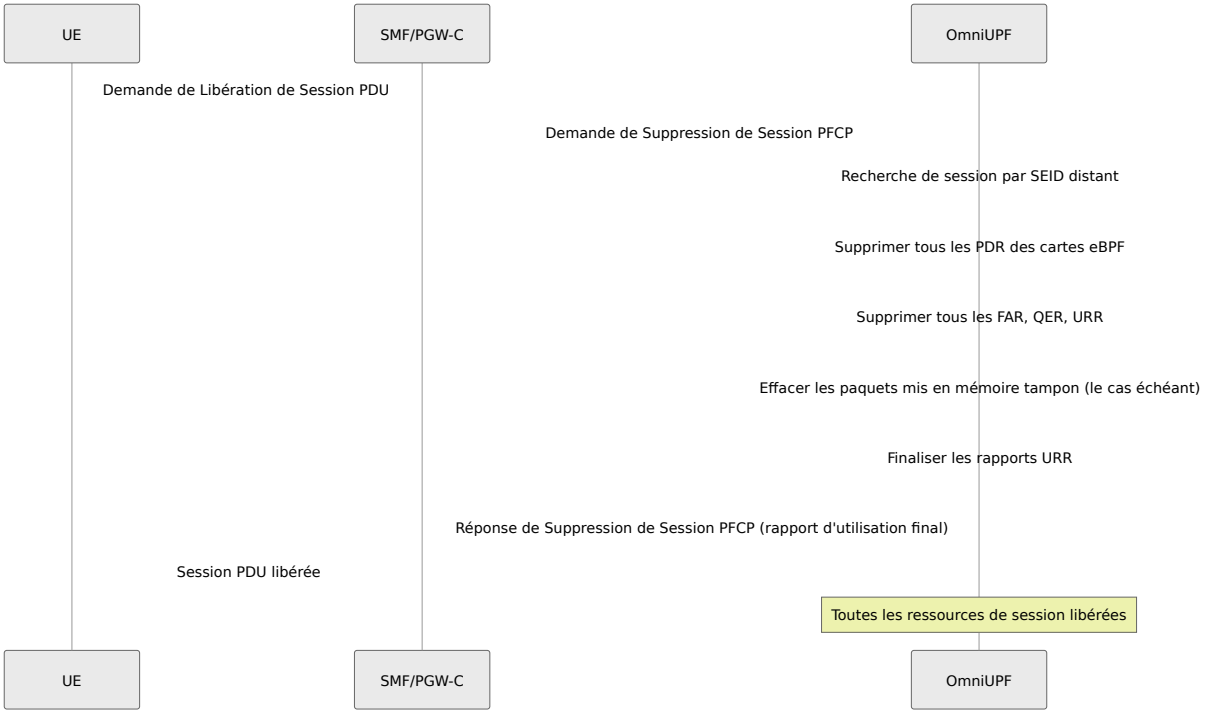


Pour la gestion des règles, consultez le Guide de Gestion des Règles.

Suppression de Session PFCP

Lorsqu'une session PDU est libérée, le SMF supprime la session PFCP au niveau de l'UPF.

Flux de Suppression de Session :



Nettoyage Effectué :

- Tous les PDR supprimés (amont et aval)
- Tous les FAR, QER, URR supprimés
- Tampons de paquets effacés
- Rapport d'utilisation final envoyé au SMF pour la facturation

Opérations Courantes

OmniUPF fournit des capacités opérationnelles complètes via son panneau de contrôle basé sur le web et son API REST. Cette section couvre les tâches opérationnelles courantes et leur signification.

Surveillance des Sessions

Comprendre les Sessions PFCP :

Les sessions PFCP représentent les sessions PDU actives de l'UE (5G) ou les contextes PDP (LTE). Chaque session contient :

- SEIDs locaux et distants (Identifiants de Point de Terminaison de Session)
- PDRs pour la classification des paquets
- FARs pour les décisions de transfert
- QERs pour l'application de la QoS (optionnel)
- URRs pour le suivi de l'utilisation (optionnel)

Opérations Clés sur les Sessions :

- **Voir toutes les sessions** avec adresses IP de l'UE, TEIDs et comptes de règles
- **Filtrer les sessions** par adresse IP ou TEID
- **Inspecter les détails de la session** y compris les configurations complètes PDR/FAR/QER/URR
- **Surveiller les comptes de session** par association PFCP

Pour des procédures détaillées sur les sessions, consultez Vue des Sessions.

Gestion des Règles

Règles de Détection de Paquets (PDR) :

Les PDR déterminent quels paquets correspondent à des flux de trafic spécifiques. Les opérateurs peuvent :

- **Voir les PDRs en amont** indexés par TEID de l'interface N3
- **Voir les PDRs en aval** indexés par adresse IP de l'UE (IPv4 et IPv6)
- **Inspecter les filtres SDF** pour la classification spécifique à l'application
- **Surveiller les comptes de PDR** et l'utilisation de la capacité

Règles d'Action de Transfert (FAR) :

Les FAR définissent quoi faire avec les paquets correspondants. Les opérateurs peuvent :

- **Voir les actions FAR** (TRANSFÉRER, SUPPRIMER, TAMBOUR, DUPLIQUER, NOTIFIER)
- **Inspecter les paramètres de transfert** (création d'en-tête externe, destination)
- **Surveiller l'état de mise en mémoire tampon** par FAR
- **Basculer la mise en mémoire tampon** pour des FAR spécifiques lors du dépannage

Règles d'Application de la QoS (QER) :

Les QER appliquent des limites de bande passante et un marquage de paquets. Les opérateurs peuvent :

- **Voir les paramètres de QoS** (MBR, GBR, budget de délai de paquet)
- **Surveiller les QERs actifs** par session
- **Inspecter les marquages QFI** pour les flux QoS 5G

Règles de Rapport d'Utilisation (URR) :

Les URRs suivent les volumes de données pour la facturation. Les opérateurs peuvent :

- **Voir les compteurs de volume** (amont, aval, total des octets)
- **Surveiller les seuils d'utilisation** et les déclencheurs de rapport
- **Inspecter les URRs actifs** à travers toutes les sessions

Pour les opérations sur les règles, consultez le Guide de Gestion des Règles.

Mise en Mémoire Tampon des Paquets

Pourquoi la Mise en Mémoire Tampon est Critique pour l'UPF

La mise en mémoire tampon des paquets est l'une des fonctions les plus importantes d'un UPF car elle empêche la perte de paquets lors des

événements de mobilité et des reconfigurations de session. Sans mise en mémoire tampon, les utilisateurs mobiles subiraient des connexions interrompues, des téléchargements interrompus et des communications en temps réel échouées chaque fois qu'ils se déplacent entre les tours cellulaires ou lorsque les conditions du réseau changent.

Le Problème : Perte de Paquets Lors de la Mobilité

Dans les réseaux mobiles, les utilisateurs se déplacent constamment. Lorsqu'un appareil passe d'une tour cellulaire à une autre (transfert), ou lorsque le réseau doit reconfigurer le chemin de données, il existe une fenêtre critique où les paquets sont en vol mais le nouveau chemin n'est pas encore prêt :

Après le Transfert

OmniUPF

Tunnel GTP-U
TEID: 67890

Nouveau gNB
Tour Cellulaire B

Pendant le Transfert -
Fenêtre Critique

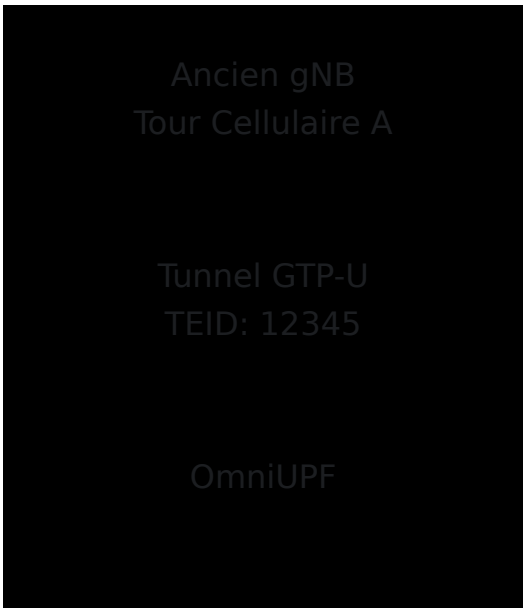
Paquets Descendants
arrivant pour l'UE

OmniUPF

Impossible d'envoyer

Où envoyer ?
Ancien chemin se ferme
Nouveau chemin pas
prêt

Avant le Transfert



Sans mise en mémoire tampon : Les paquets arrivant pendant cette fenêtre critique seraient **perdus**, provoquant :

- **Des connexions TCP qui se bloquent** ou se réinitialisent (navigation web, téléchargements interrompus)
- **Des appels vidéo qui se figent** ou se coupent (Zoom, Teams, appels WhatsApp échouent)
- **Des sessions de jeu qui se déconnectent** (jeux en ligne, applications en temps réel échouent)
- **Des appels VoIP qui ont des interruptions** ou se coupent complètement (appels téléphoniques interrompus)
- **Des téléchargements qui échouent** et doivent être redémarrés

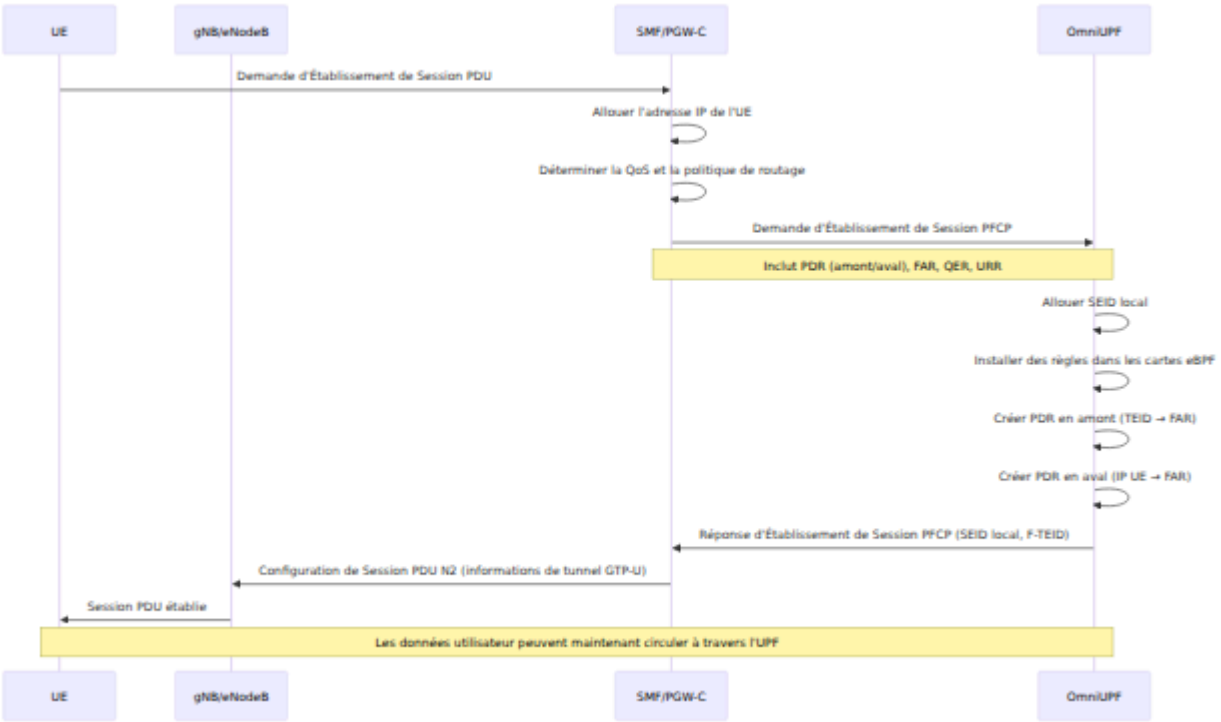
Avec mise en mémoire tampon : OmniUPF conserve temporairement les paquets jusqu'à ce que le nouveau chemin soit établi, puis les transfère sans interruption. L'utilisateur connaît **zéro interruption**.

Quand la Mise en Mémoire Tampon se Produit

OmniUPF met en mémoire tampon les paquets dans ces scénarios critiques :

1. Transfert Basé sur N2 (5G) / Transfert Basé sur X2 (4G)

Lorsque l'UE se déplace entre les tours cellulaires :



Chronologie :

- **T+0ms** : Ancien chemin toujours actif
- **T+10ms** : SMF dit à UPF de mettre en mémoire tampon (ancien chemin se ferme, nouveau chemin pas prêt)
- **T+10-50ms** : **Fenêtre critique de mise en mémoire tampon** - les paquets arrivent mais ne peuvent pas être transférés
- **T+50ms** : Nouveau chemin prêt, SMF dit à UPF de transférer
- **T+50ms+** : UPF vide les paquets mis en mémoire tampon vers le nouveau chemin, puis transfère les nouveaux paquets normalement

Sans mise en mémoire tampon : ~40ms de paquets (potentiellement des milliers) seraient **perdus**. **Avec mise en mémoire tampon** : Zéro perte de paquets, transfert sans interruption.

---

2. Modification de Session (Changement de QoS, Mise à Jour de Chemin)

Lorsque le réseau doit changer les paramètres de session :

- **Amélioration/diminution de la QoS** : L'utilisateur passe de la couverture 4G à 5G (mode NSA)

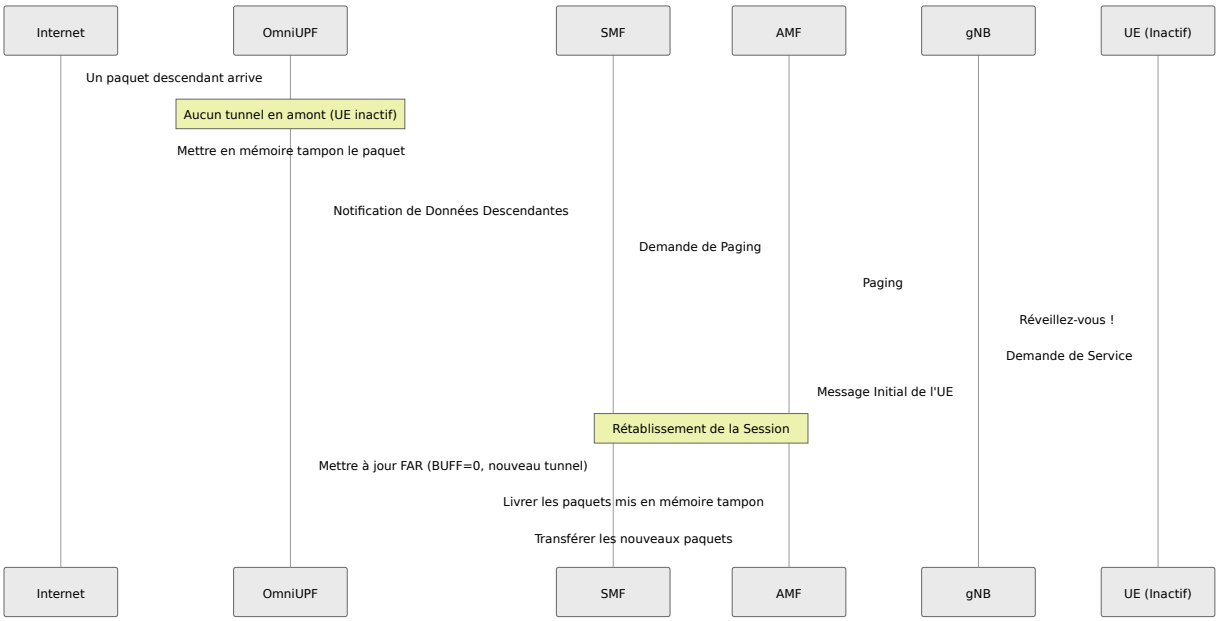
- **Changement de politique** : L'utilisateur d'entreprise entre dans le campus de l'entreprise (changement de direction du trafic)
- **Optimisation du réseau** : Le réseau central redirige le trafic vers un UPF plus proche (mise à jour ULCL)

Pendant la modification, le plan de contrôle peut avoir besoin de mettre à jour plusieurs règles de manière atomique. La mise en mémoire tampon garantit que les paquets ne sont pas transférés avec des ensembles de règles partiels/incohérents.

---

3. Notification de Données Descendantes (Récupération en Mode Inactif)

Lorsque l'UE est en mode inactif (écran éteint, économie de batterie) et que des données descendantes arrivent :



Sans mise en mémoire tampon : Le paquet initial qui a déclenché la notification serait **perdu**, nécessitant que l'expéditeur retransmette (ajoute de la latence). **Avec mise en mémoire tampon** : Le paquet qui a réveillé l'UE est livré immédiatement lorsque l'UE se reconnecte.

---

4. Transfert Inter-RAT (4G ↔ 5G)

Lorsque l'UE se déplace entre la couverture 4G et 5G :

- Changements d'architecture (eNodeB ↔ gNB)
- Changements de points de terminaison de tunnel (allocation de TEID différente)
- La mise en mémoire tampon garantit une transition fluide entre les types de RAT

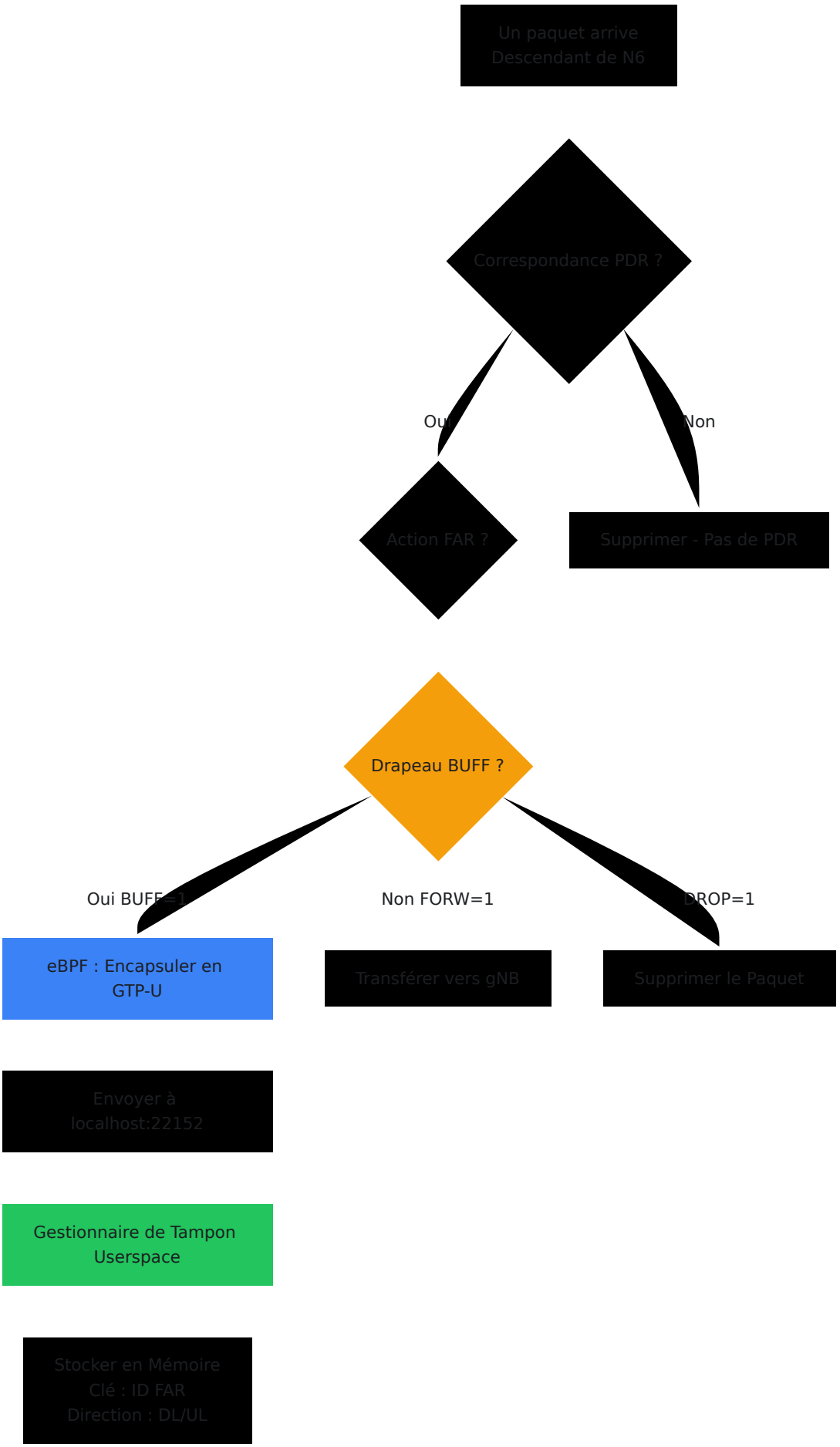
Comment la Mise en Mémoire Tampon Fonctionne dans OmniUPF

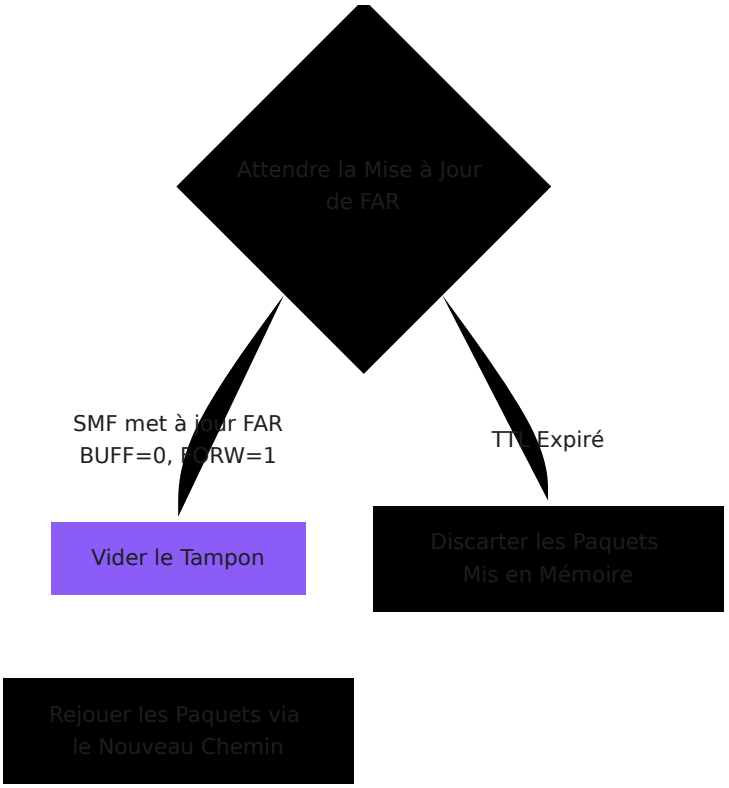
Mécanisme Technique :

OmniUPF utilise une **architecture de mise en mémoire tampon en deux étapes** :

1. **Étape eBPF (Noyau)** : Détecte les paquets nécessitant une mise en mémoire tampon en fonction des drapeaux d'action FAR
2. **Étape Userspace** : Stocke et gère les paquets mis en mémoire tampon en mémoire

Processus de Mise en Mémoire Tampon :


Un paquet arrive
Descendant de N6
Correspondance PDR ?
Oui
Non
Action FAR ?
Supprimer - Pas de PDR
Drapeau BUFF ?
Oui BUFF=1
Non FORW=1
DROP=1
eBPF : Encapsuler en GTP-U
Transférer vers gNB
Supprimer le Paquet
Envoyer à localhost:22152
Gestionnaire de Tampon Userspace
Stocker en Mémoire
Clé : ID FAR
Direction : DL/UL



Détails Clés :

- **Port de Tampon** : Port UDP 22152 (paquets envoyés de eBPF à userspace)
- **Encapsulation** : Paquets enveloppés dans GTP-U avec ID FAR comme TEID
- **Stockage** : Tampons en mémoire par FAR avec des métadonnées (horodatage, direction, taille du paquet)
- **Limites** :
  - Limite par FAR : 10 000 paquets (par défaut)
  - Limite globale : 100 000 paquets à travers tous les FARs
  - TTL : 30 secondes (par défaut) - les paquets plus anciens que le TTL sont supprimés
- **Nettoyage** : Un processus en arrière-plan supprime les paquets expirés toutes les 60 secondes

Cycle de Vie du Tampon :

1. **Mise en Mémoire Tampon Activée** : SMF définit l'action FAR BUFF=1 (bit 2) via la Modification de Session PFCP
2. **Paquets Mis en Mémoire Tampon** : eBPF détecte le drapeau BUFF, encapsule les paquets, envoie au port 22152

3. **Stockage Userspace** : Le gestionnaire de tampon stocke les paquets avec ID FAR, horodatage, direction
4. **Mise en Mémoire Tampon Désactivée** : SMF définit l'action FAR FORW=1, BUFF=0 avec de nouveaux paramètres de transfert
5. **Vider le Tampon** : Userspace rejoue les paquets mis en mémoire tampon en utilisant les nouvelles règles FAR (nouveau point de terminaison de tunnel)
6. **Reprendre Normal** : Les nouveaux paquets sont transférés immédiatement via le nouveau chemin

---

Pourquoi Cela Compte pour l'Expérience Utilisateur

Impact dans le Monde Réel :

Scénario	Sans Mise en Mémoire Tampon	Avec Mise en Mémoire Tampon
Appel Vidéo Pendant le Transfert	L'appel se fige pendant 1 à 2 secondes, peut se couper	Sans interruption, fluide
Téléchargement de Fichier à la Limite Cellulaire	Le téléchargement échoue, doit redémarrer	Le téléchargement continue sans interruption
Jeu en Ligne en Se Déplaçant	La connexion se coupe, déconnexion du jeu	Jeu fluide, pas de déconnexions
Appel VoIP dans la Voiture	L'appel se coupe à chaque transfert	Clair, sans coupures
Vidéo en Streaming dans un Train	La vidéo se met en mémoire tampon, la qualité baisse	Lecture fluide
Point d'Accès Mobile pour Ordinateur Portable	La session SSH se coupe, l'appel vidéo échoue	Toutes les connexions maintenues

Avantages pour les Opérateurs de Réseau :

- **Taux de Chute d'Appel Réduit (CDR)** : KPI critique pour la qualité du réseau
- **Satisfaction Client Élevée** : Les utilisateurs ne remarquent pas les transferts
- **Coûts de Support Réduits** : Moins de plaintes concernant les connexions perdues
- **Avantage Concurrentiel** : Marketing "Meilleur réseau pour la couverture"

Opérations de Gestion de Tampon

Les opérateurs peuvent surveiller et contrôler le tampon via l'interface Web et l'API :

Surveillance :

- **Voir les paquets mis en tampon** par ID FAR (nombre, octets, âge)
- **Suivre l'utilisation du tampon** par rapport aux limites (par FAR, global)
- **Alerte sur le débordement de tampon** ou la durée de tampon excessive
- **Identifier les tampons bloqués** (paquets mis en tampon > seuil TTL)

Opérations de Contrôle :

- **Vider les tampons** : Déclencher manuellement la lecture du tampon (dépannage)
- **Effacer les tampons** : Jeter les paquets mis en tampon (nettoyer les tampons bloqués)
- **Ajuster TTL** : Modifier le temps d'expiration des paquets
- **Modifier les limites** : Augmenter la capacité du tampon par FAR ou global

Dépannage :

- **Tampon ne se vidant pas** : Vérifiez si le SMF a envoyé une mise à jour FAR pour désactiver le tampon
- **Débordement de tampon** : Augmenter les limites ou enquêter sur la raison pour laquelle la durée de tampon est excessive
- **Anciens paquets dans le tampon** : Le TTL peut être trop élevé, ou la mise à jour FAR retardée
- **Tampon excessif** : Peut indiquer des problèmes de mobilité ou des problèmes de SMF

Pour des opérations de tampon détaillées, voir Guide de Gestion de Tampon.

Configuration de Tampon

Configurer le comportement de tampon dans `/etc/omniupf/runtime.exs` :

```
# Paramètres de tampon
buffer_port = 22152                # Port UDP pour les paquets mis
en tampon (par défaut)
```

Recommandations :

- **Réseaux à haute mobilité** (autoroutes, trains) : Augmenter `buffer_max_packets` à 20 000+
- **Zones urbaines denses** (transferts fréquents) : Diminuer `buffer_packet_ttl` à 15s
- **Applications à faible latence** : Définir `buffer_packet_ttl` à 10s pour éviter les données obsolètes
- **Réseaux IoT** : Diminuer les limites (les appareils IoT génèrent moins de trafic pendant le transfert)

Pour les options de configuration complètes, voir Guide de Configuration.

Statistiques et Surveillance

Statistiques de Paquets :

Métriques de traitement des paquets en temps réel, y compris :

- **Paquets RX** : Total reçu de toutes les interfaces
- **Paquets TX** : Total transmis à toutes les interfaces
- **Paquets perdus** : Paquets rejetés en raison d'erreurs ou de politiques
- **Paquets GTP-U** : Comptage des paquets encapsulés

Statistiques de Route :

Métriques de transfert par route :

- **Hits de route** : Paquets correspondants à chaque route
- **Comptes de transfert** : Succès/échec par destination
- **Compteurs d'erreur** : TEIDs invalides, IPs UE inconnues

Statistiques XDP :

Métriques de performance du chemin de données eXpress :

- **XDP traité** : Paquets gérés au niveau XDP
- **XDP passé** : Paquets envoyés à la pile réseau
- **XDP rejeté** : Paquets rejetés au niveau XDP
- **XDP avorté** : Erreurs de traitement

Statistiques de l'Interface N3/N6 :

Compteurs de trafic par interface :

- **N3 RX/TX** : Trafic vers/depuis RAN (gNB/eNodeB)
- **N6 RX/TX** : Trafic vers/depuis le réseau de données
- **Comptes totaux de paquets** : Statistiques agrégées par interface

Pour des détails de surveillance, voir Guide de Surveillance.

Gestion de Capacité

Surveillance de la Capacité de la Carte eBPF :

La performance du UPF dépend de la capacité de la carte eBPF. Les opérateurs peuvent :

- **Surveiller l'utilisation de la carte** avec des indicateurs de pourcentage en temps réel
- **Voir les limites de capacité** pour chaque carte eBPF
- **Alertes codées par couleur** :
  - Vert (<50%) : Normal
  - Jaune (50-70%) : Précaution
  - Ambre (70-90%) : Avertissement
  - Rouge (>90%) : Critique

Cartes Critiques à Surveiller :

- `uplink_pdr_map` : Classification du trafic montante
- `downlink_pdr_map` : Classification du trafic IPv4 descendante
- `far_map` : Règles de transfert
- `qer_map` : Règles de QoS
- `urr_map` : Suivi d'utilisation

Planification de Capacité :

- Chaque PDR consomme une entrée de carte (taille de clé + taille de valeur)
- La capacité de la carte est configurée au démarrage du UPF (limite de mémoire du noyau)
- Dépasser la capacité entraîne des échecs d'établissement de session

Pour la surveillance de la capacité, voir Gestion de Capacité.

Gestion de Configuration

Configuration du UPF :

Voir et vérifier les paramètres opérationnels du UPF :

- **Interface N3** : Adresse IP pour la connectivité RAN (GTP-U)
- **Interface N6** : Adresse IP pour la connectivité au réseau de données
- **Interface N9** : Adresse IP pour la communication inter-UPF (optionnel)
- **Interface PFCP** : Adresse IP pour la connectivité SMF
- **Port API** : Port d'écoute de l'API REST
- **Point de terminaison des métriques** : Port des métriques Prometheus

Configuration du Plan de Données :

Paramètres actifs du chemin de données eBPF :

- **Adresse N3 active** : Liaison de l'interface N3 en temps réel
- **Adresse N9 active** : Liaison de l'interface N9 en temps réel (si activée)

Pour voir la configuration, voir Vue de Configuration.

Dépannage

Cette section couvre les problèmes opérationnels courants et leurs stratégies de résolution.

Échecs d'Établissement de Session

Symptômes : Les sessions PFCP échouent à se créer, l'UE ne peut pas établir de connectivité de données

Causes Racines Courantes :

1. **Association PFCP Non Établie**
   - Vérifiez que le SMF peut atteindre l'interface PFCP du UPF (port 8805)
   - Vérifiez l'état de l'association PFCP dans la vue des Sessions
   - Vérifiez que la configuration de l'ID de nœud correspond entre le SMF et le UPF
2. **Capacité de Carte eBPF Épuisée**
   - Vérifiez la vue de Capacité pour une utilisation de carte rouge (>90%)
   - Augmentez les tailles de carte eBPF dans la configuration du UPF
   - Supprimez les sessions obsolètes si la carte est pleine
3. **Configuration PDR/FAR Invalide**
   - Vérifiez que l'adresse IP de l'UE est unique et valide
   - Vérifiez que l'allocation TEID ne crée pas de conflit
   - Assurez-vous que le FAR fait référence à des instances réseau valides
4. **Problèmes de Configuration d'Interface**
   - Vérifiez que l'IP de l'interface N3 est accessible depuis le gNB
   - Vérifiez les tables de routage pour la connectivité N6 au réseau de données
   - Confirmez que le trafic GTP-U n'est pas bloqué par un pare-feu

Pour un dépannage détaillé, voir Guide de Dépannage.

Problèmes de Perte de Paquets ou de Transfert

Symptômes : L'UE a une connectivité mais subit une perte de paquets ou aucun flux de trafic

Causes Racines Courantes :

1. **Mauvaise Configuration de PDR**
   - Vérifiez que le TEID PDR montante correspond au TEID assigné par le gNB
   - Vérifiez que le PDR descendante IP de l'UE correspond à l'IP assignée
   - Inspectez les filtres SDF pour des règles trop restrictives
2. **Problèmes d'Action FAR**
   - Vérifiez que l'action FAR est FORWARD (pas DROP ou BUFFER)
   - Vérifiez les paramètres de création d'en-tête extérieur pour GTP-U
   - Assurez-vous que le point de terminaison de destination est correct
3. **Limites de QoS Dépassées**
   - Vérifiez les paramètres MBR (Débit Maximum) du QER
   - Vérifiez l'allocation GBR (Débit Garanti)
   - Surveillez les pertes de paquets dues à la limitation de débit
4. **Problèmes de MTU d'Interface**
   - Vérifiez que le surcoût GTP-U (40-50 octets) ne provoque pas de fragmentation
   - Vérifiez la configuration MTU des interfaces N3/N6
   - Surveillez les messages ICMP indiquant que la fragmentation est nécessaire

Problèmes Liés au Tampon

Symptômes : Paquets mis en tampon indéfiniment, débordement de tampon

Causes Racines Courantes :

1. **Tampon Non Désactivé Après Transfert**
   - Vérifiez le drapeau de tampon FAR (bit 2)
   - Vérifiez que le SMF a envoyé une Modification de Session pour désactiver le tampon
   - Désactivez manuellement le tampon via le panneau de contrôle si bloqué
2. **Expiration du TTL du Tampon**
   - Vérifiez l'âge des paquets dans la vue du tampon
   - Vérifiez la configuration TTL du tampon (par défaut peut être trop long)
   - Effacez manuellement les tampons expirés
3. **Capacité du Tampon Épuisée**
   - Surveillez l'utilisation totale du tampon et les limites par FAR
   - Vérifiez les règles mal configurées provoquant un tampon excessif
   - Ajustez les limites max_per_far et max_total du tampon

Pour le dépannage du tampon, voir Opérations de Tampon.

Anomalies Statistiques

Symptômes : Compteurs de paquets inattendus, statistiques manquantes

Causes Racines Courantes :

1. **Débordement de Compteur**
   - Les cartes eBPF utilisent des compteurs 64 bits (ne devraient pas déborder)

    - Vérifiez les événements de réinitialisation de compteur dans les journaux
    - Vérifiez que le rapport URR fonctionne

2. **Statistiques de Route Non Mises à Jour**

    - Vérifiez que le programme eBPF est attaché aux interfaces
    - Vérifiez que la version du noyau prend en charge les fonctionnalités eBPF requises
    - Passez en revue les statistiques XDP pour des erreurs de traitement

3. **Mésentente des Statistiques d'Interface**

    - Comparez les statistiques N3/N6 avec les compteurs d'interface du noyau
    - Vérifiez le trafic contournant eBPF (par exemple, routage local)
    - Vérifiez que tout le trafic passe par les crochets XDP

Dégradation de Performance

Symptômes : Latence élevée, faible débit, saturation du CPU

Diagnostic :

1. **Surveillez les Statistiques XDP** : Vérifiez les rejets ou avortements XDP
2. **Vérifiez le Temps d'Accès à la Carte eBPF** : Les recherches par hachage devraient être sub-microsecondes
3. **Passez en Revue l'Utilisation du CPU** : eBPF devrait se répartir sur les cœurs
4. **Analysez l'Interface Réseau** : Vérifiez que le NIC prend en charge le déchargement XDP

Considérations de Scalabilité :

- **Performance XDP** : 10M+ paquets par seconde par cœur
- **Capacité PDR** : Des millions de PDRs limités uniquement par la mémoire du noyau

- **Nombre de Sessions** : Des milliers de sessions simultanées par instance UPF
- **Débit** : Débit multi-gigabit avec un déchargement NIC approprié

Pour l'optimisation des performances, voir Guide d'Architecture.

Documentation Supplémentaire

Guides d'Opérations Spécifiques aux Composants

Pour des opérations détaillées et le dépannage pour chaque composant UPF :

Guide de Configuration

Référence complète de configuration incluant :

- Paramètres de configuration (YAML, variables d'environnement, CLI)
- Modes de fonctionnement (UPF/PGW-U/SGW-U)
- Vue d'ensemble des modes d'attachement XDP
- Compatibilité avec les hyperviseurs (Proxmox, VMware, KVM, Hyper-V, VirtualBox)
- Compatibilité des NIC et support des pilotes XDP
- Exemples de configuration pour différents scénarios
- Dimensionnement des cartes et planification de capacité

Guide des Modes XDP

Configuration et optimisation détaillées de XDP incluant :

- Modes d'attachement XDP expliqués (générique/natif/déchargement)
- Comparaison de performance et benchmarks
- Configuration pas à pas de XDP natif sur Proxmox VE
- Configuration multi-queue pour une performance optimale
- Configuration XDP sur VMware ESXi, KVM et Hyper-V
- Vérification et dépannage de XDP

- Sélection de matériel pour la performance XDP

Guide d'Architecture

Plongée technique approfondie incluant :

- Fondement technologique eBPF et cycle de vie du programme
- Pipeline de traitement des paquets XDP avec appels de queue
- Mise en œuvre du protocole PFCP
- Architecture de tampon (encapsulation GTP-U au port 22152)
- Limitation de débit par fenêtre glissante QoS (fenêtre de 5ms)
- Caractéristiques de performance (latence de 3,5µs, 10 Mpps/cœur)

Guide de Gestion des Règles

Référence des règles PFCP incluant :

- Règles de Détection de Paquets (PDR) - Classification du trafic
- Règles d'Action de Transfert (FAR) - Décisions de routage avec drapeaux d'action
- Règles d'Application de QoS (QER) - Gestion de la bande passante (MBR/GBR)
- Règles de Rapport d'Utilisation (URR) - Suivi et rapport de volume
- Diagrammes de flux de paquets montants et descendants
- Logique de traitement des règles et priorité

Guide de Surveillance

Statistiques et gestion de capacité incluant :

- Statistiques d'interface N3/N6 et distribution du trafic
- Statistiques de traitement XDP (passer/rejeter/rediriger/avorter)
- Surveillance de la capacité de carte eBPF avec zones codées par couleur
- Métriques de performance (taux de paquets, débit, taux de perte)
- Formules de planification de capacité et estimation des sessions
- Seuils d'alerte et meilleures pratiques

Guide des Opérations de l'Interface Web

Utilisation du panneau de contrôle incluant :

- Vue d'ensemble du tableau de bord et navigation
- Surveillance des sessions (états sains/malades)
- Inspection des règles (détails PDR, FAR, QER, URR)
- Surveillance du tampon et état de mise en tampon des paquets
- Tableau de bord des statistiques en temps réel
- Visualisation de la capacité de carte eBPF
- Vue de configuration

Documentation API

Référence complète de l'API REST incluant :

- Documentation interactive OpenAPI/Swagger
- Pagination de l'API (basée sur les pages et basée sur les décalages)
- Points de terminaison des sessions et associations PFCP
- Règles de Détection de Paquets (PDR) - IPv4 et IPv6
- Règles d'Action de Transfert (FAR)
- Règles d'Application de QoS (QER)
- Règles de Rapport d'Utilisation (URR)
- Gestion des tampons de paquets
- Points de terminaison de statistiques et de surveillance
- Gestion des routes et intégration FRR
- Informations sur les cartes eBPF
- Gestion de la configuration
- Directives d'authentification et de sécurité
- Flux de travail API courants et exemples

Référence des Métriques

Documentation des métriques Prometheus incluant :

- Métriques de messages PFCP (compteurs, latence, erreurs par pair)

- Métriques d'action XDP (verdicts du plan de données)
- Métriques de paquets (compteurs au niveau du protocole avec étiquettes packet_type)
- Métriques de session et d'association PFCP (par nœud de plan de contrôle)
- Métriques URR (volume de trafic par pair PFCP)
- Métriques de mise en tampon de paquets (état du tampon, capacité, débit)
- Métriques de notification de Rapport de Données Descendantes (DLDR)
- Métriques de capacité de carte eBPF (utilisation des ressources)
- Exemples de configuration Prometheus
- Recommandations de tableau de bord Grafana

Référence des Codes de Cause PFCP

Documentation des codes d'erreur PFCP incluant :

- Définitions des codes de cause et conformité 3GPP (TS 129.244)
- Quand chaque code de cause se produit (succès, erreurs client, erreurs serveur)
- Scénarios d'échec courants avec résolutions
- Dépannage avec des métriques Prometheus
- Échecs de configuration d'association et de cycle de vie de session
- Étapes de débogage pour des taux de rejet élevés
- Recommandations d'alerte pour les codes de cause

Guide de Gestion des Routes UE

Intégration de routage FRR incluant :

- Vue d'ensemble et architecture de FRR (Free Range Routing)
- Cycle de vie de synchronisation des routes UE
- Synchronisation automatique des routes vers le démon de routage
- Annonce de routes via OSPF et BGP
- Surveillance des voisins OSPF
- Vérification de la base de données LSA externe OSPF
- Gestion de session de pair BGP
- Interface de surveillance des routes de l'interface Web

- Opérations de synchronisation manuelle des routes
- Diagrammes Mermaid pour le flux de routes et l'architecture

Guide IPv6 / Dual-Stack

Fonctionnement IPv6 du plan utilisateur pour les sessions PDN IPv6 et IPv4v6 :

- Prérequis du noyau (activation et transfert IPv6)
- Configuration du pool d'adresses IPv6 de l'UE
- Annonce OSPFv3 des routes hôtes UE `/128`
- Comportement du plan de données IPv6 descendant/montant
- Dépannage de la transmission IPv6 et de l'annonce de route

Guide de Dépannage

Diagnostic complet des problèmes incluant :

- Liste de contrôle de diagnostic rapide et outils
- Problèmes d'installation et de configuration
- Échecs d'association PFCP
- Problèmes de traitement des paquets
- Erreurs XDP et eBPF
- Dégradation de performance
- Problèmes spécifiques aux hyperviseurs (Proxmox, VMware, VirtualBox)
- Problèmes de NIC et de pilotes
- Procédures de résolution étape par étape

Documentation par Cas d'Utilisation

Installation et Configuration d'OmniUPF

1. Commencez par ce guide pour un aperçu
2. Guide de Configuration pour les paramètres de configuration
3. Guide de l'Interface Web pour accéder au panneau de contrôle

Déploiement sur Proxmox

1. Guide des Modes XDP - Configuration XDP Natif sur Proxmox - **Commencez ici pour la performance**
2. Guide de Configuration - Compatibilité avec les Hyperviseurs
3. Guide de Configuration - Configuration SR-IOV sur Proxmox
4. Dépannage - Problèmes Proxmox

Optimisation de la Performance

1. Guide des Modes XDP - **Activez XDP natif pour un boost de performance de 5 à 10x**
2. Guide d'Architecture - Optimisation de Performance
3. Guide de Configuration - Modes XDP
4. Guide de Surveillance - Métriques de Performance
5. Dépannage - Problèmes de Performance

Comprendre le Traitement des Paquets

1. Guide d'Architecture - Pipeline de Traitement des Paquets
2. Guide de Gestion des Règles
3. Guide de Surveillance - Statistiques

Planification de Capacité

1. Guide de Configuration - Dimensionnement des Cartes
2. Guide de Surveillance - Planification de Capacité
3. Guide de Surveillance - Estimation de Capacité des Sessions

Gestion des Routes UE et Intégration FRR

1. Guide de Gestion des Routes UE - Guide complet d'intégration de routage
2. Documentation API - Gestion des Routes - Points de terminaison API de routage
3. Guide de l'Interface Web - Opérations de la page des routes
4. Gestion des Routes UE - Vérification FRR - Vérification LSA OSPF

Utilisation de l'API REST

1. Documentation API - Référence API complète

2. Documentation API - Swagger UI - Explorateur API interactif
3. Documentation API - Fonctionnalités de l'API - Exemples d'utilisation de l'API
4. Guide de l'Interface Web - Interface Web comme exemple de client API

Dépannage des Problèmes

1. Guide de Dépannage - Commencez ici
2. Guide de Surveillance - Vérifiez les statistiques et la capacité
3. Guide de l'Interface Web - Utilisez les diagnostics du panneau de contrôle

---

Référence Rapide

Points de Terminaison API Courants

OmniUPF fournit une API REST pour la surveillance et la gestion :

```
# Statut et santé
GET http://localhost:8080/api/v1/upf_status

# Associations PFCP
GET http://localhost:8080/api/v1/upf_pipeline

# Sessions
GET http://localhost:8080/api/v1/sessions

# Statistiques
GET http://localhost:8080/api/v1/packet_stats
GET http://localhost:8080/api/v1/xdp_stats

# Surveillance de capacité
GET http://localhost:8080/api/v1/map_info

# Statistiques de tampon
GET http://localhost:8080/api/v1/upf_buffer_info
```

Pour la documentation API complète, accédez à l'interface Swagger à `http://<upf-ip>:8080/swagger/index.html`

Paramètres de Configuration Essentiels

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"              # Interfaces pour le trafic
N3/N6/N9
xdp_attach_mode = "native"           # "générique" | "natif" |
"déchargement"
n3_address = "10.100.50.233"         # IP de l'interface N3
pfcp_address = "10.100.50.241"       # Adresse d'écoute PFCP
pfcp_port = 8805                     # Port PFCP
node_id = "10.100.50.241"            # ID de nœud PFCP

# Capacité
max_sessions = 100_000               # Maximum de sessions
simultanées

# API
api_port = 8080                      # Port de l'API REST
```

Seuils de Surveillance Importants

- **Capacité de Carte eBPF < 70%** : Fonctionnement normal
- **Capacité de Carte eBPF 70-90%** : Planifiez une augmentation de capacité dans la semaine
- **Capacité de Carte eBPF > 90%** : Critique - action immédiate requise
- **Taux de Perte de Paquets < 0.1%** : Excellent
- **Taux de Perte de Paquets 0.1-1%** : Bon - problèmes mineurs
- **Taux de Perte de Paquets > 5%** : Critique - enquêtez immédiatement
- **XDP Avorté > 0** : Problème critique avec le programme eBPF

Référence des Normes 3GPP

OmniUPF met en œuvre les spécifications 3GPP suivantes :

Spécification	Titre	Pertinence
TS 23.501	Architecture système pour le Système 5G (5GS)	Architecture et interfaces UPF 5G
TS 23.401	Améliorations du Service de Paquet Général (GPRS) pour l'accès E-UTRAN	Architecture UPF LTE (PGW-U)
TS 29.244	Interface entre les nœuds de Plan de Contrôle et de Plan Utilisateur (PFCP)	Protocole PFCP N4
TS 29.281	Protocole de Tunnel de Système de Paquet Général (GTPv1-U)	Encapsulation GTP-U
TS 23.503	Cadre de contrôle de politique et de facturation pour le Système 5G (5GS)	QoS et facturation
TS 29.212	Contrôle de Politique et de Facturation (PCC)	Application de QoS

Glossaire

Termes de l'Architecture 5G

- **3GPP** : 3rd Generation Partnership Project - Organe de normalisation pour les télécommunications mobiles
- **AMF** : Fonction de Gestion d'Accès et de Mobilité - Élément du réseau central 5G pour le contrôle d'accès
- **CHF** : Fonction de Facturation - Système de facturation 5G
- **DN** : Réseau de Données - Réseau externe (Internet, IMS, entreprise)
- **eNodeB** : Evolved Node B - Station de base LTE

- **F-TEID** : Identifiant de Point de Tunnel Entièrement Qualifié - ID de tunnel GTP-U avec adresse IP
- **gNB** : Next Generation Node B - Station de base 5G
- **GTP-U** : Protocole de Tunnel de Système de Paquet Général - Protocole de tunneling pour les données utilisateur
- **MBR** : Débit Maximum - Paramètre QoS pour la bande passante maximale autorisée
- **GBR** : Débit Garanti - Paramètre QoS pour la bande passante minimale garantie
- **N3** : Interface entre RAN et UPF (trafic de plan utilisateur)
- **N4** : Interface entre SMF et UPF (contrôle PFCP)
- **N6** : Interface entre UPF et Réseau de Données (trafic de plan utilisateur)
- **N9** : Interface entre deux UPFs (trafic de plan utilisateur inter-UPF)
- **PCF** : Fonction de Contrôle de Politique - Serveur de politique 5G
- **PDU** : Unité de Données de Protocole - Session de données en 5G
- **PGW-C** : Passerelle PDN Plan de Contrôle - Équivalent LTE du SMF
- **PGW-U** : Passerelle PDN Plan Utilisateur - Plan utilisateur LTE (équivalent UPF)
- **QFI** : Identifiant de Flux QoS - Marquage de flux QoS 5G
- **QoS** : Qualité de Service - Priorisation du trafic et gestion de la bande passante
- **RAN** : Réseau d'Accès Radio - Réseau de station de base (gNB/eNodeB)
- **SEID** : Identifiant de Point de Session - ID de session PFCP
- **SMF** : Fonction de Gestion de Session - Élément du réseau central 5G pour le contrôle de session
- **TEID** : Identifiant de Point de Tunnel - ID de tunnel GTP-U
- **UE** : Équipement Utilisateur - Dispositif mobile
- **UPF** : Fonction de Plan Utilisateur - Élément de réseau de transfert de paquets 5G

Termes du protocole PFCP

- **Association** : Relation de contrôle entre SMF et UPF

- **FAR** : Règle d'action de transfert - Détermine le comportement de transfert des paquets
- **IE** : Élément d'information - Composant du message PFCP
- **Node ID** : Identifiant de l'UPF ou du SMF (FQDN ou adresse IP)
- **PDR** : Règle de détection de paquet - Classe les paquets en flux
- **PFCP** : Protocole de contrôle de transfert de paquets - Protocole de contrôle N4
- **QER** : Règle d'application de QoS - Applique des limites de bande passante et du marquage
- **SDF** : Flux de données de service - Filtre de trafic spécifique à l'application
- **Session** : Session PFCP représentant la session PDU de l'UE ou le contexte PDP
- **URR** : Règle de rapport d'utilisation - Suit les volumes de données pour la facturation

Termes eBPF et noyau Linux

- **BPF** : Filtre de paquets Berkeley - Technologie de filtrage de paquets du noyau
- **eBPF** : BPF étendu - Chemin de données du noyau programmable
- **Hash Map** : Magasin de paires clé-valeur eBPF pour des recherches rapides
- **XDP** : Chemin de données eXpress - Traitement de paquets du noyau au niveau du pilote
- **Verifier** : Composant du noyau qui valide les programmes eBPF pour la sécurité
- **Map** : Structure de données eBPF partagée entre le noyau et l'espace utilisateur
- **Zero-copy** : Traitement de paquets sans copie vers l'espace utilisateur

Termes du produit OmniUPF

- **OmniUPF** : Fonction de plan utilisateur basée sur eBPF (ce produit)
- **Datapath** : Moteur de traitement de paquets (programmes eBPF)
- **Control Plane** : Gestionnaire PFCP et gestion des sessions

- **REST API** : API HTTP pour la surveillance et la gestion
- **Web UI** : Panneau de contrôle basé sur le navigateur

Guide d'Architecture OmniUPF

Table des Matières

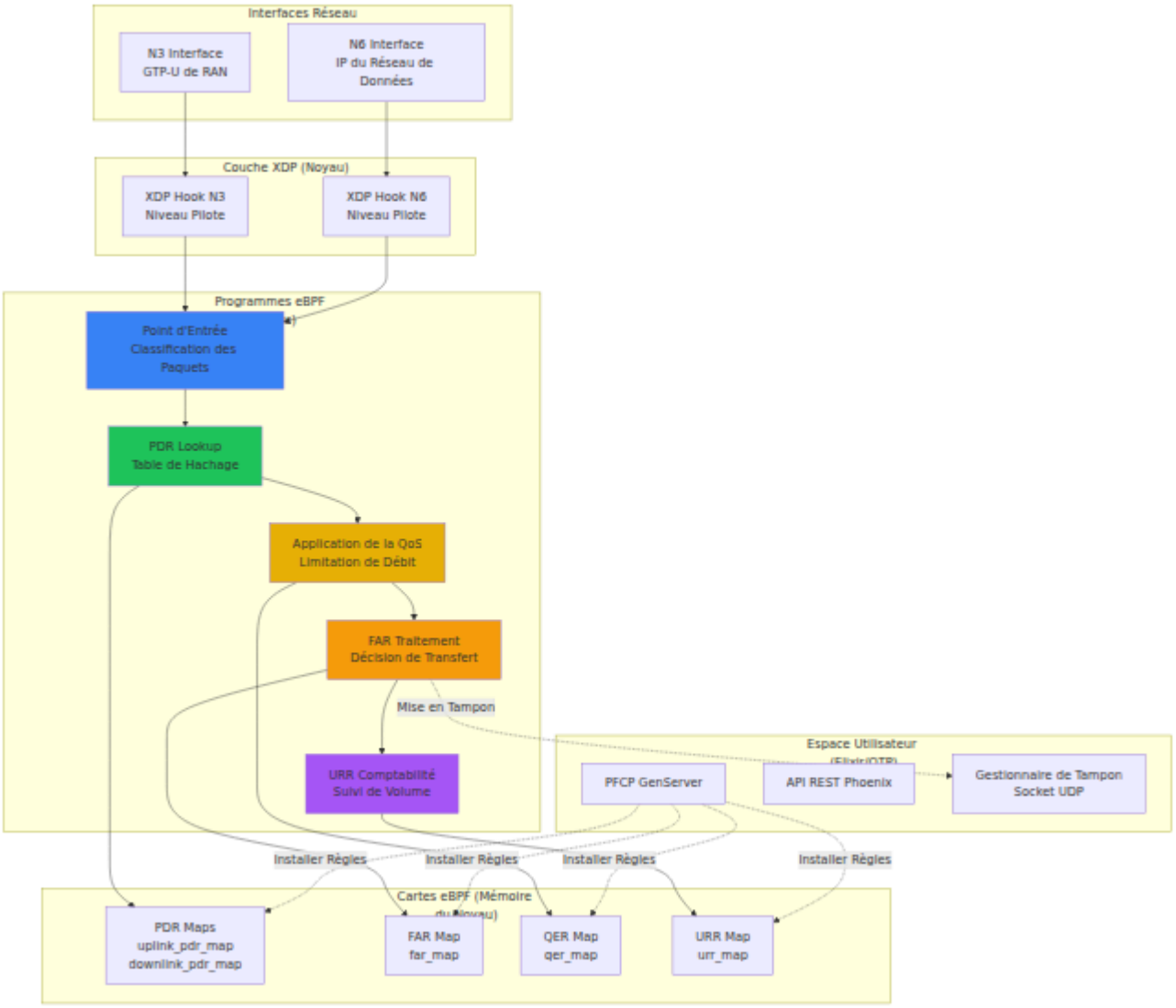
1. Aperçu
2. Fondation de la Technologie eBPF
3. Chemin de Données XDP
4. Pipeline de Traitement des Paquets
5. Architecture des Cartes eBPF
6. Mécanisme de Mise en Tampon
7. Application de la QoS
8. Caractéristiques de Performance
9. Scalabilité et Réglage

Aperçu

OmniUPF exploite eBPF (extended Berkeley Packet Filter) et XDP (eXpress Data Path) pour atteindre des performances de niveau opérateur pour le traitement des paquets 5G/LTE. Le plan de contrôle est implémenté en Elixir/OTP en utilisant la gestion de session PFCP basée sur GenServer, tandis que le plan de données exécute des programmes eBPF directement dans le noyau Linux. Cette séparation élimine le surcoût du traitement des paquets en espace utilisateur et atteint un débit multi-gigabit avec une latence en microsecondes.

Le fichier objet eBPF (`ipentrypoint_bpf.o`) est compilé à partir de la source C (dans le répertoire `ebpf/xdp/`) et chargé à l'exécution par le plan de contrôle Elixir via des liaisons NIF à libbpf.

Couches d'Architecture



Principes de Conception Clés

Traitement Zero-Copy:

- Paquets traités entièrement dans l'espace noyau
- Pas de copie de données entre le noyau et l'espace utilisateur
- Manipulation directe des paquets utilisant XDP

Structures de Données Sans Verrou:

- Les cartes eBPF utilisent des tables de hachage par CPU
- Opérations atomiques pour un accès concurrent
- Pas de surcoût de mutex/spinlock

Prêt pour le Déchargement Matériel:

- Le mode de déchargement XDP prend en charge l'exécution SmartNIC
- Compatible avec les cartes réseau prenant en charge XDP
- Retour aux modes natifs ou génériques du pilote

Fondation de la Technologie eBPF

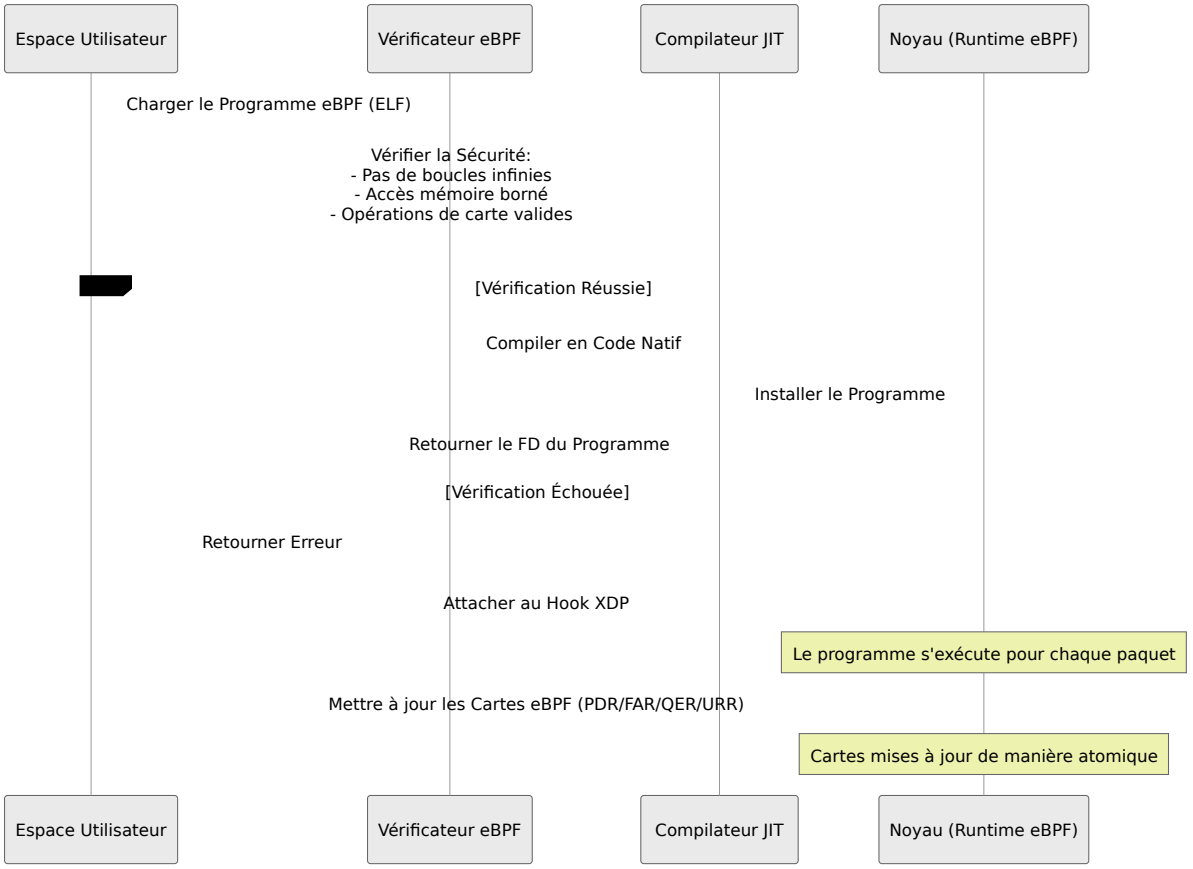
Qu'est-ce que eBPF ?

eBPF (extended Berkeley Packet Filter) est une technologie révolutionnaire du noyau Linux qui permet à des programmes sûrs et en bac à sable de s'exécuter dans l'espace noyau sans changer le code source du noyau ou charger des modules noyau.

Caractéristiques Clés:

- **Sécurité**: Le vérificateur eBPF garantit que les programmes ne peuvent pas faire planter le noyau
- **Performance**: S'exécute à la vitesse native du noyau (pas de surcoût d'interprétation)
- **Flexibilité**: Peut être mis à jour à l'exécution sans redémarrer le noyau
- **Observabilité**: Tracage et statistiques intégrés

Cycle de Vie du Programme eBPF



Cartes eBPF

Les cartes eBPF sont des structures de données noyau partagées entre les programmes eBPF et l'espace utilisateur.

Types de Cartes Utilisées dans OmniUPF:

Type de Carte	Description	Cas d'Utilisation
`BPF_MAP_TYPE_HASH`	Table de hachage avec paires clé-valeur	Recherche PDR par TEID ou IP UE
`BPF_MAP_TYPE_ARRAY`	Tableau indexé par entier	Recherche QER, FAR, URR par ID
`BPF_MAP_TYPE_PERCPU_HASH`	Table de hachage par CPU (sans verrou)	Recherches PDR haute performance
`BPF_MAP_TYPE_LRU_HASH`	Hachage LRU (Least Recently Used)	Éviction automatique des anciennes entrées

Opérations sur les Cartes:

- **Recherche**: O(1) recherche par hachage (sous-microseconde)
- **Mise à jour**: Mises à jour atomiques depuis l'espace utilisateur
- **Suppression**: Suppression immédiate des entrées
- **Itération**: Opérations par lots pour les dumps de cartes

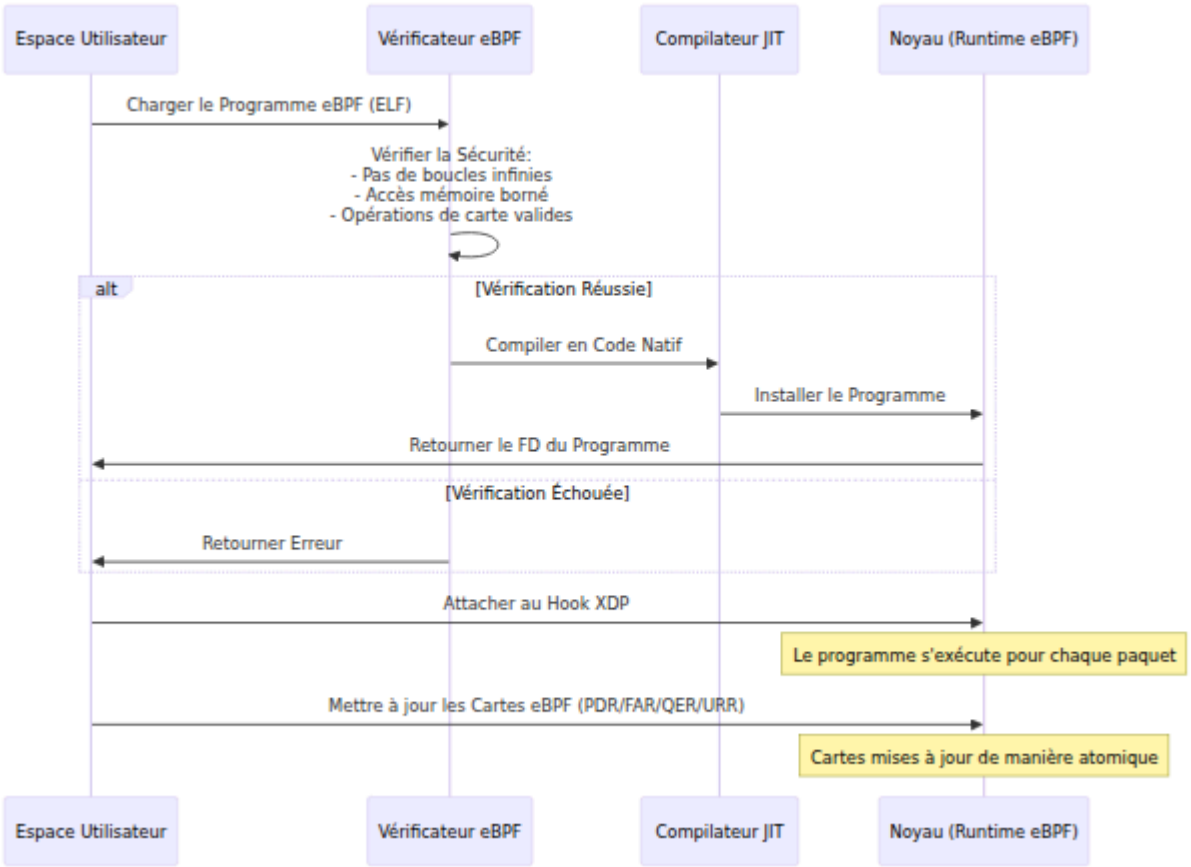
Chemin de Données XDP

Aperçu de XDP

XDP (eXpress Data Path) est un hook du noyau Linux qui permet aux programmes eBPF de traiter les paquets au point le plus précoce possible, juste après que le pilote réseau les ait reçus, avant la pile réseau du noyau.

Modes d'Attachement XDP

OmniUPF prend en charge trois modes d'attachement XDP, chacun avec des caractéristiques de performance et de compatibilité différentes.



1. Mode de Déchargement XDP

Exécution Matérielle (Meilleure Performance):

- Le programme eBPF s'exécute directement sur le matériel SmartNIC
- Traitement des paquets dans le NIC sans toucher au CPU
- Atteint un débit de 100 Gbps+
- Nécessite un SmartNIC compatible (Netronome, Mellanox ConnectX-6)

Configuration (dans `runtime.exs`):

```
xdp_attach_mode = "offload"
```

Limitations:

- Nécessite un matériel SmartNIC coûteux
- Complexité limitée des programmes eBPF
- Pas toutes les fonctionnalités eBPF prises en charge en matériel

---

2. Mode XDP Natif (Par Défaut pour la Production)

Exécution au Niveau du Pilote (Haute Performance):

- Le programme eBPF s'exécute dans le contexte du pilote réseau
- Paquets traités avant l'allocation de SKB (socket buffer)
- Atteint 10-40 Gbps par cœur
- Nécessite un pilote avec support XDP (la plupart des pilotes modernes)

Configuration (dans `runtime.exs`):

```
xdp_attach_mode = "native"
```

Avantages:

- Performance très élevée (multi-million pps)
- Large compatibilité matérielle
- Ensemble complet de fonctionnalités eBPF

Pilotes Supportés:

- Intel: i40e, ice, ixgbe, igb
- Mellanox: mlx4, mlx5
- Broadcom: bnxt
- Amazon: ena
- La plupart des cartes réseau 10G+

---

3. Mode XDP Générique

Émulation Logicielle (Compatibilité):

- Le programme eBPF s'exécute après que le noyau ait alloué SKB
- Émulation logicielle du comportement XDP
- Fonctionne sur n'importe quelle interface réseau
- Utile pour les tests et le développement

Configuration (dans `runtime.exs`):

```
xdp_attach_mode = "generic"
```

Cas d'Utilisation:

- Développement et tests
- Environnements virtualisés (VM sans SR-IOV)
- Matériel réseau plus ancien
- Tests d'interface de boucle

Performance: 1-5 Gbps (significativement plus lent que natif/déchargement)

Codes de Retour XDP

Les programmes eBPF retournent des codes d'action XDP pour indiquer au noyau quoi faire avec les paquets:

Code de Retour	Signification	Utilisation dans OmniUPF
`XDP_PASS`	Envoyer le paquet à la pile réseau du noyau	Mise en tampon (livraison locale), ICMP, trafic inconnu
`XDP_DROP`	Supprimer le paquet immédiatement	Paquets invalides, limitation de débit, suppressions de politique
`XDP_TX`	Transmettre le paquet à nouveau sur la même interface	Pas actuellement utilisé
`XDP_REDIRECT`	Envoyer le paquet à une interface différente	Chemin de transfert principal (N3 ↔ N6)
`XDP_ABORTED`	Erreur de traitement, supprimer le paquet et enregistrer	Erreurs de programme eBPF

Pipeline de Traitement des Paquets

Structure du Programme

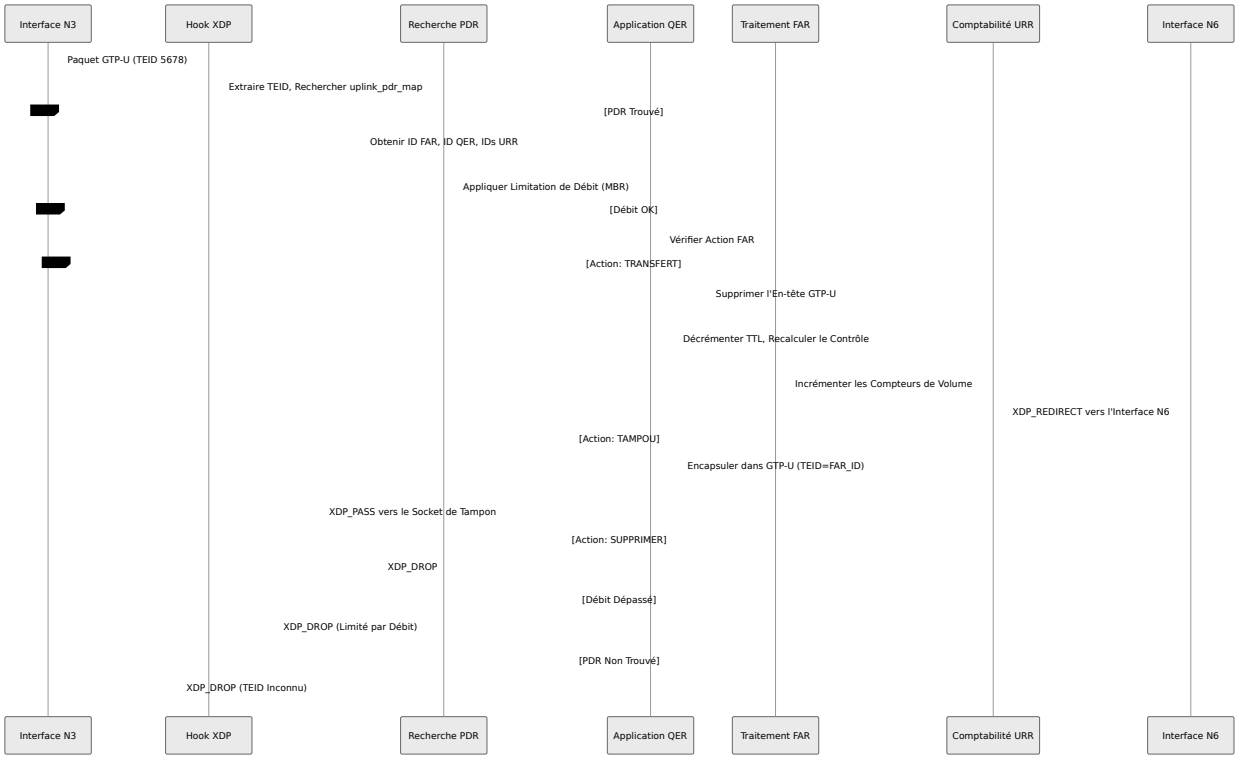
OmniUPF utilise des appels de queue eBPF pour créer un pipeline de traitement des paquets modulaire.

Appels de Queue:

- Permettent aux programmes eBPF d'appeler d'autres programmes eBPF
- Réutilise le même cadre de pile (profondeur de pile bornée)
- Permet un design de pipeline modulaire

- Profondeur maximale d'appel de queue de 33

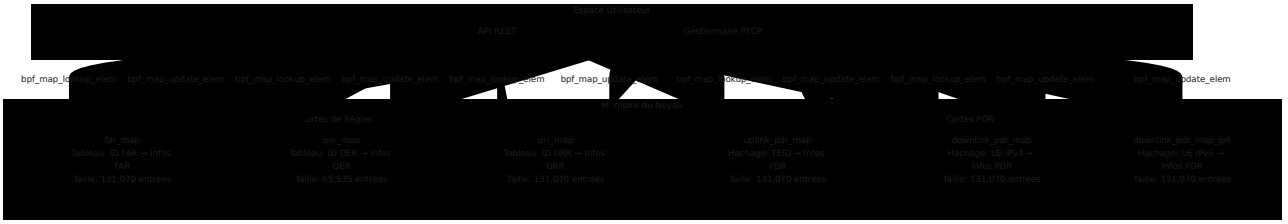
Traitement des Paquets Uplink



Traitement des Paquets Downlink

Architecture des Cartes eBPF

Disposition de la Mémoire des Cartes



Dimensionnement des Cartes

OmniUPF calcule automatiquement les tailles des cartes en fonction de la configuration `max_sessions`:

```
Cartes PDR = 2 × max_sessions  (uplink + downlink)
Cartes FAR = 2 × max_sessions  (uplink + downlink)
Cartes QER = 1 × max_sessions  (partagées par session)
Cartes URR = 3 × max_sessions  (plusieurs URR par session)
```

Exemple (max_sessions = 65,535):

- Cartes PDR: 131,070 entrées chacune
- Carte FAR: 131,070 entrées
- Carte QER: 65,535 entrées
- Carte URR: 131,070 entrées

Mémoire Totale:

```
Cartes PDR: 3 × 131,070 × 212 B = ~83 MB
Carte FAR:  131,070 × 20 B = ~2.6 MB
Carte QER:  65,535 × 36 B = ~2.3 MB
Carte URR:  131,070 × 20 B = ~2.6 MB
Total: ~91 MB de mémoire noyau
```

Mécanisme de Mise en Tampon

Aperçu de la Mise en Tampon

OmniUPF implémente la mise en tampon des paquets pour les scénarios de transfert en encapsulant les paquets dans GTP-U et en les envoyant à un processus en espace utilisateur via un socket UDP.

Architecture de Mise en Tampon

Parse error on line 4: ...outer l'En-tête UDP (port 22152)
3. ----------^ Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '-)', 'STADIUMEND', 'SUBROUTINEEND', 'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND', 'TRAPEND', 'INVTRAPEND', 'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'PS'

Réessayer

Détails de l'Encapsulation de Tampon

Lorsque la mise en tampon est activée (bit d'action FAR 2 défini), le programme eBPF:

1. **Calcule la Taille du Paquet Original**:

```
orig_packet_len = ntohs(ip->tot_len);  // À partir de l'en-tête IP
```

2. **Élargit l'En-tête du Paquet**:

```
// Ajouter de l'espace pour: IP Extérieure + UDP + GTP-U
gtp_encap_size = sizeof(struct iphdr) + sizeof(struct udphdr) + sizeof(struct gtpuhdr);
bpf_xdp_adjust_head(ctx, -gtp_encap_size);
```

3. **Construit l'En-tête IP Extérieur**:

```
ip->saddr = original_sender_ip;  // Préserver la source pour éviter le filtrage martien
ip->daddr = local_upf_ip;        // IP locale où le listener en espace utilisateur se lie
ip->protocol = IPPROTO_UDP;
ip->ttl = 64;
```

4. **Construit l'En-tête UDP**:

```
udp->source = htons(22152);  // BUFFER_UDP_PORT
udp->dest = htons(22152);
udp->len = htons(sizeof(udphdr) + sizeof(gtpuhdr) +
orig_packet_len);
```

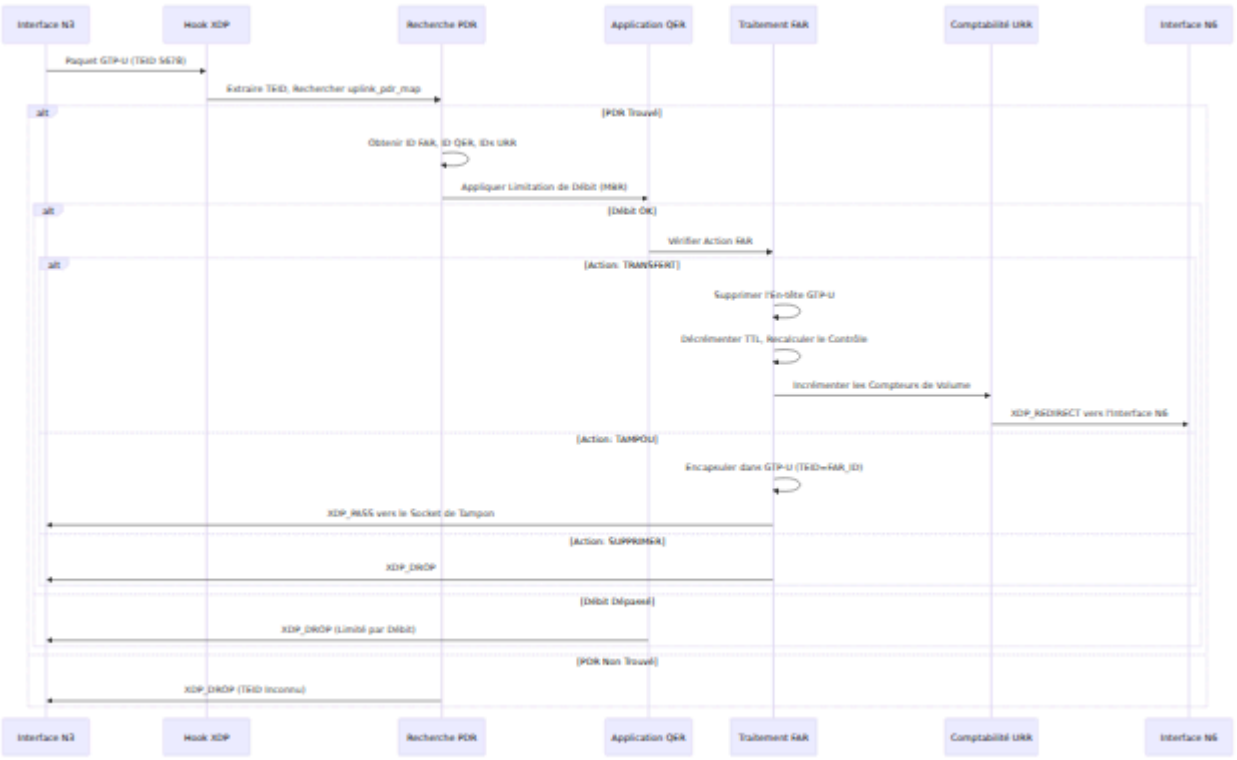
5. **Construit l'En-tête GTP-U**:

```
gtp->version = 1;
gtp->message_type = GTPU_G_PDU;
gtp->teid = htonl(far_id | (direction << 24));  // Encoder l'ID
FAR et la direction
gtp->message_length = htons(orig_packet_len);
```

6. **Retourne XDP_PASS**:
   - Le noyau livre le paquet au socket UDP local sur le port 22152
   - Le gestionnaire de tampon en espace utilisateur reçoit et stocke le paquet

Opération de Vidage de Tampon

Lorsque le transfert est terminé, le SMF met à jour le FAR pour effacer le drapeau BUFFER. Les paquets mis en tampon sont rejoués:



Paramètres de Gestion de Tampon

Paramètre	Par Défaut	Description
Max par FAR	10,000 paquets	Maximum de paquets mis en tampon par FAR
Max Total	100,000 paquets	Maximum total de paquets mis en tampon
TTL du Paquet	30 secondes	Temps avant que les paquets mis en tampon n'expirent
Port de Tampon	22152	Port UDP pour la livraison du tampon
Intervalle de Nettoyage du Tampon	60 secondes	À quelle fréquence vérifier les paquets expirés

Application de la QoS

Algorithme de Limitation de Débit

OmniUPF implémente un **limiteur de débit à fenêtre glissante** pour l'application de la QoS.

Parse error on line 5: ...= packet_size × 8 × (NSEC_PER_SEC / rate ----------^ Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '-)', 'STADIUMEND', 'SUBROUTINEEND', 'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND', 'TRAPEND', 'INVTRAPEND', 'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'PS'

Réessayer

Mise en Œuvre de la Fenêtre Glissante

Algorithme (depuis `qer.h`):

```
static __always_inline enum xdp_action limit_rate_sliding_window(
    const __u64 packet_size,
    volatile __u64 *window_start,
    const __u64 rate)
{
    static const __u64 NSEC_PER_SEC = 1000000000ULL;
    static const __u64 window_size = 5000000ULL;  // fenêtre de
5ms

    // Débit = 0 signifie illimité
    if (rate == 0)
        return XDP_PASS;

    // Calculer le temps de transmission pour ce paquet
    __u64 tx_time = packet_size * 8 * (NSEC_PER_SEC / rate);
    __u64 now = bpf_ktime_get_ns();

    // Vérifier si nous sommes en avance sur la fenêtre (le paquet
serait transmis dans le futur)
    __u64 start = *window_start;
    if (start + tx_time > now)
        return XDP_DROP;  // Limite de débit dépassée

    // Si la fenêtre est passée, la réinitialiser
    if (start + window_size < now) {
        *window_start = now - window_size + tx_time;
        return XDP_PASS;
    }

    // Mettre à jour la fenêtre pour tenir compte de ce paquet
    *window_start = start + tx_time;
    return XDP_PASS;
}
```

Paramètres Clés:

- **Taille de la Fenêtre**: 5ms (5,000,000 nanosecondes)
- **Par Direction**: Fenêtres séparées pour l'ascendant et le descendant
- **Mises à Jour Atomiques**: Utilise des pointeurs volatils pour un accès concurrent
- **MBR = 0**: Traité comme une bande passante illimitée

Exemple de Calcul de QoS

Scénario: MBR = 100 Mbps, Taille de Paquet = 1500 octets

1. **Temps de Transmission**:

```
tx_time = 1500 octets × 8 bits/octet × (1,000,000,000 ns/sec ÷
100,000,000 bps)
tx_time = 1500 × 8 × 10 = 120,000 ns = 120 µs
```

2. **Vérification de Débit**:

   - Si le dernier paquet transmis à `t=0`, le prochain paquet peut être transmis à `t=120µs`
   - Si le paquet arrive à `t=100µs`, il est supprimé (trop tôt)
   - Si le paquet arrive à `t=150µs`, il est transféré (fenêtre avancée)

3. **Taux Maximum de Paquets**:

```
Max PPS = (100 Mbps ÷ 8) ÷ 1500 octets = 8,333 paquets/seconde
Espace entre les paquets = 120 µs
```

Caractéristiques de Performance

Débit

Configuration	Débit	Paquets/Seconde	Latence
XDP Déchargement (SmartNIC)	100 Gbps	148 Mpps	< 1 µs
XDP Natif (NIC 10G, cœur unique)	10 Gbps	8 Mpps	2-5 µs
XDP Natif (NIC 10G, 4 cœurs)	40 Gbps	32 Mpps	2-5 µs
XDP Générique	1-5 Gbps	0.8-4 Mpps	50-100 µs

Répartition de Latence

Latence Totale de Traitement des Paquets (XDP Natif):

Étape	Latence	Cumulative
RX NIC	0.5 µs	0.5 µs
Invocation du Hook XDP	0.1 µs	0.6 µs
Recherche PDR (Hachage)	0.3 µs	0.9 µs
Vérification de Débit QER	0.1 µs	1.0 µs
Traitement FAR	0.5 µs	1.5 µs
Mise à Jour URR	0.2 µs	1.7 µs
Encapsulation/Décapsulation GTP-U	0.8 µs	2.5 µs
XDP_REDIRECT	0.5 µs	3.0 µs
TX NIC	0.5 µs	3.5 µs

Total: ~3.5 µs par paquet (XDP Natif, NIC 10G)

Utilisation du CPU

Capacité de Traitement par Cœur:

- Cœur unique: 8-10 Mpps (XDP Natif)
- Avec hyper-threading: 12-15 Mpps
- Scalabilité multi-cœur: presque linéaire jusqu'à 8 cœurs

Utilisation du CPU par Taux de Paquets:

```
CPU % ≈ (Taux de Paquets / 10,000,000) × 100% par cœur
```

Exemple: Un trafic de 2 Mpps utilise ~20% d'un cœur

Bande Passante Mémoire

Accès aux Cartes eBPF:

- Recherche par hachage: ~100 ns (hit de cache)
- Recherche par hachage: ~300 ns (miss de cache)
- Recherche par tableau: ~50 ns (toujours hit de cache)

Bande Passante Mémoire Requise:

```
Bande Passante = Taux de Paquets × (Taille Moyenne du Paquet +
Recherches de Carte × 64 octets)
```

Exemple: 10 Mpps × (1500 B + 3 recherches × 64 B) ≈ 160 Gbps de bande passante mémoire

Scalabilité et Réglage

Scalabilité Horizontale

Instances UPF Multiples:

Setting SMF as parent of SMF would create a cycle

Réessayer

Distribution des Sessions:

- Le SMF distribue les sessions entre les instances UPF
- Chaque UPF gère un sous-ensemble de sessions UE
- Pas de communication inter-UPF nécessaire (sans état)

Scalabilité Verticale

Réglage du CPU:

1. Activer l'affinité CPU pour le traitement XDP

2. Utiliser RSS (Receive Side Scaling) pour distribuer les files d'attente RX
3. Fixer les programmes eBPF à des cœurs spécifiques

Réglage du NIC:

1. Augmenter la taille du tampon RX
2. Activer les NIC multi-queues (RSS)
3. Utiliser un directeur de flux pour le guidage du trafic

Réglage du Noyau:

```
# Augmenter la limite de mémoire verrouillée pour les cartes eBPF
ulimit -l unlimited

# Désactiver l'équilibre des IRQ pour les cœurs XDP
systemctl stop irqbalance

# Définir le gouverneur CPU sur performance
cpupower frequency-set -g performance

# Augmenter les tailles de tampon réseau
sysctl -w net.core.rmem_max=134217728
sysctl -w net.core.wmem_max=134217728
```

Planification de Capacité

Formule:

```
Cœurs CPU Requis = (PPS Attendu ÷ 10,000,000) × 1.5  (50% de
marge)
Mémoire Requise = (Sessions Max × 212 B × 3) + 100 MB (cartes eBPF
+ overhead)
Réseau Requis = (Débit de Pointe × 2) + 10 Gbps (marge)
```

Exemple (1 million de sessions, 20 Gbps de pointe):

- CPU: (20 Gbps ÷ 10 Gbps par cœur) × 1.5 = 3-4 cœurs
- Mémoire: (1M × 212 B × 3) + 100 MB ≈ 750 MB

- Réseau: (20 Gbps × 2) + 10 Gbps = 50 Gbps d'interfaces

Documentation Connexe

- **Guide d'Opérations UPF** - Opérations générales UPF et déploiement
- **Guide de Gestion des Règles** - Détails PDR, FAR, QER, URR
- **Guide de Surveillance** - Surveillance des performances et métriques
- **Guide d'Opérations de l'Interface Web** - Utilisation du panneau de contrôle
- **Guide de Dépannage** - Problèmes courants et diagnostics
- **Guide de Jardin Clos** - Redirection hors crédit utilisant le mécanisme BUFFER pour l'interception de paquets en espace utilisateur

Guide de Configuration OmniUPF

Table des Matières

1. Aperçu
2. Modes de Fonctionnement
3. Modes de Rattachement XDP
4. Paramètres de Configuration
5. Fichier de Configuration
6. Compatibilité Hyperviseur
7. Compatibilité NIC
8. Exemples de Configuration
9. Dimensionnement de la Carte et Planification de Capacité

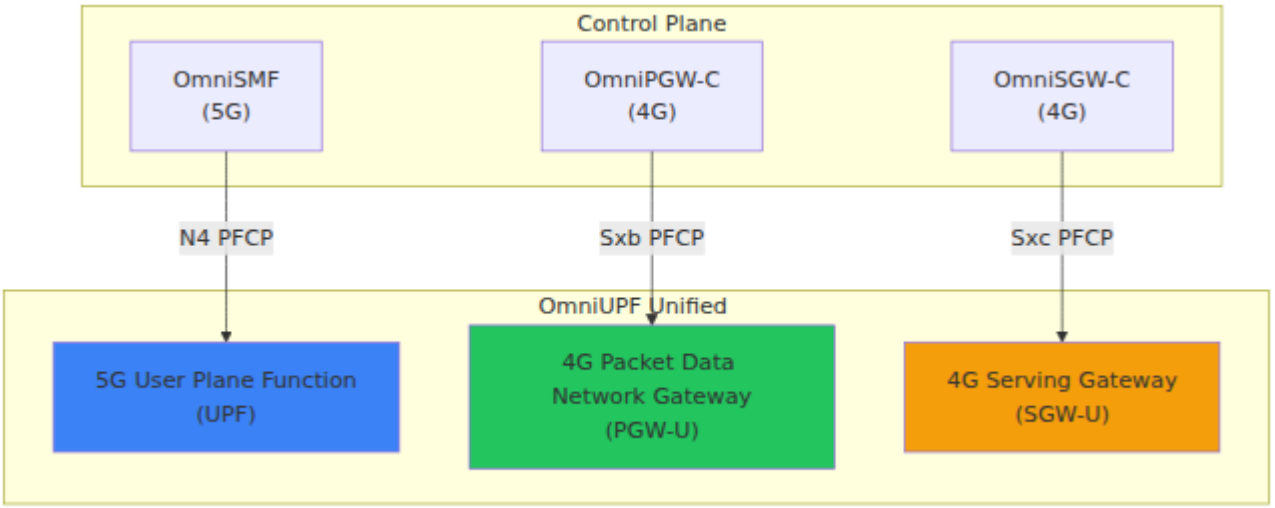
Aperçu

OmniUPF est une fonction de plan utilisateur polyvalente qui peut fonctionner en plusieurs modes pour prendre en charge à la fois les réseaux de cœur 4G (EPC) et 5G. Le plan de contrôle est implémenté en Elixir/OTP, et la configuration est gérée via un fichier de configuration Elixir (`runtime.exs`).

Lors de l'installation du package, le fichier de configuration est placé à `/etc/omniupf/runtime.exs` et préservé lors des mises à jour. Le UPF doit être redémarré après des modifications de configuration.

Modes de Fonctionnement

OmniUPF est une **plateforme unifiée** qui peut fonctionner simultanément comme :



Configuration du Mode

Le mode de fonctionnement est **déterminé par le plan de contrôle** (SMF, PGW-C ou SGW-C) qui établit des associations PFCP avec OmniUPF. Aucune configuration spécifique d'OmniUPF n'est requise pour passer d'un mode à l'autre.

Fonctionnement Simultané :

- OmniUPF peut accepter des associations PFCP de plusieurs plans de contrôle simultanément
- Une seule instance d'OmniUPF peut agir comme UPF, PGW-U et SGW-U **en même temps**
- Les sessions provenant de différents plans de contrôle sont isolées et gérées indépendamment

---

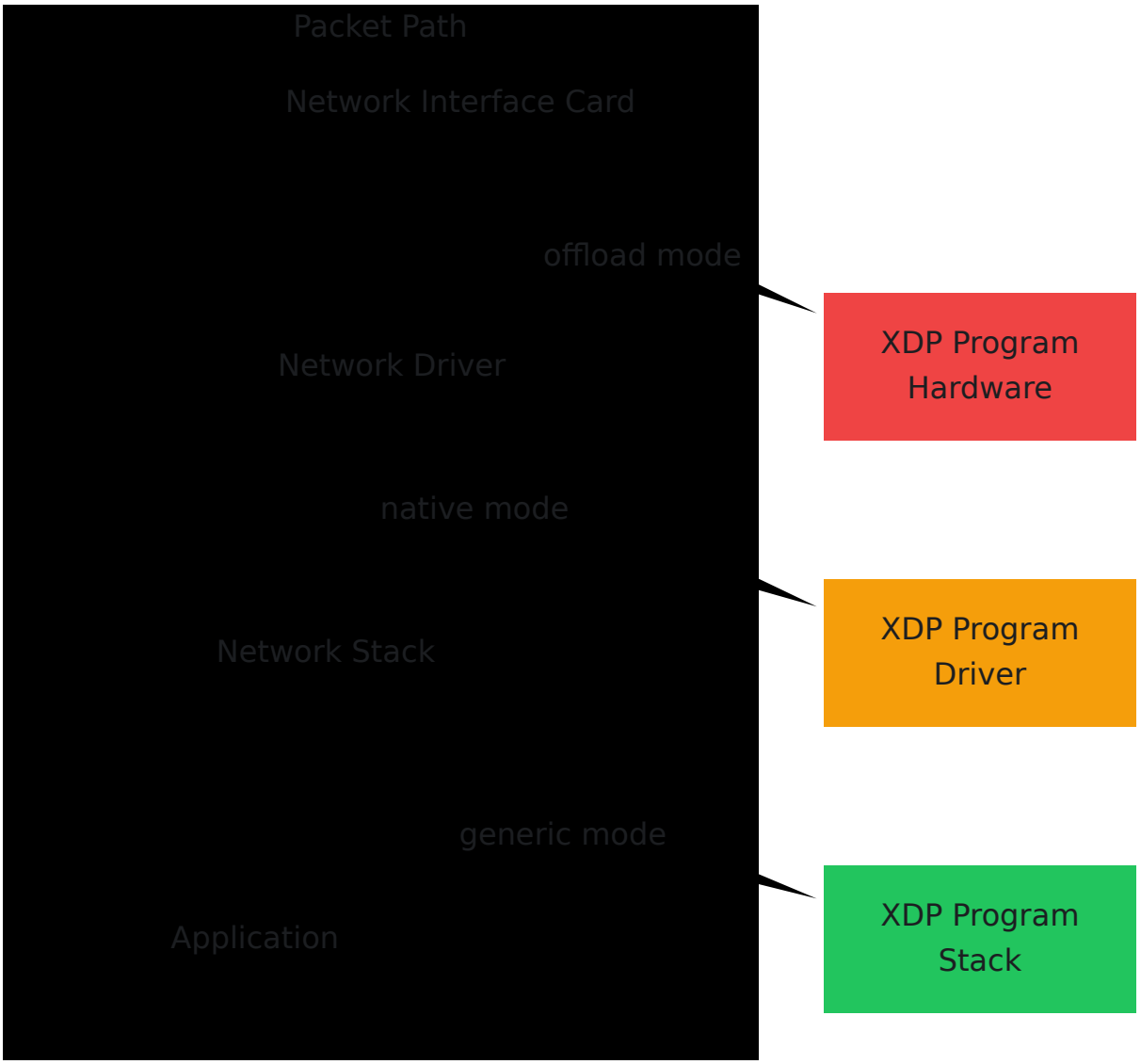
Modes de Rattachement XDP

OmniUPF utilise XDP (eXpress Data Path) pour le traitement de paquets à haute performance. Trois modes de rattachement sont pris en charge.

Pour des instructions détaillées sur la configuration de XDP, en particulier pour Proxmox et d'autres hyperviseurs, consultez le Guide des Modes XDP.

Comparaison des Modes

Mode	Point de Rattachement	Performance	Cas d'Utilisation	Exige NI
Générique	Pile réseau (noyau)	~1-2 Mpps	Test, développement, compatibilité	Tout NIC
Natif	Pilote réseau (noyau)	~5-10 Mpps	Production (bare metal, VM avec SR-IOV)	Pilote compati XDP
Offload	Matériel NIC (SmartNIC)	~10-40 Mpps	Production à haut débit	SmartNI avec décharg XDP



Mode Générique (Par Défaut)

Description: Le programme XDP s'exécute dans la pile réseau du noyau

Avantages:

- Fonctionne avec **tout** interface réseau
- Pas d'exigences spéciales de pilote ou de matériel
- Idéal pour les tests et le développement
- Compatible avec tous les hyperviseurs et plateformes de virtualisation

Inconvénients:

- Performance inférieure (~1-2 Mpps par cœur)

- Les paquets sont déjà passés par le pilote avant le traitement XDP

Configuration (dans `runtime.exs`):

```
xdp_attach_mode = "generic"
```

Meilleur pour:

- Machines virtuelles sans SR-IOV
- Environnements de test et de validation
- NIC sans support de pilote XDP
- Hyperviseurs comme Proxmox, VMware, VirtualBox

Mode Natif (Recommandé)

Description: Le programme XDP s'exécute au niveau du pilote réseau

Avantages:

- Haute performance (~5-10 Mpps par cœur)
- Paquets traités avant d'entrer dans la pile réseau
- Latence significativement inférieure à celle du mode générique
- Fonctionne sur bare metal et VM SR-IOV

Inconvénients:

- Nécessite un pilote réseau avec support XDP
- Tous les NIC/pilotes ne prennent pas en charge XDP natif

Configuration (dans `runtime.exs`):

```
xdp_attach_mode = "native"
```

Meilleur pour:

- Déploiements de production sur bare metal

- VMs avec passage SR-IOV
- NIC avec pilotes compatibles XDP (Intel, Mellanox, etc.)

Exigences:

- Pilote réseau compatible XDP (voir Compatibilité NIC)
- Noyau Linux 5.15+ avec support XDP activé

Mode Offload (Performance Maximale)

Description: Le programme XDP s'exécute directement sur le matériel SmartNIC

Avantages:

- Performance maximale (~10-40 Mpps)
- Aucun surcoût CPU pour le traitement des paquets
- Latence sub-microseconde
- Libère le CPU pour le traitement du plan de contrôle

Inconvénients:

- Nécessite un matériel SmartNIC coûteux
- Disponibilité limitée des SmartNIC
- Configuration et mise en place complexes

Configuration (dans `runtime.exs`):

```
xdp_attach_mode = "offload"
```

Meilleur pour:

- Déploiements de production à ultra-haut débit
- Informatique en périphérie avec des exigences strictes en matière de latence
- Environnements où les ressources CPU sont limitées

Exigences:

- SmartNIC avec support de déchargement XDP (Netronome Agilio CX, Mellanox BlueField)
- Firmware et pilotes spécialisés

Paramètres de Configuration

Interfaces Réseau

Paramètre	Description	Type	Par Défaut
`xdp_interfaces`	Interfaces réseau pour le trafic N3/N6/N9 (séparées par des virgules, ou `"auto"` pour toutes les non-boucle)	String	`"auto"`
`gtpu_interface`	NIC portant tout GTP-U (N3/N9/S5-S8). Définir avec `sgi_interface` pour attacher un programme XDP uniquement GTP-U à cette NIC ; laisser non défini pour le programme combiné	String	`nil`
`sgi_interface`	NIC portant SGi/N6 en IP simple. Définir avec `gtpu_interface` pour attacher un programme XDP uniquement descendant à cette NIC	String	`nil`
`n3_address`	Adresse IPv4 pour l'interface N3 (GTP-U depuis RAN)	IP	`"127.0.0.1"`
`n9_address`	Adresse IPv4 pour l'interface N9 (UPF vers UPF pour ULCL)	IP	Identique à `n3_address`

Exemple (dans `runtime.exs`):

```
xdp_interfaces = "eth0,eth1"
n3_address = "10.100.50.233"
n9_address = "10.100.50.234"
```

Séparation du chemin de données par NIC. Lorsque GTP-U et SGi passent par des NIC séparées, nommez chacune afin que le chargeur attache un programme XDP à usage unique par NIC (uniquement GTP-U sur `gtpu_interface`, uniquement descendant sur `sgi_interface`) au lieu du programme combiné "on-a-stick". Les deux doivent également apparaître dans `xdp_interfaces`. Laissez les deux non définis (la valeur par défaut) pour attacher le programme combiné à chaque interface.

```
xdp_interfaces = "eth1,eth2"
gtpu_interface = "eth1"   # tout GTP-U (N3/N9/S5-S8)
sgi_interface  = "eth2"   # SGi/N6 en IP simple
```

Configuration PFCP

Paramètre	Description	Type	Par Défaut
`pfcp_address`	Adresse locale pour le serveur PFCP (interface N4/Sxb/Sxc)	IP String	`"127.0.0.1"`
`pfcp_port`	Port PFCP	Integer	`8805`
`node_id`	ID de nœud local pour le protocole PFCP	IP String	`"127.0.0.1"`
`heartbeat_interval_ms`	Intervalle de heartbeat PFCP (millisecondes)	Integer	`5000`
`heartbeat_timeout_ms`	Délai d'attente de heartbeat PFCP (millisecondes)	Integer	`5000`
`heartbeat_retries`	Nombre de tentatives de heartbeat avant de déclarer le pair mort	Integer	`3`
`association_setup_timeout`	Délai d'attente pour une réponse de configuration d'association PFCP lorsque le UPF initie une association (millisecondes)	Integer	`5000`

Paramètre	Description	Type	Par Défaut
`pfcp_remote_nodes`	Liste des chaînes d'adresses IP du plan de contrôle avec lesquelles le UPF établit activement des associations au démarrage (initié par le UPF). Vide signifie que le UPF n'accepte que les associations initiées par le plan de contrôle.	Liste de String	`[]`

Exemple (dans `runtime.exs`):

```
pfcp_address = "10.100.50.241"
pfcp_port = 8805
node_id = "10.100.50.241"
heartbeat_interval_ms = 10_000
heartbeat_retries = 5
association_setup_timeout = 5_000
pfcp_remote_nodes = []                # ou par exemple
["10.100.50.20", "10.100.50.22"]
```

API et Surveillance

Paramètre	Description	Type	Par Défaut
`api_port`	Port local pour le serveur API REST (Phoenix)	Integer	`8080`
`log_level`	Niveau de journalisation (`:debug`, `:info`, `:warning`, `:error`)	Atom	`:info`

Exemple (dans `runtime.exs`):

```
api_port = 8080
log_level = :debug
```

Gestion du Chemin GTP

Paramètre	Description	Type	Par Défaut
`gtpu_port`	Port GTP-U	Integer	`2152`
`gtp_echo_interval`	Intervalle d'écho par défaut en secondes	Integer	`10`
`gtp_echo_retries`	Max d'échos manqués avant échec de chemin	Integer	`3`
`gtp_peers`	Liste des pairs GTP pour les keepalives de requête d'écho	Liste	`[]`

Exemple (dans `runtime.exs`):

```
gtp_echo_interval = 15
gtp_echo_retries = 3
gtp_peers = [
  %{address: parse_ip.("10.100.50.50"), echo: true, echo_interval:
10},
  %{address: parse_ip.("10.100.50.60"), echo: false},
]
```

GTP-U de Taille Excessive Descendante

GTP-U ajoute 36 octets à chaque paquet descendant. Un paquet de taille complète de 1500 octets provenant d'Internet devient donc un paquet GTP-U de 1536 octets, qui ne s'adapte pas à un lien N3 de 1500 octets. L'expéditeur GTP-U est tenu de fragmenter de tels paquets (TS 29.281 §4.2.4), et de nombreux UE et routeurs ignorent le MTU de lien proposé dans le PCO, donc cela se produit lors de l'opération normale.

Le chemin rapide XDP ne peut pas fragmenter. Lorsqu'un paquet GTP-U descendant est plus grand que le MTU de sortie, il est remis au chemin lent de l'espace utilisateur, qui l'envoie depuis un socket UDP ordinaire afin que le noyau fragmente le paquet extérieur. Cela s'applique à la fois lorsque le UPF encapsule (N6 à N3) et lorsqu'il relaie entre des tunnels GTP-U (SGW-U).

Paramètre	Description	Type	Par Défaut
`gtpu_oversize_punt_enabled`	`true`: le GTP-U descendant de taille excessive est envoyé à l'espace utilisateur et fragmenté. `false`: il est abandonné	Boolean	aucun, **requis**

Le paramètre est requis : le UPF ne démarrera pas sans lui. Chaque paquet abandonné quitte le chemin rapide, donc un abonné envoyant un trafic soutenu de taille complète à travers un MTU de tunnel plus petit (par exemple un

téléchargement en masse vers un appareil qui ignore le MTU PCO) déplace cette charge dans l'espace utilisateur. Surveillez `upf_gtpu_oversize_punted_total` (voir METRICS.md) et définissez le paramètre sur `false` pour réduire la charge, au prix d'abandonner ces paquets.

Exemple (dans `runtime.exs`):

```
gtpu_oversize_punt_enabled = true
```

Les GTP-U fragmentés reçus du RAN sont réassemblés par le noyau et réinjectés dans le chemin de données. Cela ne nécessite aucune configuration.

Limitation de MSS TCP

Un appareil qui ignore le MTU de lien dans le PCO (ou l'annonce de routeur) annonce un MSS de 1460 octets dans son handshake TCP. Les deux extrémités envoient alors des paquets de 1500 octets, et chacun d'eux passe par le chemin lent de taille excessive décrit ci-dessus. La limitation du MSS dans le handshake garde ce trafic sur le chemin rapide.

Paramètre	Description	Type	Par Défaut
`tcp_mss_clamp`	Plus grand MSS TCP (IPv4) qu'un SYN ou SYN-ACK peut annoncer à travers ce UPF. Les valeurs plus grandes sont réduites à cela. IPv6 utilise une valeur 20 inférieure. `false` laisse le MSS intact	`false` ou entier 536-1460	aucun, **requis**

Seuls les paquets SYN et SYN-ACK sont modifiés, un ou deux par connexion ; les paquets de données ne sont pas touchés. Les deux directions sont limitées : le SYN de l'UE limite ce que le serveur envoie à l'UE, et le SYN-ACK du serveur limite ce que l'UE envoie. Le checksum TCP est mis à jour de manière incrémentale. La limitation s'applique là où le UPF décapsule ou encapsule le paquet utilisateur (PGW-U / UPF), pas sur un relais pur SGW-U.

Définissez-le sur le MTU de lien IPv4 proposé dans le PCO moins 40 octets : `1310` pour un MTU de 1350.

Exemple (dans `runtime.exs`):

```
tcp_mss_clamp = 1310
```

La limitation ne couvre que TCP. Le trafic UDP et tout autre trafic plus grand que le MTU du tunnel prend toujours le chemin lent de taille excessive, et donc TCP dont le handshake le UPF n'a pas vu.

Capacité de la Carte eBPF

Paramètre	Description	Type	Par Défaut
`max_sessions`	Nombre maximum de sessions concurrentes	Integer	`1_000_000`
`far_map_size`	Taille de la carte eBPF FAR	Integer	`131_070`
`qer_map_size`	Taille de la carte eBPF QER	Integer	`65_535`
`urr_map_size`	Taille de la carte eBPF URR	Integer	`65_535`

Exemple (dans `runtime.exs`):

```
max_sessions = 100_000
far_map_size = 131_070
qer_map_size = 65_535
urr_map_size = 131_070
```

Configuration du Tampon

Les paquets descendants sont mis en mémoire tampon pendant qu'un UE est inactif, puis vidés lorsque l'UE est paginé. Ces paramètres définissent combien est conservé et pendant combien de temps.

Paramètre	Description	Type	Par Défaut
`buffer_port`	Port UDP pour les paquets mis en mémoire tampon depuis eBPF	Integer	`22152`
`buffer_max_total`	Nombre maximum de paquets mis en mémoire tampon à travers tous les FARs (plafond global)	Integer	`10_000`
`buffer_max_per_far`	Nombre maximum de paquets mis en mémoire tampon par FAR	Integer	`100`
`buffer_ttl_ms`	Temps pendant lequel un paquet mis en mémoire tampon est conservé avant d'être abandonné (millisecondes)	Integer	`30_000`
`buffer_cleanup_interval_ms`	Intervalle auquel les paquets mis en mémoire tampon expirés sont balayés (millisecondes)	Integer	`5_000`

Exemple (dans `runtime.exs`):

```
buffer_port = 22152
buffer_max_total = 10_000
buffer_max_per_far = 100
buffer_ttl_ms = 30_000
buffer_cleanup_interval_ms = 5_000
```

Les abandons causés par ces limites sont comptés par `upf_buffer_packets_dropped_total` (étiquettes `global_limit`, `far_limit`, `expired`). Voir la Référence des Métriques.

Drapeaux de Fonctionnalité

Paramètre	Description	Type	Par Défaut
`feature_ueip`	Activer l'allocation d'IP UE par OmniUPF	Boolean	`false`
`ueip_pool`	Pool d'IP pour l'allocation d'IP UE (nécessite `feature_ueip`)	CIDR String	`"10.60.0.0/24"`
`feature_ftup`	Activer l'allocation de F-TEID par OmniUPF	Boolean	`true`
`teid_pool_start`	Début de la plage d'allocation de TEID	Integer	`1`
`teid_pool_end`	Fin de la plage d'allocation de TEID	Integer	`10_000_000`

Exemple (allocation d'IP UE) (dans `runtime.exs`):

```
feature_ueip = true
ueip_pool = "10.45.0.0/16"
```

Exemple (allocation de F-TEID) (dans `runtime.exs`):

```
feature_ftup = true
teid_pool_start = 1
teid_pool_end = 1_000_000
```

Configuration du Gestionnaire de Routes

Pour la synchronisation des routes UE avec le démon FRR (Free Range Routing). Voir le Guide de Gestion des Routes pour plus de détails.

Paramètre	Description	Type	Pa
`route_manager_enabled`	Activer la synchronisation automatique des routes UE	Boolean	`true`
`route_manager_type`	Type de démon de routage (`"frr"` ou `"static"`)	String	`"frr`
`route_manager_nexthop`	Adresse du prochain saut pour les routes UE injectées (vide = directement connecté)	String	`""`
`route_manager_vtysh_path`	Chemin vers le binaire FRR `vtysh` (utilisé lorsque `route_manager_type` est `"frr"`)	String	`"/us`
`route_manager_sync_interval_ms`	Intervalle de réconciliation complet entre les routes suivies et le démon de routage (millisecondes)	Integer	`600_`
`route_batch_interval_ms`	Fenêtre de débounce pour le regroupement des opérations d'ajout/suppression de routes	Integer	`1_00`

Paramètre	Description	Type	Pa
	incrémentales (millisecondes)		

Exemple (dans `runtime.exs`):

```
route_manager_enabled = true
route_manager_type = "frr"
route_manager_nexthop = ""
route_manager_vtysh_path = "/usr/bin/vtysh"
route_manager_sync_interval_ms = 600_000
route_batch_interval_ms = 1_000
```

Quand Activer:

- Déploiements Multi-UPF nécessitant une annonce de route
- Intégration avec les protocoles de routage OSPF ou BGP
- Nécessite le démon FRRouting installé et configuré

Configuration eBPF / XDP

Paramètre	Description	Type	Par Défaut
`xdp_interfaces`	Liste séparée par des virgules des interfaces pour XDP, ou `"auto"` pour toutes les non-boucle	String	`"auto"`
`xdp_attach_mode`	Mode XDP : `"generic"`, `"native"` ou `"offload"`	String	`"generic"`
`ebpf_pin_path`	Chemin de pin du système de fichiers BPF	String	`"/sys/fs/bpf/upf_pipeline"`
`xdp_obj_path`	Chemin vers l'objet eBPF compilé	String	`"/etc/omniupf/ipentrypoint_bp`
`ebpf_backend`	Backend de carte de plan de données : `:nif` (NIF direct, par défaut), `:port`	Atom	`:nif`

Paramètre	Description	Type	Par Défaut
	(processus de port externe), ou `:mock` (en mémoire, pour des tests sans eBPF)		

Exemple (dans `runtime.exs`):

```
xdp_interfaces = "auto"              # ou "eth0,eth1" pour des
interfaces spécifiques
xdp_attach_mode = "native"
ebpf_pin_path = "/sys/fs/bpf/upf_pipeline"
xdp_obj_path = "/etc/omniupf/ipentrypoint_bpf.o"
ebpf_backend = :nif
```

Fichier de Configuration

Fichier de Configuration Elixir (runtime.exs)

Fichier: `/etc/omniupf/runtime.exs`

Le fichier de configuration utilise la syntaxe Elixir avec des affectations de variables simples pour la lisibilité. Les variables sont appliquées à la configuration de l'application en bas du fichier via `config :upf_ex, ...`.

```
import Config

parse_ip = fn str ->
  {:ok, addr} = :inet.parse_address(String.to_charlist(str))
  addr
end

#
==========================================================================
# Configuration PFCP (N4/Sx)
#
==========================================================================
pfcp_address = "10.100.50.241"       # Adresse IP pour écouter les me
PFCP
pfcp_port = 8805                      # Port PFCP (standard : 8805)
node_id = "10.100.50.241"            # ID de nœud annoncé dans la
configuration d'association

#
==========================================================================
# Configuration du Plan de Données GTP-U
#
==========================================================================
n3_address = "10.100.50.233"         # IP de l'interface N3 (GTP-U ve
n9_address = n3_address               # IP de l'interface N9 (par déf
gtpu_port = 2152                      # Port GTP-U (standard : 2152)
buffer_port = 22152                   # Port d'écoute du tampon

#
==========================================================================
# Configuration eBPF / XDP
#
==========================================================================
xdp_interfaces = "auto"              # "auto" pour toutes les non-bou
liste séparée par des virgules
xdp_attach_mode = "native"           # Mode XDP : "generic", "native"
"offload"
ebpf_pin_path = "/sys/fs/bpf/upf_pipeline"
xdp_obj_path = "/etc/omniupf/ipentrypoint_bpf.o"

#
==========================================================================
# Configuration des Sessions & Pool de Ressources
```

```
#
==========================================================================
max_sessions = 100_000
teid_pool_start = 1
teid_pool_end = 10_000_000
far_map_size = 131_070
qer_map_size = 65_535
urr_map_size = 65_535

#
==========================================================================
# Drapeaux de Fonctionnalité
#
==========================================================================
feature_ueip = true
feature_ftup = true
ueip_pool = "10.45.0.0/16"

#
==========================================================================
# Gestion des Routes
#
==========================================================================
route_manager_enabled = true
route_manager_type = "frr"

#
==========================================================================
# Configuration de Heartbeat
#
==========================================================================
heartbeat_interval_ms = 10_000
heartbeat_timeout_ms = 5_000
heartbeat_retries = 5

#
==========================================================================
# Surveillance des Échos des Pairs GTP-U
#
==========================================================================
gtp_echo_interval = 10
gtp_echo_retries = 3
gtp_peers = [
  %{address: parse_ip.("10.100.50.50"), echo: true, echo_interval: 10
```

```
]

#
==========================================================================
# API HTTP et Journalisation
#
==========================================================================
api_port = 8080
log_level = :info

#
==========================================================================
# Appliquer la Configuration (ne pas éditer en dessous de cette ligne
#
==========================================================================
config :upf_ex,
  pfcp_address: parse_ip.(pfcp_address),
  pfcp_port: pfcp_port,
  n3_address: parse_ip.(n3_address),
  n9_address: parse_ip.(n9_address),
  node_id: parse_ip.(node_id),
  api_port: api_port,
  ebpf_pin_path: ebpf_pin_path,
  xdp_obj_path: xdp_obj_path,
  interface_names: String.split(xdp_interfaces, ","),
  xdp_attach_mode: xdp_attach_mode,
  feature_ueip: feature_ueip,
  feature_ftup: feature_ftup,
  ueip_pool: ueip_pool,
  teid_pool_start: teid_pool_start,
  teid_pool_end: teid_pool_end,
  far_map_size: far_map_size,
  qer_map_size: qer_map_size,
  urr_map_size: urr_map_size,
  max_sessions: max_sessions,
  route_manager_enabled: route_manager_enabled,
  route_manager_type: route_manager_type,
  buffer_port: buffer_port,
  gtpu_port: gtpu_port,
  heartbeat_interval_ms: heartbeat_interval_ms,
  heartbeat_timeout_ms: heartbeat_timeout_ms,
  heartbeat_retries: heartbeat_retries,
  gtp_echo_interval: gtp_echo_interval,
  gtp_echo_retries: gtp_echo_retries,
```

```
  gtp_peers: gtp_peers

config :logger, level: log_level
```

Gestion du Service

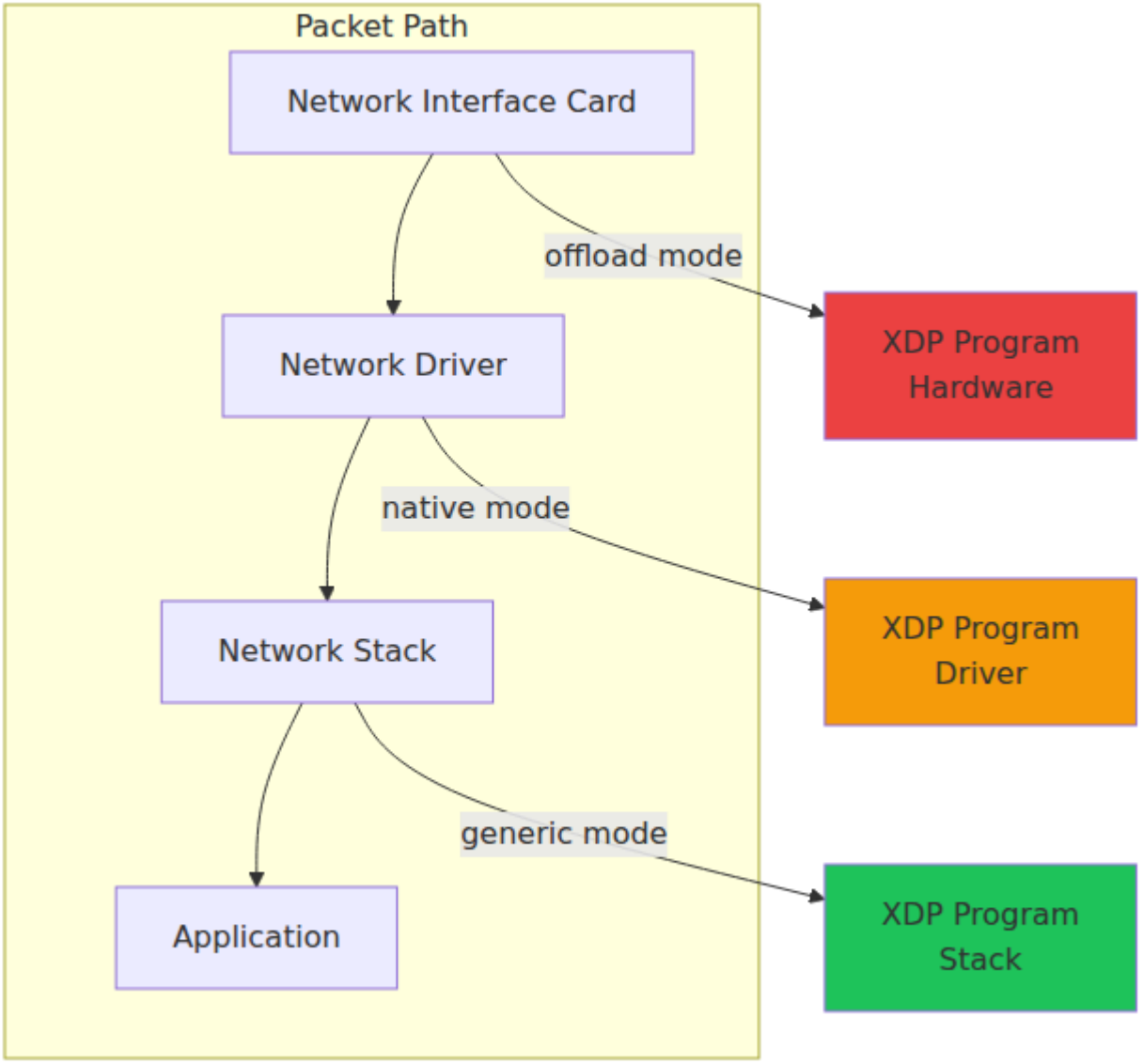
```
# Démarrer le UPF
sudo systemctl start omniupf

# Arrêter le UPF
sudo systemctl stop omniupf

# Redémarrer après des modifications de configuration
sudo systemctl restart omniupf

# Vérifier l'état
sudo systemctl status omniupf

# Voir les journaux
sudo journalctl -u omniupf -f

# Ou utilisez directement le binaire de version
sudo /opt/omniupf/bin/upf_ex start
sudo /opt/omniupf/bin/upf_ex stop
sudo /opt/omniupf/bin/upf_ex daemon   # démarrer en tant que démon
en arrière-plan
```

Compatibilité avec les Hyperviseurs

Vue d'ensemble

OmniUPF est compatible avec tous les principaux hyperviseurs et plateformes de virtualisation. Le mode d'attachement XDP et la configuration réseau dépendent des capacités réseau de l'hyperviseur.

Pour des instructions étape par étape sur l'activation de XDP natif sur Proxmox et d'autres hyperviseurs, voir le Guide des Modes XDP.



Proxmox VE

Configurations Supportées :

1. Mode Pont (XDP Générique)

Cas d'utilisation : Réseau standard de VM

Configuration :

- Dispositif Réseau : VirtIO ou E1000

- Mode XDP : `generic`
- Performance : ~1-2 Mpps

Paramètres de VM Proxmox :

```
Dispositif Réseau : net0
Modèle : VirtIO (paravirtualisé)
Pont : vmbr0
```

Configuration OmniUPF (dans `runtime.exs`) :

```
xdp_interfaces = "eth0"
xdp_attach_mode = "generic"
```

---

2. Passerelle SR-IOV (XDP Natif)

Cas d'utilisation : Production haute performance

Configuration :

- Dispositif Réseau : Fonction Virtuelle SR-IOV
- Mode XDP : `native`
- Performance : ~5-10 Mpps

Exigences :

- NIC physique avec support SR-IOV (Intel X710, Mellanox ConnectX-5)
- SR-IOV activé dans le BIOS
- IOMMU activé (`intel_iommu=on` ou `amd_iommu=on` dans GRUB)

Activer SR-IOV sur Proxmox :

```
# Modifier la configuration GRUB
nano /etc/default/grub

# Ajouter à GRUB_CMDLINE_LINUX_DEFAULT :
intel_iommu=on iommu=pt

# Mettre à jour GRUB et redémarrer
update-grub
reboot

# Activer les VFs sur NIC (exemple : 4 fonctions virtuelles sur
eth0)
echo 4 > /sys/class/net/eth0/device/sriov_numvfs

# Rendre persistant
echo "echo 4 > /sys/class/net/eth0/device/sriov_numvfs" >>
/etc/rc.local
chmod +x /etc/rc.local
```

Paramètres de VM Proxmox :

```
Matériel -> Ajouter -> Dispositif PCI
Sélectionner : Fonction Virtuelle SR-IOV
Toutes les Fonctions : Non
GPU Principal : Non
PCI-Express : Oui (optionnel)
```

Configuration OmniUPF (dans `runtime.exs`) :

```
xdp_interfaces = "ens1f0"    # Nom de la VF SR-IOV
xdp_attach_mode = "native"
```

---

3. Passerelle PCI (XDP Natif)

Cas d'utilisation : NIC dédié pour une seule VM

Configuration :

- NIC physique entière passée à la VM
- Mode XDP : `native` ou `offload` (si SmartNIC)
- Performance : ~5-40 Mpps (dépend du NIC)

Paramètres de VM Proxmox :

```
Matériel -> Ajouter -> Dispositif PCI
Sélectionner : NIC physique (ex. : 0000:01:00.0)
Toutes les Fonctions : Oui
GPU Principal : Non
PCI-Express : Oui
```

Configuration OmniUPF (dans `runtime.exs`) :

```
xdp_interfaces = "ens1f0"
xdp_attach_mode = "native"    # ou "offload" pour SmartNIC
```

KVM/QEMU

Mode Pont :

```
virt-install \
  --name omniupf \
  --network bridge=br0,model=virtio \
  --disk path=/var/lib/libvirt/images/omniupf.qcow2 \
  ...
```

Passerelle SR-IOV :

```
<interface type='hostdev' managed='yes'>
  <source>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x10'
function='0x1'/>
  </source>
</interface>
```

VMware ESXi

vSwitch Standard (XDP Générique) :

- Adaptateur Réseau : VMXNET3
- Mode XDP : `generic`

SR-IOV (XDP Natif) :

- Activer SR-IOV dans les paramètres de l'hôte ESXi
- Ajouter un adaptateur réseau SR-IOV à la VM
- Mode XDP : `native`

Microsoft Hyper-V

Commutateur Virtuel (XDP Générique) :

- Adaptateur Réseau : Synthétique
- Mode XDP : `generic`

SR-IOV (XDP Natif) :

- Activer SR-IOV dans Hyper-V Manager
- Configurer SR-IOV sur l'adaptateur réseau virtuel
- Mode XDP : `native`

VirtualBox

Mode NAT/Pont (XDP Générique uniquement) :

- Adaptateur Réseau : VirtIO-Net ou Intel PRO/1000
- Mode XDP : `generic`
- Remarque : VirtualBox ne prend **pas** en charge SR-IOV

Compatibilité des NIC

Comprendre Mpps vs Débit

Les paquets par seconde (Mpps) et le débit (Gbps) ne sont pas directement équivalents - la relation dépend entièrement de la taille des paquets. Le trafic réseau mobile varie considérablement en taille de paquet, des petits paquets VoIP aux grands trames de streaming vidéo.

Impact de la Taille des Paquets sur le Débit

Dans les réseaux mobiles, le UPF traite les paquets encapsulés GTP-U sur l'interface N3 et les paquets IP natifs sur l'interface N6.

Surcharge d'Encapsulation GTP-U (Interface N3) :

- **En-tête IPv4 externe** : 20 octets
- **En-tête UDP externe** : 8 octets
- **En-tête GTP-U** : 8 octets
- **Surcharge totale GTP-U** : 36 octets

Paquet GTP-U Minimum (N3) :

- **En-tête IP interne** : 20 octets (IPv4)
- **En-tête UDP interne** : 8 octets
- **Charge utile minimum** : 1 octet
- **Total du paquet interne** : 29 octets
- **Plus la surcharge GTP-U** : 36 octets
- **Taille totale du paquet** : 65 octets

Débit à 1 Mpps avec des paquets GTP-U minimum :

```
65 octets x 1 000 000 pps x 8 bits/octet = 520 Mbps
```

Paquet GTP-U Maximum (N3 avec MTU 1500) :

- **MTU IP interne** : 1500 octets (paquet IP interne complet)

- **Plus la surcharge GTP-U** : 36 octets
- **Taille totale du paquet** : 1536 octets

Débit à 1 Mpps avec des paquets GTP-U maximum :

```
1536 octets x 1 000 000 pps x 8 bits/octet = 12 288 Mbps ~ 12,3
Gbps
```

Paquets IP Natifs (Interface N6) :

Sur N6 (vers Internet), les paquets sont des IP natifs sans GTP-U :

Paquet N6 Minimum :

- **En-tête IP** : 20 octets
- **En-tête UDP** : 8 octets
- **Charge utile minimum** : 1 octet
- **Total** : 29 octets

Débit à 1 Mpps avec des paquets N6 minimum :

```
29 octets x 1 000 000 pps x 8 bits/octet = 232 Mbps
```

Paquet N6 Maximum (MTU 1500) :

- **MTU IP** : 1500 octets
- **Total** : 1500 octets

Débit à 1 Mpps avec des paquets N6 maximum :

```
1500 octets x 1 000 000 pps x 8 bits/octet = 12 000 Mbps = 12 Gbps
```

Exemples de Performance dans le Monde Réel

NIC Intel X710 (capacité de 10 Mpps sur l'interface N3 avec GTP-U) :

Modèle de Trafic	Taille de Paquet Interne	Total GTP-U	Débit à 10 Mpps	Cas d'utilisation Typique
Appels VoIP (N3)	65-150 octets	101-186 octets	**0,8-1,5 Gbps**	Voix AMR-WB, G.711
Web léger (N3)	400-600 octets	436-636 octets	**3,5-5,1 Gbps**	HTTP/HTTPS, messagerie
Mobile moderne (N3)	**1200 octets**	**1236 octets**	**9,9 Gbps**	**Mélange de trafic typique 2024**
Streaming vidéo (N3)	1400-1450 octets	1436-1486 octets	**11,5-11,9 Gbps**	Morceaux vidéo HD/4K
MTU maximum (N3)	1500 octets	1536 octets	**12,3 Gbps**	Téléchargements TCP importants

Sur l'interface N6 (IP natif, pas de GTP-U) :

Modèle de Trafic	Taille de Paquet	Débit à 10 Mpps	Cas d'utilisation Typique
Paquets VoIP	65-150 octets	**0,5-1,2 Gbps**	Flux RTP de voix
Web léger	400-600 octets	**3,2-4,8 Gbps**	Requêtes HTTP
Mobile moderne	**1200 octets**	**9,6 Gbps**	**Trafic typique 2024**
Streaming vidéo	1400-1450 octets	**11,2-11,6 Gbps**	Téléchargements vidéo
MTU maximum	1500 octets	**12,0 Gbps**	Transferts de fichiers importants

À 10 Mpps avec un trafic mobile moderne (taille moyenne de 1200 octets), attendez-vous à un **débit d'environ 10 Gbps** sur les interfaces N3 et N6.

Planification de Capacité Pratique :

Avec une taille de paquet moyenne de 1200 octets (typique pour les réseaux mobiles modernes avec streaming vidéo) :

Capacité Mpps NIC	Débit N3 (GTP-U)	Débit N6 (IP Natif)	Déploiement Réaliste
1 Mpps	~1,0 Gbps	~1,0 Gbps	Site de petite cellule, passerelle IoT
5 Mpps	~4,9 Gbps	~4,8 Gbps	Site de cellule moyen, entreprise
10 Mpps	~9,9 Gbps	~9,6 Gbps	Site de grande cellule, petite ville
20 Mpps	~19,7 Gbps	~19,2 Gbps	Zone métropolitaine, ville moyenne
40 Mpps	~39,4 Gbps	~38,4 Gbps	Grande métropole, hub régional

Pilotes de Réseau Compatibles avec XDP

OmniUPF nécessite des pilotes de réseau avec support XDP pour les modes **natif** et **déchargement**. Le mode générique fonctionne avec **n'importe quel** NIC.

NIC Intel

Modèle	Pilote	Support XDP	Mode	Performance
Intel X710	i40e	Oui	Natif	~10 Mpps
Intel XL710	i40e	Oui	Natif	~10 Mpps
Intel E810	ice	Oui	Natif	~15 Mpps
Intel 82599ES	ixgbe	Oui	Natif	~8 Mpps
Intel I350	igb	Limité	Générique	~1 Mpps
Intel E1000	e1000	Non	Générique uniquement	~1 Mpps

NICs Mellanox/NVIDIA

Modèle	Pilote	Support XDP	Mode	Performance
Mellanox ConnectX-5	mlx5	Oui	Natif	~12 Mpps
Mellanox ConnectX-6	mlx5	Oui	Natif	~20 Mpps
Mellanox BlueField	mlx5	Oui	Natif + Déchargement	~40 Mpps
Mellanox ConnectX-4	mlx4	Limité	Générique	~2 Mpps

NICs Broadcom

Modèle	Pilote	Support XDP	Mode	Performance
Broadcom BCM57xxx	bnxt_en	Oui	Natif	~8 Mpps
Broadcom NetXtreme II	bnx2x	Non	Générique uniquement	~1 Mpps

Autres Fournisseurs

Modèle	Pilote	Support XDP	Mode	Performance
Netronome Agilio CX	nfp	Oui	Déchargement	~30 Mpps
Amazon ENA	ena	Oui	Natif	~5 Mpps
Solarflare SFC9xxx	sfc	Oui	Natif	~8 Mpps
VirtIO	virtio_net	Limité	Générique	~2 Mpps

Vérification du Support XDP des NIC

Vérifiez si le pilote prend en charge XDP :

```
# Trouver le pilote NIC
ethtool -i eth0 | grep driver

# Vérifier le support XDP dans le pilote
modinfo <driver_name> | grep -i xdp

# Exemple pour Intel i40e
modinfo i40e | grep -i xdp
```

Vérifiez l'attachement du programme XDP :

```
# Vérifiez si le programme XDP est attaché
ip link show eth0 | grep -i xdp

# Exemple de sortie (XDP attaché) :
# 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 xdp qdisc mq
```

NICs Recommandés par Cas d'Utilisation

Avec une taille de paquet moyenne de 1200 octets (trafic mobile moderne) :

Cas d'Utilisation	NIC Recommandé	Mode	Capacité Mpps	Dé (N
Test/Développement	N'importe quel NIC (VirtIO, E1000)	Générique	1-2 Mpps	1-2 Gbp
Site de Petite Cellule	Intel X710, Mellanox CX-5	Natif	5-10 Mpps	5-1 Gbp
Cellule Moyenne/Métro	Intel E810, Mellanox CX-6	Natif	10-20 Mpps	10- Gbp
Grande Métropole	Mellanox CX-6, Intel E810 (double)	Natif	20-40 Mpps	20- Gbp
Hub Régional	Mellanox BlueField, Netronome Agilio	Déchargement	40+ Mpps	40- Gbp
VM Proxmox (Pont)	VirtIO	Générique	1-2 Mpps	1-2 Gbp
VM Proxmox (SR-IOV)	Intel X710/E810 VF, Mellanox CX-5 VF	Natif	5-10 Mpps	5-1 Gbp

---

Ressources Supplémentaires

Documentation Officielle XDP :

- Projet XDP

- Documentation XDP du Noyau

Listes de Compatibilité des NIC :

- Support Matériel XDP de Cilium
- Pilotes XDP d'IO Visor

Exemples de Configuration

Exemple 1 : Environnement de Développement (Mode Générique)

Scénario : Tester OmniUPF sur un ordinateur portable ou une VM sans SR-IOV

```
# Configuration de développement (/etc/omniupf/runtime.exs)
xdp_interfaces = "eth0"
xdp_attach_mode = "generic"
api_port = 8080
pfcp_address = "127.0.0.1"
pfcp_port = 8805
node_id = "127.0.0.1"
n3_address = "127.0.0.1"
log_level = :debug
max_sessions = 1_000
```

Exemple 2 : Production Bare Metal (Mode Natif)

Scénario : UPF de production sur serveur bare metal avec NIC Intel X710

```
# Configuration bare metal de production
(/etc/omniupf/runtime.exs)
xdp_interfaces = "ens1f0,ens1f1"    # N3 sur ens1f0, N6 sur ens1f1
xdp_attach_mode = "native"
api_port = 8080
pfcp_address = "10.100.50.241"
pfcp_port = 8805
node_id = "10.100.50.241"
n3_address = "10.100.50.233"
n9_address = "10.100.50.234"
log_level = :info
max_sessions = 500_000
gtp_echo_interval = 30
gtp_peers = [
  %{address: parse_ip.("10.100.50.10"), echo: true, echo_interval:
30},
  %{address: parse_ip.("10.100.50.11"), echo: true, echo_interval:
30},
]
heartbeat_interval_ms = 10_000
feature_ueip = true
ueip_pool = "10.45.0.0/16"
```

Exemple 3 : VM Proxmox avec SR-IOV (Mode Natif)

Scénario : UPF de production sur VM Proxmox avec passerelle SR-IOV

```
# Configuration SR-IOV Proxmox (/etc/omniupf/runtime.exs)
xdp_interfaces = "ens1f0"           # VF SR-IOV
xdp_attach_mode = "native"
api_port = 8080
pfcp_address = "192.168.100.10"
pfcp_port = 8805
node_id = "192.168.100.10"
n3_address = "192.168.100.10"
log_level = :info
max_sessions = 100_000
gtp_echo_interval = 15
gtp_peers = [
  %{address: parse_ip.("192.168.100.50"), echo: true},
]
```

Exemple 4 : Mode PGW-U (EPC 4G)

Scénario : OmniUPF agissant comme PGW-U dans un réseau EPC 4G

```
# Configuration PGW-U (/etc/omniupf/runtime.exs)
xdp_interfaces = "eth0"
xdp_attach_mode = "native"
api_port = 8080
pfcp_address = "10.200.1.10"
pfcp_port = 8805
node_id = "10.200.1.10"
n3_address = "10.200.1.10"          # Interface S5/S8 (GTP-U)
log_level = :info
max_sessions = 200_000
gtp_echo_interval = 20
gtp_peers = [
  %{address: parse_ip.("10.200.1.50"), echo: true, echo_interval:
20},
]
heartbeat_interval_ms = 5_000
```

Exemple 5 : Multi-Mode (UPF + PGW-U Simultanément)

Scénario : OmniUPF servant à la fois des réseaux 5G et 4G simultanément

```
# Configuration multi-mode (/etc/omniupf/runtime.exs)
xdp_interfaces = "eth0,eth1"
xdp_attach_mode = "native"
api_port = 8080
pfcp_address = "10.50.1.100"
pfcp_port = 8805
node_id = "10.50.1.100"
n3_address = "10.50.1.100"
n9_address = "10.50.1.101"
log_level = :info
max_sessions = 300_000
gtp_echo_interval = 15
gtp_peers = [
  %{address: parse_ip.("10.50.2.10"), echo: true, echo_interval:
15},
  %{address: parse_ip.("10.50.2.20"), echo: true, echo_interval:
15},
]
heartbeat_interval_ms = 10_000
feature_ueip = true
ueip_pool = "10.60.0.0/16"
```

Exemple 6 : Mode Offload SmartNIC

Scénario : Déploiement ultra-haut débit avec Netronome Agilio CX SmartNIC

```
# Configuration offload SmartNIC (/etc/omniupf/runtime.exs)
xdp_interfaces = "enp1s0np0"        # Interface SmartNIC
xdp_attach_mode = "offload"
api_port = 8080
pfcp_address = "10.10.1.50"
pfcp_port = 8805
node_id = "10.10.1.50"
n3_address = "10.10.1.50"
log_level = :warning
max_sessions = 1_000_000
far_map_size = 2_000_000
qer_map_size = 1_000_000
gtp_echo_interval = 30
gtp_peers = [
  %{address: parse_ip.("10.10.2.10"), echo: true},
  %{address: parse_ip.("10.10.2.20"), echo: true},
  %{address: parse_ip.("10.10.2.30"), echo: true},
]
heartbeat_interval_ms = 15_000
```

Dimensionnement des Cartes et Planification de Capacité

Définir les Tailles de Carte

Définissez `max_sessions` et configurez les tailles de carte pour votre déploiement :

```
max_sessions = 100_000
far_map_size = 131_070
qer_map_size = 65_535
```

Utilisation de la mémoire : ~91 Mo pour 100K sessions

Estimation de Capacité

Calculer le nombre maximum de sessions :

```
Max Sessions = min(
  pdr_map_size / 2,
  far_map_size / 2,
  qer_map_size
)
```

Exemple :

- Carte PDR : 200 000
- Carte FAR : 200 000
- Carte QER : 100 000

Max Sessions = min(100 000, 100 000, 100 000) = **100 000**

Exigences en Mémoire

Utilisation de mémoire par session :

- PDR : 2 x 212 B = 424 B
- FAR : 2 x 20 B = 40 B
- QER : 1 x 36 B = 36 B
- URR : 2 x 20 B = 40 B
- **Total** : ~540 B par session

Pour 100K sessions : ~52 Mo de mémoire noyau

Recommandation : Assurez-vous que la limite de mémoire verrouillée permet 2x l'utilisation estimée :

```
# Vérifier la limite actuelle
ulimit -l

# Définir illimité (nécessaire pour eBPF)
ulimit -l unlimited
```

Documentation Connexe

- **Guide d'Architecture** - Détails techniques eBPF/XDP et optimisation des performances
- **Guide de Gestion des Règles** - Configuration PDR, FAR, QER, URR
- **Guide de Surveillance** - Statistiques, surveillance de capacité et alertes
- **Référence des Métriques** - Référence complète des métriques Prometheus
- **Guide de l'Interface Web** - Opérations du panneau de contrôle
- **Guide des Opérations** - Vue d'ensemble de l'architecture et du déploiement UPF
- **Guide du Jardin Clos** - Redirection hors crédit, portail captif et configuration de liste blanche

IPv6 / Dual-Stack sur le Plan Utilisateur

Comment l'OmniUPF gère le trafic utilisateur IPv6 pour les sessions PDN IPv6 et IPv4v6 : les prérequis du noyau, le pool d'adresses UE, la programmation de la liaison descendante IPv6 depuis PFCP, et la publicité des routes hôtes UE au réseau via OSPFv3.

Cette page couvre les responsabilités du **plan utilisateur**. Le plan de contrôle (négociation du type de PDN, PAA, Gx, et signalisation de l'adresse IPv6 de l'UE au UPF via PFCP) est géré par le SMF/PGW-C. Voir aussi Gestion des Routes et le Guide de Configuration.

Table des Matières

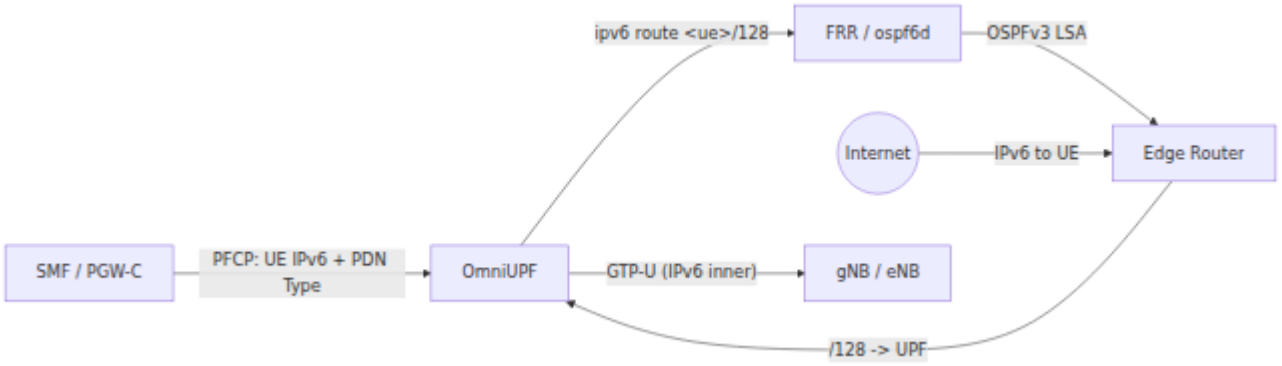
- Aperçu
- Prérequis du Noyau
- Pool d'Adresses IPv6 de l'UE
- Publicité de Route OSPFv3
- Comportement du Plan de Données
- Dépannage

Aperçu

Lorsqu'une session PDN IPv6 (ou IPv4v6) est établie, le SMF/PGW-C envoie l'adresse IPv6 de l'UE et le type de PDN au UPF via PFCP. Le UPF alors :

1. Programme la **recherche de liaison descendante IPv6** afin que les paquets destinés à l'UE soient appariés et tunnelés.
2. Ajoute une **route hôte** `/128` pour l'adresse de l'UE et la publie au routeur de bord via **OSPFv3**, afin que le trafic descendant soit attiré vers ce UPF.

3. Transmet le trafic utilisateur dans les deux directions (décapsulation montante, encapsulation GTP-U descendante) pour les paquets internes IPv6.



Prérequis du Noyau

IPv6 doit être activé et le transfert activé au niveau de l'hôte. Appliquez via un drop-in sysctl (par exemple sous `/etc/sysctl.d/`).

Paramètre	Valeur requise	Description
`net.ipv6.conf.all.disable_ipv6`	`0`	IPv6 doit être activé. L'interface de plan de données du UPF a besoin d'une adresse IPv6 globale pour le transit et le routage.
`net.ipv6.conf.default.disable_ipv6`	`0`	Assure qu'IPv6 reste activé sur les interfaces nouvellement apparues.
`net.ipv6.conf.all.forwarding`	`1`	Active le transfert IPv6 afin que le trafic montante de l'UE sorte et que le trafic descendante soit routé vers le plan de données.

L'interface de plan de données a également besoin d'une adresse IPv6 globale sur le segment de transit (en direction du routeur de bord) pour qu'OSPFv3 puisse former une adjacence et échanger des routes.

Pool d'Adresses IPv6 de l'UE

Lorsque le UPF alloue des adresses UE (plutôt que de les recevoir du SMF), définissez un pool IPv6 dans `runtime.exs`. Lorsque le SMF/PGW-C fournit l'adresse de l'UE via PFCP, cette adresse est utilisée directement et ce pool n'est pas consulté pour la session.

```
# /etc/omniupf/runtime.exs
config :upf_ex,
  ueip_pool: "2001:db8:6f85:70::/64"
```

Paramètre	Type	Requis	Par défaut	Description
`ueip_pool`	String (CIDR)	Non	-	Pool IP de l'UE utilisé lorsque le UPF effectue l'allocation d'adresses. Accepte un CIDR IPv4 ou IPv6 ; la famille d'adresses est détectée à partir de la chaîne.

Publicité de Route OSPFv3

Chaque adresse IPv6 UE active est installée comme une route hôte `/128` et est publiée au routeur de bord, reflétant le comportement IPv4 `/32`. Les routes sont gérées par le gestionnaire de routes et redistribuées dans OSPF par FRR.

```
# /etc/omniupf/runtime.exs
config :upf_ex,
  route_manager_enabled: true,
  route_manager_type: "frr"
```

Paramètre	Type	Requis	Par défaut	Description
`route_manager_enabled`	Boolean	Non	`false`	Active la gestion des routes hôtes UE. Doit être `true` pour publier les routes UE.
`route_manager_type`	String	Non	-	Backend de route. `"frr"` injecte des routes via FRR/`vtysh` ; le backend statique utilise directement les commandes de route `ip`/`ipv6` du noyau. Utilisez `"frr"` pour la publicité OSPFv3.

L'instance FRR doit exécuter **OSPFv3** (`ospf6d`) sur l'interface de transit et redistribuer les routes statiques, afin que les routes hôtes UE `/128` soient inondées vers le routeur de bord. Les routes hôtes IPv6 sont ajoutées avec la commande `ipv6 route <addr>/128`

- la commande IPv4 `ip route` n'accepte pas les préfixes IPv6. Voir Gestion des Routes pour le cycle de vie du gestionnaire de routes.

Comportement du Plan de Données

Direction	Comportement
Descendant (réseau → UE)	La destination interne IPv6 est appariée avec la table de liaison descendante IPv6 programmée depuis la session PFCP ; les paquets correspondants sont encapsulés en GTP-U vers le gNB/eNB (ou mis en tampon si l'action de la règle est BUFF).
Montant (UE → réseau)	GTP-U est décapsulé et le paquet interne IPv6 est transmis par l'interface N6 en utilisant la FIB IPv6 du noyau (d'où le sysctl de transfert).
Mise en Tampon (BUFF)	Les paquets IPv6 descendantes mis en tampon sont livrés au listener de tampon en espace utilisateur et vidés lors de la transition vers le transfert, de la même manière qu'IPv4.

Aucun assouplissement `accept_local` / filtre de chemin inverse n'est requis sur les versions actuelles ; le chemin de tampon utilise une source extérieure non locale afin que les paquets ne soient pas rejetés comme martiens.

Dépannage

Le trafic descendant IPv6 n'atteint pas l'UE

Symptômes : La session est établie et la montée fonctionne, mais le trafic du réseau vers l'UE est rejeté.

Causes possibles :

- La route hôte `/128` de l'UE n'est pas publiée (gestionnaire de routes désactivé, ou OSPFv3 non exécuté sur le UPF).
- L'adjacence OSPFv3 entre le UPF et le routeur de bord n'est pas établie.

- Le routeur de bord n'a pas encore installé le `/128` (délai de convergence).

Résolution :

1. Confirmez `route_manager_enabled: true` et `route_manager_type: "frr"`.
2. Vérifiez que l'instance FRR du UPF exécute OSPFv3 sur l'interface de transit et redistribue les routes statiques.
3. Confirmez que l'adjacence OSPFv3 est complète et que le routeur de bord montre le `/128` de l'UE appris via OSPF.

Le trafic utilisateur IPv6 n'est pas du tout transmis

Symptômes : Ni la montée ni la descente IPv6 ne fonctionnent, même avec des routes présentes.

Causes possibles :

- IPv6 est désactivé sur l'hôte, ou le transfert est désactivé.
- L'interface de plan de données n'a pas d'adresse IPv6 globale.

Résolution :

1. Vérifiez les prérequis du noyau (`disable_ipv6=0`, `forwarding=1`).
2. Confirmez que l'interface de plan de données a une adresse IPv6 globale sur le segment de transit.

Référence des métriques

Ce document décrit toutes les métriques Prometheus exposées par OmniUPF sur le point de terminaison `/api/metrics` (l'API HTTP fonctionne sur `api_port`, par défaut `8080`). Notez le préfixe `/api` - chaque route API est servie sous celui-ci.

Les noms, types et étiquettes des métriques ci-dessous reflètent les métriques exposées par OmniUPF.

Catégories de métriques

1. **Métriques PFCP** - compteurs de messages de plan de contrôle, d'association, de session et d'erreur
2. **Métriques du cycle de vie des sessions** - compteurs de création/suppression/modification de session et latence d'établissement
3. **Métriques de chemin lent et de mise en tampon en espace utilisateur** - compteurs de mise en tampon de liaison descendante, de jardin clos et d'indications d'erreur TEID inconnu
4. **Métriques de rapport de données de liaison descendante (DLDR)** - notifications de demande de rapport de session PFCP et suivi de l'index FAR
5. **Métriques GTP-U** - réponses d'écho et compteurs d'indications d'erreur
6. **Métriques URR** - compteurs de rapport d'utilisation et minuteries URR actives
7. **Métriques de carte eBPF** - jauges d'entrée de carte de plan de données
8. **Débit du plan de données** - compteurs `_total` par direction (N3/N6), régulés par `expose_xdp_metrics` (#57) ; également à la demande via `/api/v1/xdp_stats`
9. **Métriques de ressources / routage** - jauges de pool TEID/FAR-ID et de route UE
10. **Métriques de jardin clos** - interception de portail captif, transfert et compteurs de spoofing DNS

Métriques PFCP

Compteurs de protocole de plan de contrôle entre le UPF et ses pairs PFCP (SMF / PGW-C / SGW-C).

Nom de la métrique	Type	Étiquettes	D
`upf_pfcp_associations`	Gauge	none	Associati actuellem les pairs)
`upf_pfcp_sessions`	Gauge	none	Sessions actives (t
`upf_pfcp_peer_restarts_total`	Counter	`peer`	Détectior des pairs timestam
`upf_pfcp_duplicate_requests_total`	Counter	none	Requêtes détectées
`upf_orphaned_sessions_deleted_total`	Counter	none	Sessions suite d'ur pair
`upf_pfcp_messages_rx_total`	Counter	`type`, `peer`	Messages type de r
`upf_pfcp_messages_tx_total`	Counter	`type`, `peer`	Messages type de r
`upf_pfcp_errors_total`	Counter	`type`	Rejets de étiquetés rejet (par `session_` `mandator`

Métriques du cycle de vie des sessions

Nom de la métrique	Type	Étiquet
`upf_sessions_created_total`	Counter	none
`upf_sessions_deleted_total`	Counter	none
`upf_sessions_modified_total`	Counter	none
`upf_session_establishment_duration_microseconds`	Histogram	none

Métriques de chemin lent et de mise en tampon en espace utilisateur

Le plan de données eBPF envoie des paquets vers le chemin lent en espace utilisateur (port UDP `22152`, par défaut) pour trois tâches distinctes : **mise en tampon de liaison descendante** lorsque l'UE est inactif (retenu jusqu'à ce que l'UE soit pagé), **redirection de jardin clos**, et réponse aux **G-PDUs montants TEID inconnu** avec une indication d'erreur GTP-U. Seules les

métriques `buffered`/`flushed`/`current` ci-dessous décrivent le tampon proprement dit ; le jardin clos a ses propres métriques `upf_walled_garden_*` et le TEID inconnu est compté comme `upf_gtpu_error_indications_sent_total`.

Nom de la métrique	Type	Étiquettes	D
`upf_userspace_punt_received_total`	Counter	none	Tous les p vers le ch utilisateu jardin clo
`upf_gtpu_oversize_punted_total`	Counter	none	Paquets G descenda excessive au-dessu sortie) en utilisateu fragment extérieur soutenue abonné p taille com chemin le `gtpu_ove`
`upf_buffer_packets_buffered_total`	Counter	none	Paquets a au tampo
`upf_buffer_packets_flushed_total`	Counter	none	Paquets v
`upf_buffer_packets_dropped_total`	Counter	`reason`	Paquets s admis da
`upf_buffer_packets_current`	Gauge	none	Paquets a dans le ta
`upf_buffer_flush_operations_total`	Counter	none	Opération tampon r de FAR)

Nom de la métrique	Type	Étiquettes	D
`upf_buffer_flush_packets_sent_total`	Counter	none	Paquets e opération
`upf_buffer_flush_errors_total`	Counter	`reason`	Échecs d vidage de

Valeurs `reason` de `upf_buffer_packets_dropped_total` (provenant des suppressions d'ajout au tampon et des erreurs d'écoute du tampon) :

- `global_limit` - capacité totale du tampon (`buffer_max_total`) atteinte
- `far_limit` - limite de tampon par FAR (`buffer_max_per_far`) atteinte
- `expired` - TTL (`buffer_ttl_ms`) dépassé avant le vidage
- `read_error` - erreur de lecture depuis le socket du tampon
- `too_small` - paquet trop petit pour un en-tête GTP
- `invalid_gtp_type` - type de message GTP non-G-PDU
- `not_buffering_far` - le FAR n'a pas d'action BUFF
- `error_indication_failed` - impossible d'envoyer l'indication d'erreur GTP-U pour un uplink TEID inconnu

> L'uplink TEID inconnu **n'est pas** une suppression de tampon : il est répondu avec une indication d'erreur GTP-U et compté par `upf_gtpu_error_indications_sent_total` (une réponse correcte du chemin lent, pas de perte de paquet).

Valeurs `reason` de `upf_buffer_flush_errors_total` :

- `far_lookup_failed` - impossible de rechercher les informations FAR dans la carte eBPF
- `no_forw_action` - le FAR n'a pas d'action FORW définie
- `connection_failed` - échec de l'ouverture d'un socket UDP pour le vidage

Métriques de rapport de données de liaison descendante (DLDR)

Métriques pour les notifications de demande de rapport de session PFCP envoyées au plan de contrôle lorsque des paquets de liaison descendante sont mis en tampon. Ces notifications déclenchent le SMF/PGW-C pour paginer l'UE.

Nom de la métrique	Type	Étiquettes	Description
`upf_dldr_sent_total`	Counter	`peer`	Rapports de données de liaison descendante envoyés, étiquetés par adresse SMF/PGW-C
`upf_dldr_errors_total`	Counter	none	Erreurs d'encodage / d'envoi DLDR
`upf_dldr_skipped_total`	Counter	none	DLDR sautés car le FAR a déjà été notifié
`upf_far_index_size`	Gauge	none	FAR enregistrés dans le FarIndex pour notification DLDR

Métriques GTP-U

Compteurs de gestion de chemin GTP-U et d'indication d'erreur. Les indications d'erreur sont échangées lorsqu'un pair reçoit des paquets pour un TEID inconnu (généralement après un redémarrage).

Nom de la métrique	Type	Étiquettes
`upf_gtpu_echo_responses_tx_total`	Counter	none
`upf_gtpu_error_indications_sent_total`	Counter	none
`upf_gtpu_error_indications_rx_total`	Counter	none
`upf_error_indication_sessions_deleted_total`	Counter	none

Quand les indications d'erreur sont envoyées : le UPF reçoit un paquet GTP-U pour un TEID qui n'existe pas (par exemple après un redémarrage, ou la session a déjà été supprimée), et informe l'expéditeur d'arrêter.

Quand les indications d'erreur sont reçues : un pair en aval ne reconnaît pas un TEID que le UPF a transmis ; le UPF supprime la session affectée pour arrêter le transfert vers un tunnel mort.

Métriques URR (Règle de rapport d'utilisation)

Nom de la métrique	Type	Étiquettes	Description
`upf_urr_reports_sent_total`	Counter	`trigger`	Rapports d'utilisation URR envoyés, étiquetés par déclencheur de rapport
`upf_urr_report_errors_total`	Counter	none	Erreurs d'encodage / d'envoi de rapport URR
`upf_urr_active_timers`	Gauge	none	Minuteries URR actives (périodiques / seuil de temps / validité de quota)

`trigger` reflète le déclencheur de rapport URR qui a été activé - par exemple `periodic` (PERIO), `volume_threshold` (VOLTH), `time_threshold` (TIMTH), `quota_validity` (QUVTI).

Les compteurs de bytes par URR ne sont pas exportés en tant que métriques Prometheus (pour éviter une haute cardinalité). Lisez les statistiques URR individuelles via l'API REST à `/api/v1/urr_map`.

Métriques de carte eBPF

Jauges pour l'utilisation de la carte de plan de données, rafraîchies à chaque extraction.

Nom de la métrique	Type	Étiquettes	Description
`upf_ebpf_far_entries`	Gauge	none	Entrées FAR dans la carte eBPF
`upf_ebpf_pdr_uplink_entries`	Gauge	none	Entrées PDR montantes dans la carte eBPF (PDR avec un TEID)
`upf_ebpf_pdr_downlink_entries`	Gauge	none	Entrées PDR descendantes dans la carte eBPF (PDR indexés par IP UE)

`upf_ebpf_far_entries` reflète le pool d'ID FAR alloués (`upf_far_ids_allocated`), puisque la far_map contient une entrée par ID FAR alloué. Les jauges PDR sont agrégées à partir de l'état de session en direct à chaque extraction (le coût est O(total sessions)).

Débit du plan de données (N3/N6)

Compteurs de paquets et de bytes par direction pour le trafic de plan utilisateur (**N3** = côté GTP-U vers le gNB/eNB ; **N6** = côté SGi vers le réseau de données) proviennent de la carte percpu `upf_ext_stat` eBPF.

Nom de la métrique	Type	Étiquettes	Description
`upf_n3_rx_packets_total` / `upf_n3_tx_packets_total`	Counter	none	Paquets N3 (GTP-U) entrant / sortant du chemin de données XDP
`upf_n6_rx_packets_total` / `upf_n6_tx_packets_total`	Counter	none	Paquets N6 (SGi) entrant / sortant du chemin de données XDP
`upf_n3_rx_bytes_total` / `upf_n3_tx_bytes_total`	Counter	none	Bytes N3 (GTP-U) entrant / sortant
`upf_n6_rx_bytes_total` / `upf_n6_tx_bytes_total`	Counter	none	Bytes N6 (SGi) entrant / sortant

Régulé par `expose_xdp_metrics` (problème #57). La lecture de `upf_ext_stat` sur le chemin de demande (par extraction Prometheus) a été prouvée pour déclencher un **wedge TX-ring** générique-XDP `virtio_net`. Deux atténuations s'appliquent : la lecture est effectuée hors du chemin de demande par le `Sampler` en arrière-plan (à un intervalle fixe, jamais lors d'une extraction), et elle est régulée derrière `config :upf_ex, expose_xdp_metrics: true`. Lorsqu'elle est désactivée, ces compteurs ne sont pas déclarés ou exportés.

Ce sont des **Compteurs** cumulés `_total`. La carte sous-jacente `upf_ext_stat` est fixée, donc elle survit à un redémarrage d'omniupf mais est remise à zéro lors d'un redéploiement eBPF `.o` ; le collecteur traduit sa valeur absolue en incréments de compteur monotoniques afin que `rate()` reste correct à travers les deux. Ne lisez pas ces valeurs comme des jauges. Utilisez `rate()`.

À la demande JSON : `GET /api/v1/xdp_stats` renvoie les mêmes champs et touche uniquement eBPF lorsque vous l'appelez explicitement. Champs : `rx_n3`

/ `tx_n3` / `rx_n6` / `tx_n6` (paquets) et `rx_n3_bytes` / `tx_n3_bytes` / `rx_n6_bytes` / `tx_n6_bytes`.

Métriques de ressources / routage

Nom de la métrique	Type	Étiquettes	Description
`upf_routes_total`	Gauge	none	Routes IP UE actuellement suivies par le gestionnaire de routes
`upf_teid_allocated`	Gauge	none	TEIDs actuellement alloués depuis le pool
`upf_far_ids_allocated`	Gauge	none	ID FAR actuellement alloués depuis le pool

Métriques de jardin clos

Compteurs pour le jardin clos hors crédit / portail captif. Voir le Guide du jardin clos pour le comportement.

Nom de la métrique	Type	Étiquettes
`upf_walled_garden_active_redirects`	Gauge	none
`upf_walled_garden_packets_intercepted_total`	Counter	none
`upf_walled_garden_packets_dropped_total`	Counter	none
`upf_walled_garden_packets_forwarded_total`	Counter	`dst_ip`
`upf_walled_garden_bytes_forwarded_total`	Counter	`dst_ip`
`upf_walled_garden_dns_spoofed_total`	Counter	`domain`

Nom de la métrique	Type	Étiquettes
`upf_walled_garden_dns_forwarded_total`	Counter	`domain`

> **Remarque sur la cardinalité.** Les étiquettes `dst_ip` et `domain` sont illimitées. Gardez la liste blanche étroite, et agrégerez ces étiquettes dans les règles d'enregistrement si vous extrayez un jardin clos occupé.

Utilisation des métriques Prometheus

Accéder aux métriques

Les métriques sont exposées sur le point de terminaison `/api/metrics` du serveur API HTTP, qui écoute sur `api_port` (par défaut `8080`). Notez le préfixe `/api` - chaque route API est servie sous celui-ci :

```
# Voir les métriques brutes
curl http://localhost:8080/api/metrics

# Exemple de sortie
upf_pfcp_sessions 42
upf_pfcp_associations 2
upf_dldr_sent_total{peer="10.100.50.241"} 17
```

Configuration de Prometheus

Ajoutez la cible OmniUPF à votre `prometheus.yml` :

```yaml
scrape_configs:
  - job_name: 'omniupf'
    metrics_path: /api/metrics
    static_configs:
      - targets: ['localhost:8080']
```

Tableaux de bord Grafana

Panneaux utiles à construire à partir de ces métriques :

- Comptes de sessions et d'associations (`upf_pfcp_sessions`, `upf_pfcp_associations`)
- Le débit N3/N6 par direction est disponible à la demande via `/api/v1/xdp_stats` (supprimé de l'extraction, voir #57)
- Pression sur le tampon (`upf_buffer_packets_current`, taux de suppression par `reason`)
- Activité DLDR / de pagination (`rate(upf_dldr_sent_total[5m])`)
- Taux d'indication d'erreur et de redémarrage de pair pour l'alerte de santé du tunnel

Documentation connexe

- **Guide de surveillance** - Surveillance des statistiques, planification de capacité et alertes
- **Guide de configuration** - Configurer `api_port`, réglage du tampon et autres options UPF
- **Guide de l'interface Web** - Voir les métriques dans la page des statistiques
- **Guide d'architecture** - Chemin de données eBPF et optimisation des performances
- **Guide de gestion des règles** - Comprendre les métriques PDR, FAR, QER, URR
- **Guide du jardin clos** - Comportement de redirection de portail captif et de liste blanche
- **Guide de dépannage** - Utiliser les métriques pour le diagnostic

Guide de Surveillance

Table des Matières

1. Aperçu
2. Surveillance des Statistiques
3. Surveillance de la Capacité
4. Métriques de Performance
5. Alertes et Seuils
6. Planification de la Capacité
7. Dépannage des Problèmes de Performance

Aperçu

Une surveillance efficace d'OmniUPF est essentielle pour maintenir la qualité du service, prévenir l'épuisement des capacités et dépanner les problèmes de performance. OmniUPF fournit des métriques complètes en temps réel via son interface Web et son API REST.

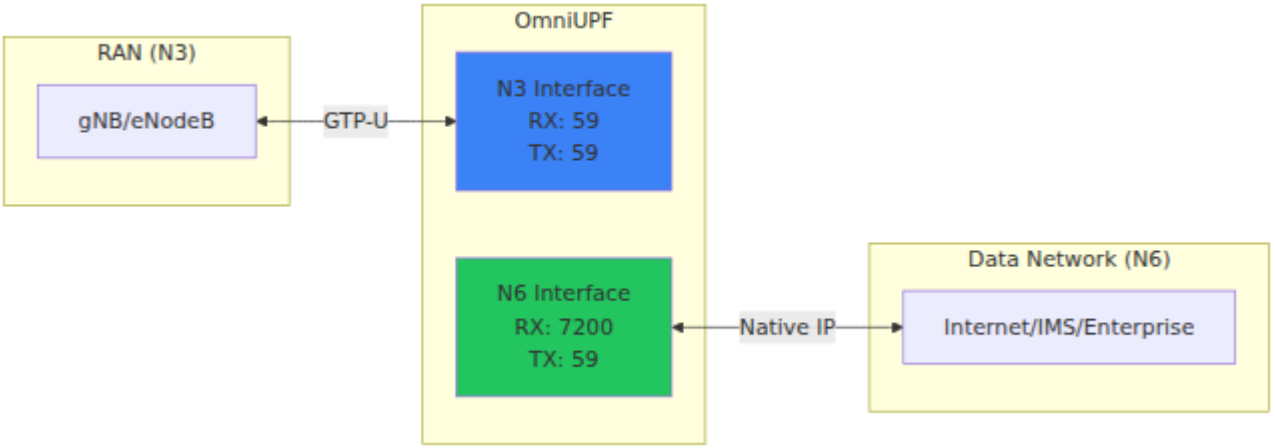
Catégories de Surveillance

Catégorie	Objectif	Fréquence de Mise à Jour	Métriques Clés
Statistiques de Paquets	Suivre les taux de traitement des paquets et les erreurs	En temps réel	Paquets RX/TX, pertes, répartition des protocoles
Statistiques d'Interface	Surveiller la distribution du trafic N3/N6	En temps réel	N3 RX/TX, N6 RX/TX
Statistiques XDP	Suivre la performance du chemin de données du noyau	En temps réel	XDP traités, passés, abandonnés, avortés
Statistiques de Routage	Surveiller les décisions de routage des paquets	En temps réel	Recherches FIB, hits/misses de cache
Capacité de la Carte eBPF	Prévenir l'épuisement des ressources	Toutes les 10s	Pourcentages d'utilisation de la carte, utilisé vs. capacité
Statistiques de Tampon	Suivre la mise en tampon des paquets pendant la mobilité	Toutes les 5s	Paquets mis en tampon, âge du tampon, nombre de FAR

Surveillance des Statistiques

Statistiques d'Interface N3/N6

Les statistiques d'interface N3/N6 fournissent une visibilité sur la distribution du trafic entre le RAN (N3) et le Réseau de Données (N6).



Métriques:

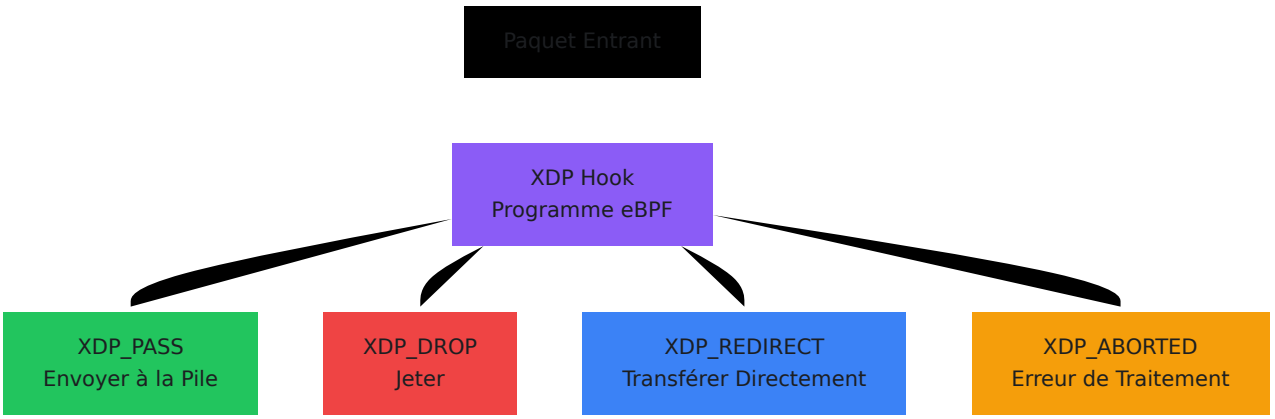
- **RX N3**: Paquets reçus du RAN (trafic GTP-U montant)
- **TX N3**: Paquets transmis au RAN (trafic GTP-U descendant)
- **RX N6**: Paquets reçus du Réseau de Données (IP native descendante)
- **TX N6**: Paquets transmis au Réseau de Données (IP native montante)
- **Total**: Compte agrégé des paquets sur toutes les interfaces

Comportement Attendu:

- **RX N3 ≈ TX N6**: Les paquets montants circulent du RAN au Réseau de Données
- **RX N6 ≈ TX N3**: Les paquets descendants circulent du Réseau de Données au RAN
- Un déséquilibre significatif peut indiquer :
  - Trafic asymétrique (téléchargements >> téléversements)
  - Pertes de paquets ou erreurs de transfert
  - Mauvaises configurations de routage

Statistiques XDP

Les statistiques XDP (eXpress Data Path) montrent la performance du traitement des paquets au niveau du noyau.



Métriques:

- **Abandonné**: Le programme XDP a rencontré une erreur (doit toujours être 0)

- **Jeter**: Paquets intentionnellement jetés par le programme XDP
- **Passer**: Paquets passés à la pile réseau pour un traitement ultérieur
- **Rediriger**: Paquets redirigés directement vers l'interface de sortie
- **TX**: Paquets transmis via XDP

Interprétation:

- **Abandonné > 0**: Problème critique avec le programme eBPF ou compatibilité du noyau
- **Jeter > 0**: Pertes basées sur des politiques ou paquets invalides
- **Pass élevé**: La plupart des paquets traités dans la pile réseau (normal)
- **Rediriger élevé**: Paquets transférés directement (performance optimale)

Statistiques de Paquets

Répartition détaillée des protocoles de paquets et compteurs de traitement.

Compteurs de Protocoles:

- **RX ARP**: Paquets du Protocole de Résolution d'Adresse
- **RX GTP ECHO**: Requête/Réponse GTP-U Echo (keepalive)
- **RX GTP OTHER**: Autres messages de contrôle GTP
- **RX GTP PDU**: Données utilisateur encapsulées GTP-U (trafic principal)
- **RX GTP UNEXP**: Types de paquets GTP inattendus
- **RX ICMP**: Protocole de Contrôle Internet (ping, erreurs)
- **RX ICMP6**: Paquets ICMPv6
- **RX IP4**: Paquets IPv4
- **RX IP6**: Paquets IPv6
- **RX OTHER**: Autres protocoles
- **RX TCP**: Paquets du Protocole de Contrôle de Transmission
- **RX UDP**: Paquets du Protocole de Datagramme Utilisateur

Cas d'Utilisation:

- **Surveiller le compte de PDU GTP-U**: Indicateur principal du trafic utilisateur
- **Vérifier ICMP pour la connectivité**: Test de la portée du réseau
- **Suivre le ratio TCP vs UDP**: Modèles de trafic d'application
- **Détecter des protocoles inattendus**: Problèmes de sécurité ou de mauvaise configuration

---

Statistiques de Routage

Statistiques de recherche FIB (Forwarding Information Base) pour les décisions de routage.

Recherche FIB IPv4:

- **Cache**: Recherches de route mises en cache (chemin rapide)
- **OK**: Recherches de route réussies

Recherche FIB IPv6:

- **Cache**: Recherches de route IPv6 mises en cache
- **OK**: Recherches de route IPv6 réussies

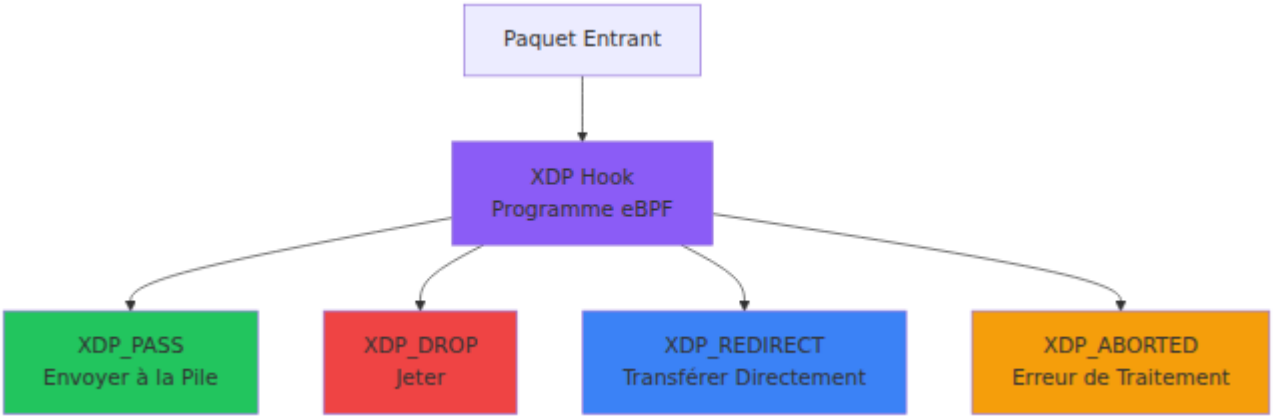
Indicateurs de Performance:

- **Taux de Hit de Cache Élevé**: Indique une bonne performance du cache de routage
- **Nombre OK Élevé**: Confirme que les tables de routage sont correctement configurées
- **Peu ou Pas de Recherches**: Peut indiquer que le trafic ne circule pas ou que le routage est contourné

Surveillance de la Capacité

Capacité de la Carte eBPF

La surveillance de la capacité de la carte eBPF prévient les échecs d'établissement de session dus à l'épuisement des ressources.



Cartes eBPF Critiques

far_map (Règles d'Action de Transfert):

- **Capacité**: 131,070 entrées
- **Taille de Clé**: 4 B (ID FAR)
- **Taille de Valeur**: 16 B (paramètres de transfert)
- **Utilisation de Mémoire**: ~2.6 MB
- **Criticité**: Élevée - Utilisée pour toutes les décisions de transfert de paquets

pdr_map_downlin (PDRs Descendants - IPv4):

- **Capacité**: 131,070 entrées
- **Taille de Clé**: 4 B (adresse IPv4 de l'UE)
- **Taille de Valeur**: 208 B (info PDR)
- **Utilisation de Mémoire**: ~27 MB
- **Criticité**: Critique - L'établissement de session échoue si plein

pdr_map_downlin_ip6 (PDRs Descendants - IPv6):

- **Capacité**: 131,070 entrées
- **Taille de Clé**: 16 B (adresse IPv6 de l'UE)
- **Taille de Valeur**: 208 B (info PDR)
- **Utilisation de Mémoire**: ~29 MB
- **Criticité**: Critique - L'établissement de session IPv6 échoue si plein

pdr_map_teid_ip (PDRs Montants):

- **Capacité**: 131,070 entrées
- **Taille de Clé**: 4 B (TEID)
- **Taille de Valeur**: 208 B (info PDR)
- **Utilisation de Mémoire**: ~27 MB
- **Criticité**: Critique - Le trafic montant échoue si plein

qer_map (Règles d'Application de QoS):

- **Capacité**: 65,535 entrées
- **Taille de Clé**: 4 B (ID QER)
- **Taille de Valeur**: 32 B (paramètres de QoS)
- **Utilisation de Mémoire**: ~2.3 MB
- **Criticité**: Moyenne - Application de QoS uniquement

urr_map (Règles de Rapport d'Utilisation):

- **Capacité**: 131,070 entrées
- **Taille de Clé**: 4 B (ID URR)
- **Taille de Valeur**: 16 B (compteurs de volume)
- **Utilisation de Mémoire**: ~2.6 MB
- **Criticité**: Faible - N'affecte que la facturation

Seuils de Capacité

Seuil	Action Requise
0-50% (Vert)	Fonctionnement normal - Aucune action requise
50-70% (Jaune)	Précaution - Surveiller les tendances de croissance, planifier une augmentation de capacité
70-90% (Ambre)	Avertissement - Planifier une augmentation de capacité dans la semaine
90-100% (Rouge)	Critique - Action immédiate requise, de nouvelles sessions échoueront

Procédure d'Augmentation de Capacité

Avant d'augmenter la capacité:

1. Examiner les tendances d'utilisation actuelles
2. Estimer le taux de croissance futur
3. Calculer la capacité requise

Étapes pour augmenter la capacité de la carte:

1. Arrêter le service OmniUPF
2. Mettre à jour le fichier de configuration UPF avec les nouvelles tailles de carte
3. Redémarrer le service OmniUPF
4. Vérifier la nouvelle capacité dans la vue de Capacité
5. Surveiller l'établissement de session réussi

Remarque: Changer la capacité de la carte eBPF nécessite un redémarrage de l'UPF et efface toutes les sessions existantes.

Métriques de Performance

Pour des informations détaillées sur toutes les métriques Prometheus exposées par OmniUPF, voir la **Référence des Métriques**.

Taux de Traitement des Paquets

Calcul:

```
Taux de Paquets (pps) = (Delta du Compte de Paquets) / (Delta de
Temps en secondes)
```

Exemple:

- Paquets RX initiaux: 7,000

- Après 10 secondes: 17,000
- Taux de Paquets = (17,000 - 7,000) / 10 = 1,000 pps

Objectifs de Performance:

- **Petit UPF**: 10,000 - 100,000 pps
- **UPF Moyen**: 100,000 - 1,000,000 pps
- **Grand UPF**: 1,000,000 - 10,000,000 pps

Indicateurs de Goulot d'Étranglement:

- Compte XDP abandonné en augmentation
- Utilisation CPU élevée
- Pertes de paquets en augmentation
- Latence en augmentation

Calcul du Débit

Calcul:

```
Débit (Mbps) = (Delta du Compte d'Octets × 8) / (Delta de Temps en
secondes × 1,000,000)
```

Exemple:

- Octets RX initiaux: 500 MB
- Après 60 secondes: 800 MB
- Débit = (300 MB × 8) / (60 × 1,000,000) = 40 Mbps

Le débit par direction est exporté en tant que compteurs `_total` (`upf_n3_rx_packets_total`, `upf_n3_rx_bytes_total`, … équivalents N6) lorsque `expose_xdp_metrics` est activé. N3 est le côté GTP-U (gNB), N6 est le côté SGi (réseau de données). La lecture eBPF `upf_ext_stat` est effectuée sur le chemin de collecte par le `Sampler` en arrière-plan (la lecture par collecte était un déclencheur prouvé de `virtio_net` TX-ring-wedge, problème #57), et est contrôlée par le drapeau. Les mêmes champs sont également disponibles à

la demande sous forme de JSON à `/api/v1/xdp_stats`. Voir la Référence des Métriques pour plus de détails.

Planification de la Capacité:

- Surveiller les heures de débit de pointe (par exemple, heures du soir)
- Comparer à la capacité de lien (vitesses des interfaces N3/N6)
- Planifier pour 2x le débit de pointe pour une marge de manœuvre

---

Taux de Pertes

Calcul:

```
Taux de Pertes (%) = (Paquets Jetés / Total des Paquets RX) × 100
```

Seuils Acceptables:

- **< 0.1%**: Excellent (perte de paquets normale due à des erreurs)
- **0.1% - 1%**: Bon (problèmes mineurs ou limitation de taux)
- **1% - 5%**: Mauvais (enquêter sur des problèmes de QoS ou de capacité)
- **> 5%**: Critique (problème majeur de transfert ou de capacité)

Causes Courantes de Pertes:

- Limitation de taux QER (MBR dépassé)
- Échecs de recherche de carte eBPF
- TEIDs ou IPs UE invalides
- Erreurs de routage

Alertes et Seuils

Alertes Recommandées

Alertes Critiques (Réponse immédiate requise):

- Capacité de la carte eBPF > 90%
- Compte abandonné XDP > 0
- Taux de pertes > 5%
- Échec du contrôle de santé UPF

Alertes d'Avertissement (Réponse dans l'heure):

- Capacité de la carte eBPF > 70%
- Taux de pertes > 1%
- Taux de paquets approchant la capacité de lien
- TTL de tampon dépassé (paquets de plus de 30s)

Alertes Informatiques (Surveiller les tendances):

- Capacité de la carte eBPF > 50%
- Compte de paquets mis en tampon en augmentation
- Nouvelles associations PFCP établies/libérées
- Seuils de volume URR dépassés

Configuration des Alertes

Les alertes peuvent être configurées via :

1. **Métriques Prometheus**: Exporter des métriques pour une surveillance externe (voir Référence des Métriques pour la liste complète)
2. **Surveillance des Logs**: Analyser les logs d'OmniUPF pour des motifs d'erreur
3. **Interrogation de l'API REST**: Interroger périodiquement les points de terminaison `/map_info`, `/packet_stats`

4. **Surveillance de l'Interface Web**: Surveillance manuelle via les pages Statistiques et Capacité

Planification de la Capacité

Estimation de la Capacité des Sessions

Calculer le nombre maximum de sessions:

```
Max Sessions = min(
  Capacité de la Carte PDR / 2,  # PDRs descendants + montants par
session
  Capacité de la Carte FAR / 2,  # FARs descendants + montants par
session
  Capacité de la Carte QER       # Optionnel, un QER par session
)
```

Exemple:

- Capacité de la Carte PDR: 131,070
- Capacité de la Carte FAR: 131,070
- Capacité de la Carte QER: 65,535

Max Sessions = min(131,070 / 2, 131,070 / 2, 65,535) = **65,535 sessions**

Capacité Mémoire

Calculer la mémoire totale de la carte eBPF:

```
Mémoire = Σ (Capacité de la Carte × (Taille de Clé + Taille de
Valeur))
```

Exemple de Configuration:

- Cartes PDR: 3 × 131,070 × 212 B = 83.3 MB

- Carte FAR: 131,070 × 20 B = 2.6 MB
- Carte QER: 65,535 × 36 B = 2.3 MB
- Carte URR: 131,070 × 20 B = 2.6 MB
- **Total**: ~91 MB de mémoire noyau

Considérations sur la Mémoire du Noyau:

- Assurez-vous d'une limite de mémoire verrouillée suffisante (`ulimit -l`)
- Réserver 2x l'utilisation estimée pour une marge de sécurité
- Surveiller la disponibilité de la mémoire du noyau

Capacité de Trafic

Calculer la capacité de débit requise:

1. **Estimer le débit moyen par session**:
   - Streaming vidéo: ~5 Mbps
   - Navigation Web: ~1 Mbps
   - VoIP: ~0.1 Mbps
2. **Calculer le débit agrégé**:

   ```
   Débit Total = Sessions × Débit Moyen par Session
   ```

3. **Ajouter une marge de manœuvre**:

   ```
   Capacité Requise = Débit Total × 2  # 100% de marge de manœuvre
   ```

Exemple:

- 10,000 sessions simultanées
- Moyenne de 2 Mbps par session
- Total: 20 Gbps
- Capacité requise: 40 Gbps (interfaces N3 + N6)

Planification de la Croissance

Analyse des Tendances:

1. Enregistrer le nombre de sessions de pointe quotidien
2. Calculer le taux de croissance hebdomadaire
3. Extrapoler jusqu'à la limite de capacité

Formule du Taux de Croissance:

```
Semaines jusqu'à la Capacité = (Capacité - Utilisation Actuelle) /
(Croissance Hebdomadaire)
```

Exemple:

- Sessions actuelles: 30,000
- Capacité: 65,535 sessions
- Croissance hebdomadaire: 2,000 sessions
- Semaines jusqu'à la capacité: (65,535 - 30,000) / 2,000 = **17.8 semaines**

Action: Planifier une mise à niveau de capacité dans 12 semaines (laissant 5 semaines de marge).

Dépannage des Problèmes de Performance

Taux de Pertes de Paquets Élevé

Symptômes: Taux de pertes > 1%, plaintes des utilisateurs concernant une mauvaise connectivité

Diagnostic:

1. Vérifier Statistiques → Statistiques de Paquets
2. Identifier si les pertes sont spécifiques à un protocole

3. Examiner les Statistiques XDP pour les pertes XDP vs. abandons

Causes Courantes:

- **Limitation de Taux QER**: Vérifier les valeurs MBR QER par rapport au trafic réel
- **TEIDs Invalides**: Vérifier que le TEID PDR montant de l'UE correspond à l'attribution gNB
- **IPs UE Inconnues**: Vérifier que le PDR descendant existe pour l'IP de l'UE
- **Débordement de Tampon**: Vérifier les statistiques de tampon

Résolution:

- Augmenter le MBR QER si limitation de taux
- Vérifier que le SMF a créé les PDR corrects
- Vider les tampons si un débordement est détecté

Erreurs de Traitement XDP

Symptômes: XDP abandonné > 0

Diagnostic:

1. Naviguer vers Statistiques → Statistiques XDP
2. Vérifier le compteur abandonné
3. Examiner les logs d'OmniUPF pour des erreurs eBPF

Causes Courantes:

- Échec de vérification du programme eBPF
- Incompatibilité de version du noyau
- Erreurs d'accès à la carte eBPF
- Corruption de mémoire

Résolution:

- Redémarrer le service OmniUPF

- Vérifier que la version du noyau répond aux exigences minimales (Linux 5.4+)
- Examiner les logs du programme eBPF
- Contacter le support si le problème persiste

Épuisement de la Capacité

Symptômes: Échecs d'établissement de session, capacité de carte à 100%

Diagnostic:

1. Naviguer vers la page de Capacité
2. Identifier quelle carte est à 100%
3. Vérifier si les sessions sont bloquées (non supprimées)

Atténuation Immédiate:

1. Identifier les sessions obsolètes (vérifier la page des Sessions)
2. Demander au SMF de supprimer les anciennes sessions
3. Vider les tampons pour libérer des entrées FAR

Résolution à Long Terme:

1. Augmenter la capacité de la carte eBPF
2. Planifier un redémarrage de l'UPF avec des cartes plus grandes
3. Mettre en œuvre des politiques de nettoyage des sessions

Dégradation de la Performance

Symptômes: Latence élevée, faible débit, saturation CPU

Diagnostic:

1. Vérifier le taux de paquets par rapport à la base de référence historique
2. Examiner les statistiques XDP pour des retards de traitement
3. Surveiller l'utilisation CPU sur l'hôte UPF

4. Vérifier l'utilisation des interfaces N3/N6

Causes Courantes:

- Trafic dépassant la capacité de l'UPF
- Cœurs CPU insuffisants pour le traitement des paquets
- Goulot d'étranglement de l'interface réseau
- Collisions de hachage de carte eBPF

Résolution:

- Élargir l'UPF horizontalement (ajouter plus d'instances)
- Améliorer le CPU ou activer le RSS (Répartition de Charge Côté Réception)
- Améliorer les interfaces réseau à des vitesses plus élevées
- Ajuster la fonction de hachage de la carte eBPF

Documentation Connexe

- **Référence des Métriques** - Référence complète des métriques Prometheus
- **Guide des Opérations UPF** - Architecture générale et opérations de l'UPF
- **Guide de Gestion des Règles** - Configuration des PDR, FAR, QER, URR
- **Guide des Opérations de l'Interface Web** - Fonctionnalités de surveillance du panneau de contrôle
- **Guide de Dépannage** - Problèmes courants et diagnostics
- **Guide d'Architecture** - Chemin de données eBPF et optimisation des performances

Référence des Codes de Cause PFCP

Aperçu

PFCP (Packet Forwarding Control Protocol) utilise des codes de cause dans les messages de réponse pour indiquer le résultat des demandes. Ce document décrit les codes de cause implémentés dans OmniUPF et quand ils se produisent lors du traitement des messages PFCP.

Tous les codes de cause sont conformes aux spécifications **3GPP TS 129.244** et sont renvoyés dans les messages de réponse PFCP pour indiquer le succès, l'échec ou des conditions d'erreur spécifiques.

Suivi des Codes de Cause

OmniUPF suit les résultats des messages PFCP à l'aide de métriques Prometheus. Chaque réponse PFCP comprend un code de cause qui est enregistré dans :

```
upf_pfcp_rx_errors{message_name="...", cause_code="...",
peer_address="..."}
```

Cela permet de surveiller :

- **Taux de succès** par type de message et nœud de plan de contrôle
- **Modèles d'erreur** indiquant des erreurs de configuration ou des problèmes de protocole
- **Santé de l'association** basée sur les taux de rejet

Voir Référence des Métriques pour la documentation complète des métriques PFCP.

Catégories de Codes de Cause

Codes de Succès

Code	Nom	Quand Cela Se Produit
1	**RequestAccepted**	Demande traitée avec succès. Tous les IEs obligatoires présents et valides. Règles créées/modifiées/supprimées avec succès.

Codes d'Erreur Client

Code	Nom	Quand Cela Se Produit
64	**RequestRejected**	Rejet général pour des erreurs non spécifiées. Utilisé lorsque aucun code de cause spécifique ne s'applique.
65	**SessionContextNotFound**	Modification ou suppression de session demandée pour un SEID inconnu. La session spécifiée n'existe pas sur ce UPF.
66	**MandatoryIEMissing**	Élément d'information requis absent. Exemples : NodeID manquant dans la configuration de l'association, F-SEID manquant dans l'établissement de session, RecoveryTimeStamp manquant.
67	**ConditionalIEMissing**	IE conditionnellement requis manquant en fonction des autres IEs présents. Utilisé lorsque les IEs dépendent de la présence des autres.
69	**MandatoryIEIncorrect**	IE requis présent mais contenant des données invalides. Exemples : format de NodeID non analysable, valeur de

Code	Nom	Quand Cela Se Produit
		RecoveryTimeStamp invalide, F-SEID mal formé.
72	**NoEstablishedPFCPAssociation**	Opération de session tentée sans association active. Doit établir une association PFCP avant de créer des sessions.
73	**RuleCreationModificationFailure**	Erreur lors de l'application des règles PDR, FAR, QER ou URR au chemin de données eBPF. Causes possibles : capacité de la carte eBPF épuisée, paramètres de règle invalides, échec de l'allocation de ressources.

Codes d'Erreur Serveur/Ressource

Code	Nom	Quand Cela Se Produit
74	**PFCPEntityInCongestion**	UPF subissant une forte charge ou une exhaustion de ressources. Incapable de traiter temporairement les demandes.
75	**NoResourcesAvailable**	Ressources insuffisantes pour satisfaire la demande. Exemples : capacité de la carte eBPF épuisée, échec de l'allocation de mémoire, pool TEID épuisé.
77	**SystemFailure**	Erreur interne critique empêchant le traitement de la demande. Exemples : échec du programme eBPF, erreur d'interface du noyau, corruption de base de données.

Codes de Fonctionnalité Non Supportée

Code	Nom	Quand Cela Se Produit
68	**InvalidLength**	Le champ de longueur de l'IE ne correspond pas à la longueur réelle des données. Actuellement inutilisé dans OmniUPF.
70	**InvalidForwardingPolicy**	Politique de transfert non supportée par le UPF. Actuellement inutilisé dans OmniUPF.
71	**InvalidFTEIDAllocationOption**	Option d'allocation F-TEID non supportée. Actuellement inutilisé dans OmniUPF.
76	**ServiceNotSupported**	Service ou fonctionnalité demandée non implémenté. Actuellement inutilisé dans OmniUPF.
78	**RedirectionRequested**	UPF demande une redirection vers une autre instance UPF. Actuellement inutilisé dans OmniUPF.

Scénarios et Causes Courants

Échecs de Configuration d'Association

Scénario : NodeID Manquant

```
SMF → UPF : Demande de Configuration d'Association (pas de NodeID)
UPF → SMF : Réponse de Configuration d'Association (Cause :
MandatoryIEMissing)
```

Résolution : Assurez-vous que le SMF inclut l'IE NodeID dans toutes les demandes de configuration d'association.

Scénario : Format de NodeID Invalide

```
SMF → UPF : Demande de Configuration d'Association
(NodeID="invalide")
UPF → SMF : Réponse de Configuration d'Association (Cause :
MandatoryIEIncorrect)
```

Résolution : NodeID doit être un FQDN valide ou une adresse IPv4/IPv6.

Scénario : Timestamp de Récupération Manquant

```
SMF → UPF : Demande de Configuration d'Association (pas de
RecoveryTimeStamp)
UPF → SMF : Réponse de Configuration d'Association (Cause :
MandatoryIEMissing)
```

Résolution : Inclure RecoveryTimeStamp dans la demande de configuration d'association.

Échecs d'Établissement de Session

Scénario : Pas d'Association Établie

```
SMF → UPF : Demande d'Établissement de Session
UPF → SMF : Réponse d'Établissement de Session (Cause :
NoEstablishedPFCPAssociation)
```

Résolution : Établir l'association PFCP avant de créer des sessions.

Scénario : Échec de Création de Règle

```
SMF → UPF : Demande d'Établissement de Session
UPF traite les FAR, QER, URR avec succès
UPF échoue à créer PDR (carte eBPF pleine)
UPF → SMF : Réponse d'Établissement de Session (Cause :
RuleCreationModificationFailure)
```

Résolution :

- Vérifiez la capacité de la carte eBPF (voir Suivi de Capacité)
- Augmentez les tailles de carte dans la configuration UPF
- Réduisez le nombre de sessions actives

Scénario : F-SEID Manquant

```
SMF → UPF : Demande d'Établissement de Session (pas de CP F-SEID)
UPF → SMF : Réponse d'Établissement de Session (Cause :
MandatoryIEMissing)
```

Résolution : Inclure CP F-SEID dans la demande d'établissement de session.

Échecs de Modification de Session

Scénario : SEID Inconnu

```
SMF → UPF : Demande de Modification de Session (SEID=12345)
UPF n'a pas de session avec SEID 12345
UPF → SMF : Réponse de Modification de Session (Cause :
SessionContextNotFound)
```

Résolution :

- Vérifiez que le SEID correspond à la valeur de la réponse d'établissement de session
- Vérifiez si la session a déjà été supprimée
- Assurez-vous d'utiliser la bonne instance UPF (scénarios de boucle N9)

Échecs de Suppression de Session

Scénario : SEID Inconnu

```
SMF → UPF : Demande de Suppression de Session (SEID=67890)
UPF n'a pas de session avec SEID 67890
UPF → SMF : Réponse de Suppression de Session (Cause :
SessionContextNotFound)
```

Résolution : Le SEID peut avoir déjà été supprimé ou n'a jamais existé.

Dépannage avec les Codes de Cause

Utilisation des Métriques Prometheus

Interrogez Prometheus pour identifier les modèles d'erreur :

```
# Taux d'erreur par code de cause
rate(upf_pfcp_rx_errors{cause_code!="RequestAccepted"}[5m])

# Principales causes de rejet
topk(5, sum by (cause_code) (upf_pfcp_rx_errors))

# Erreurs par pair SMF
sum by (peer_address, cause_code)
(upf_pfcp_rx_errors{cause_code!="RequestAccepted"})

# Échecs d'établissement de session
upf_pfcp_rx_errors{message_name="SessionEstablishmentRequest",
cause_code!="RequestAccepted"}
```

Utilisation de l'Interface Web

Naviguez vers la page **Sessions** pour voir :

- Nombre de sessions actives par nœud de plan de contrôle
- Taux de succès/échec d'établissement de session
- Erreurs récentes de session

Naviguez vers la page **Capacité** pour diagnostiquer :

- Utilisation de la carte eBPF (cause racine de RuleCreationModificationFailure)
- Indicateurs d'épuisement des ressources

Voir Guide de l'Interface Web pour des instructions détaillées de suivi.

Étapes de Débogage Courantes

Taux Élevé de MandatoryIEMissing :

1. Vérifiez la configuration SMF pour les IEs requis
2. Vérifiez la compatibilité de la version de la bibliothèque PFCP
3. Examinez les journaux SMF pour les erreurs de construction d'IE

Échecs Fréquents de RuleCreationModificationFailure :

1. Vérifiez la capacité de la carte eBPF : `GET /api/v1/map_info`
2. Surveillez l'utilisation de la carte : `upf_ebpf_map_used / upf_ebpf_map_capacity`
3. Augmentez les tailles de carte dans la configuration si > 70 % utilisé
4. Voir Planification de Capacité

Erreurs NoEstablishedPFCPAssociation :

1. Vérifiez que l'association existe : `GET /api/v1/pfcp_associations`
2. Vérifiez la configuration du délai d'attente du heartbeat
3. Examinez les journaux de configuration de l'association
4. Assurez-vous que le SMF et le UPF peuvent se joindre

SessionContextNotFound lors de la Modification :

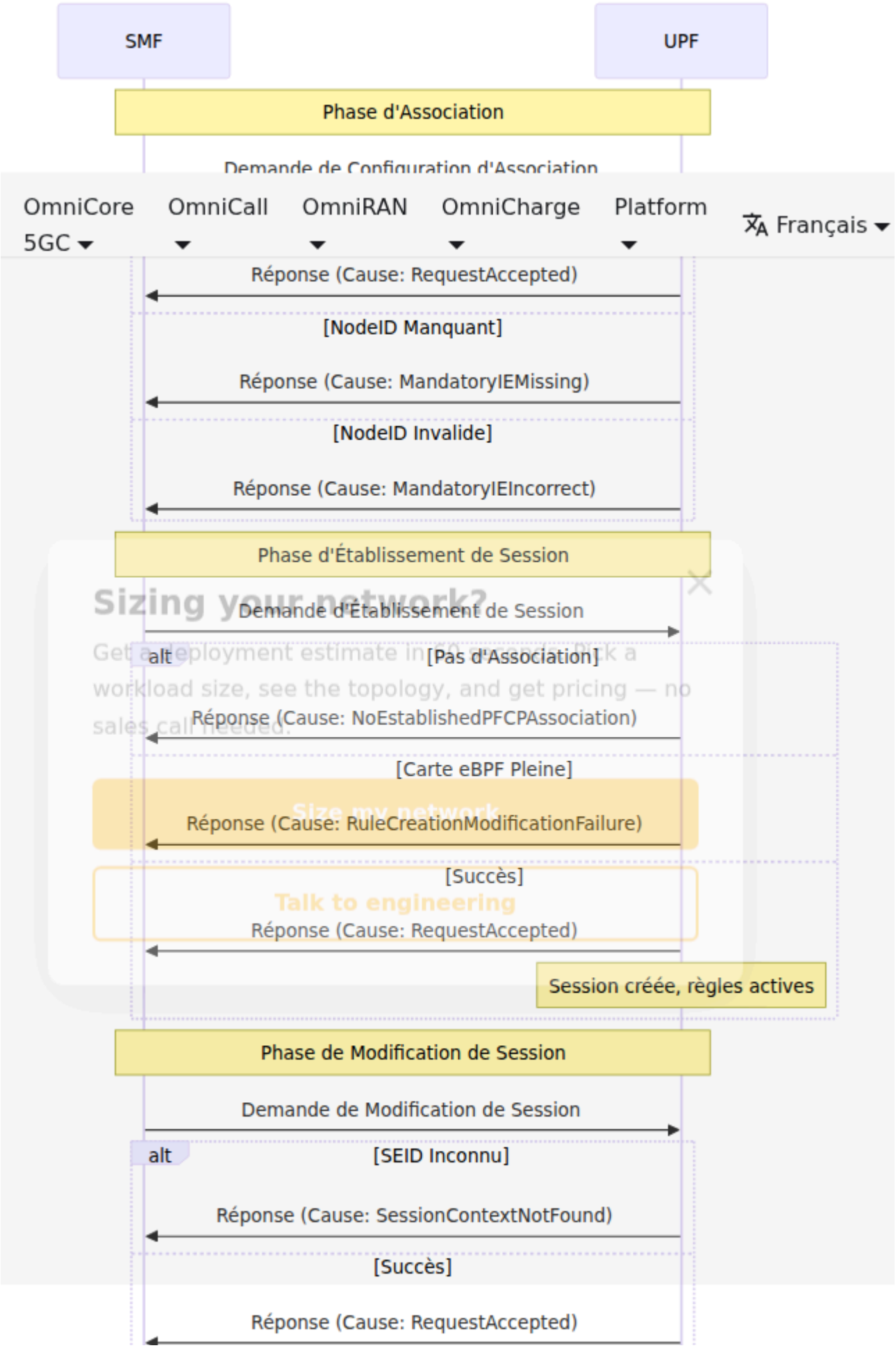
1. Vérifiez le SEID de la réponse d'établissement de session

2. Vérifiez si la session a été supprimée
3. Pour la boucle N9 : Assurez-vous d'utiliser le bon point de terminaison UPF
4. Interrogez les sessions actives : `GET /api/v1/pfcp_sessions`

Impact des Codes de Cause sur les

Opérations

Cycle de Vie de la Session



Sizing your network?
Get a deployment estimate in 60 seconds. Pick a workload size, see the topology, and get pricing — no sales call needed.
Size my network
Talk to engineering

Métriques et Alertes

Alertes Recommandées :

```
# Critique : Taux de rejet élevé
- alert: PfcpHighRejectionRate
  expr: |
    rate(upf_pfcp_rx_errors{cause_code!="RequestAccepted"}[5m]) > 0.1
  annotations:
    summary: "Taux de rejet PFCP élevé : {{ $value }}/s"

# Avertissement : Problèmes de capacité
- alert: PfcpRuleCreationFailures
  expr: |

rate(upf_pfcp_rx_errors{cause_code="RuleCreationModificationFailure"}
[5m]) > 0
  annotations:
    summary: "Échecs de création de règles PFCP détectés"

# Avertissement : Problèmes d'association
- alert: PfcpNoAssociation
  expr: |

rate(upf_pfcp_rx_errors{cause_code="NoEstablishedPFCPAssociation"}
[5m]) > 0
  annotations:
    summary: "Sessions PFCP tentées sans association"
```

Conformité aux Normes 3GPP

OmniUPF implémente des codes de cause conformément à :

- **3GPP TS 129.244 v16.4.0** - spécification PFCP
- **Section 8.2.1** - définition de l'IE de cause

- **Section 8.19** - tableau des valeurs de cause

Documentation Connexe

- **Intégration du Protocole PFCP** - architecture PFCP et gestion des messages
- **Référence des Métriques** - documentation de la métrique upf_pfcp_rx_errors
- **Guide de Suivi** - suivi de capacité et alertes
- **Guide de Dépannage** - problèmes d'association et de session PFCP
- **Guide de l'Interface Web** - suivi des sessions et des associations

Guide de gestion des règles

Table des matières

1. Aperçu
2. Règles de détection de paquets (PDR)
3. Règles d'action de transfert (FAR)
4. Règles d'application de la QoS (QER)
5. Règles de reporting d'utilisation (URR)
6. Relations entre les règles
7. Opérations courantes
8. Dépannage

Aperçu

OmniUPF utilise un ensemble de règles interconnectées pour classer, transférer, modeler et suivre le trafic du plan utilisateur. Ces règles sont installées par le SMF via PFCP et stockées dans des cartes eBPF pour un traitement de paquets à haute performance. Comprendre ces règles et leurs relations est essentiel pour faire fonctionner et dépanner le UPF.

Types de règles

Type de règle	Objectif	Champ clé	Installé par
PDR (Règle de détection de paquets)	Classer les paquets en flux	TEID ou IP UE	SMF via établissement/modification de session PFCP
FAR (Règle d'action de transfert)	Déterminer l'action de transfert	ID FAR	SMF via établissement/modification de session PFCP
QER (Règle d'application de la QoS)	Appliquer des limites de bande passante et un marquage	ID QER	SMF via établissement/modification de session PFCP
URR (Règle de reporting d'utilisation)	Suivre les volumes de données pour la facturation	ID URR	SMF via établissement/modification de session PFCP

Flux de traitement des règles

Règles de détection de paquets (PDR)

Objectif

Les PDR classifient les paquets entrants en flux de trafic. Ils sont le point d'entrée pour tout traitement de paquets dans le UPF.

Structure PDR

PDR descendant

Clé : Adresse IP UE
IPv4 ou IPv6

ID FAR
ID QER
IDs URR
Mode SDF
Filtres SDF

PDR montant

Clé : TEID
entier 32 bits

ID FAR
ID QER
IDs URR
Suppression de l'en-tête
externe

PDR montant

Les PDR montant correspondent aux paquets arrivant sur l'interface N3 depuis le RAN.

Champ clé : TEID (Identifiant de point de terminaison de tunnel)

- Entier non signé de 32 bits
- Assigné par le SMF et signalé au gNB
- Unique par flux de trafic UE

Champs de valeur :

- **ID FAR** : Référence à la règle d'action de transfert
- **ID QER** : Référence à la règle d'application de la QoS (optionnel)
- **IDs URR** : Liste des règles de reporting d'utilisation (optionnel)
- **Suppression de l'en-tête externe** : Indicateur pour retirer l'encapsulation GTP-U

Processus de recherche :

1. Extraire le TEID de l'en-tête GTP-U
2. Recherche de hachage dans la carte eBPF `uplink_pdr_map`
3. Si une correspondance est trouvée, récupérer l'ID FAR, l'ID QER et les IDs URR
4. Si aucune correspondance, supprimer le paquet

Exemple :

```
TEID : 5678
ID FAR : 2
ID QER : 1
Suppression de l'en-tête externe : Faux
Mode SDF : Pas de SDF
```

PDR descendant

Les PDR descendant correspondent aux paquets arrivant sur l'interface N6 depuis le réseau de données.

Champ clé : Adresse IP UE

- Adresse IPv4 (32 bits) ou adresse IPv6 (128 bits)
- Assignée par le SMF lors de l'établissement de la session PDU
- Unique par UE

Champs de valeur :

- **ID FAR** : Référence à la règle d'action de transfert
- **ID QER** : Référence à la règle d'application de la QoS (optionnel)
- **IDs URR** : Liste des règles de reporting d'utilisation (optionnel)
- **Mode SDF** : Mode de filtre de flux de données de service
  - `Pas de SDF` : Pas de filtrage, tout le trafic correspond

    - `SDF uniquement` : Seul le trafic correspondant au SDF est transféré
    - `SDF + Par défaut` : Le trafic correspondant au SDF utilise des règles spécifiques, l'autre trafic utilise le FAR par défaut
- **Filtres SDF** : Filtres spécifiques à l'application (ports, protocoles, plages IP)

Processus de recherche :

1. Extraire l'IP de destination de l'en-tête du paquet
2. Recherche de hachage dans `downlink_pdr_map` (IPv4) ou `downlink_pdr_map_ip6` (IPv6)
3. Si une correspondance est trouvée, vérifier les filtres SDF (si configurés)
4. Récupérer l'ID FAR, l'ID QER et les IDs URR
5. Si aucune correspondance, supprimer le paquet

Exemple :

```
IP UE : 10.45.0.1
ID FAR : 1
ID QER : 1
Suppression de l'en-tête externe : Faux
Mode SDF : Pas de SDF
```

Filtres SDF (Flux de données de service)

Les filtres SDF fournissent une classification de trafic spécifique à l'application au sein d'un PDR.

Cas d'utilisation :

- Différencier le trafic YouTube de la navigation web
- Appliquer une QoS différente pour VoIP par rapport aux données à meilleur effort
- Acheminer des applications spécifiques à travers différents chemins réseau

Critères de filtre :

- **Protocole** : TCP, UDP, ICMP
- **Plage de ports** : Ports de destination (par exemple, 443 pour HTTPS, 5060 pour SIP)
- **Plage d'adresses IP** : Réseaux de destination spécifiques
- **Description de flux** : Modèles de flux définis par la 3GPP

Exemple de configuration SDF :

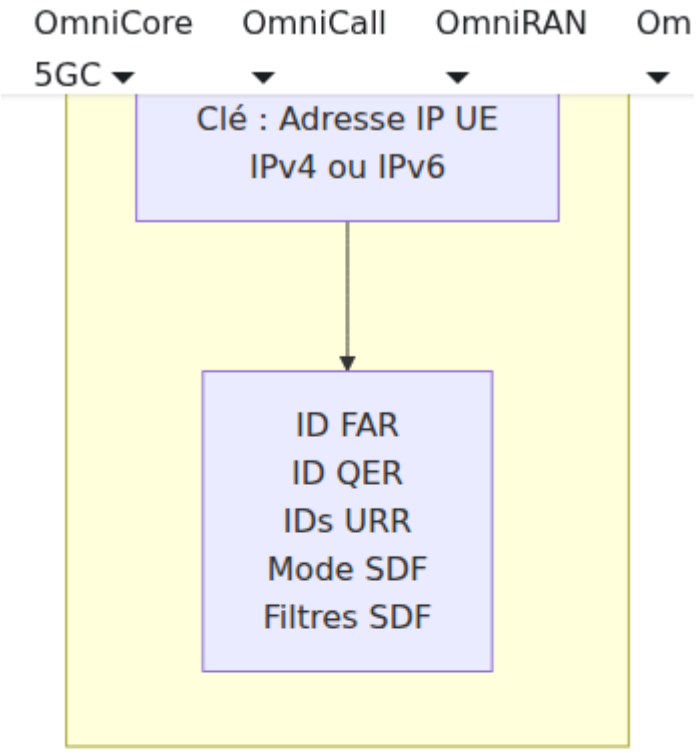
```
ID PDR : 10
IP UE : 10.45.0.1
Mode SDF : SDF uniquement
Filtres SDF :
  - Protocole : UDP, Ports : 5060-5061 → ID FAR 5 (FAR VoIP)
  - Protocole : TCP, Port : 443 → ID FAR 1 (FAR par défaut)
```

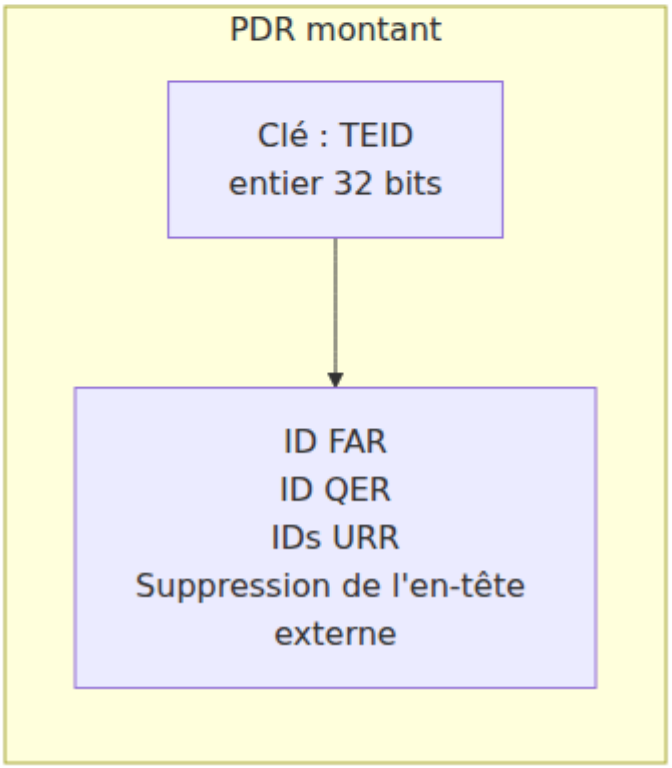
Règles d'action de transfert (FAR)

Objectif

Les FAR déterminent quoi faire avec les paquets qui correspondent à un PDR. Ils définissent les actions de transfert, les paramètres d'encapsulation GTP-U et les points de terminaison de destination.

Structure FAR





Indicateurs d'action

Les actions FAR sont des indicateurs de bits qui peuvent être combinés :

Indicateur	Bit	Valeur	Description
TRANSFÉRER	1	2	Transférer le paquet à la destination
TAMPONNER	2	4	Stocker le paquet dans le tampon
SUPPRIMER	0	1	Jeter le paquet
NOTIFIER	3	8	Envoyer une notification au plan de contrôle
DUPLIQUER	4	16	Dupliquer le paquet vers plusieurs destinations

Combinaisons d'actions courantes :

- `Action : 2 (TRANSFÉRER)` - Transfert normal (le plus courant)
- `Action : 6 (TRANSFÉRER + TAMPONNER)` - Transférer et tamponner pendant le transfert
- `Action : 4 (TAMPONNER)` - Tamponner uniquement (pendant le changement de chemin)
- `Action : 1 (SUPPRIMER)` - Supprimer le paquet (rare, généralement pour l'application de politiques)

Contrôle de la mise en mémoire tampon

L'indicateur BUFFER (bit 2) contrôle la mise en mémoire tampon des paquets pendant les événements de mobilité. La mise en mémoire tampon est une fonctionnalité critique du UPF qui empêche la perte de paquets pendant les transitions d'état de l'UE.

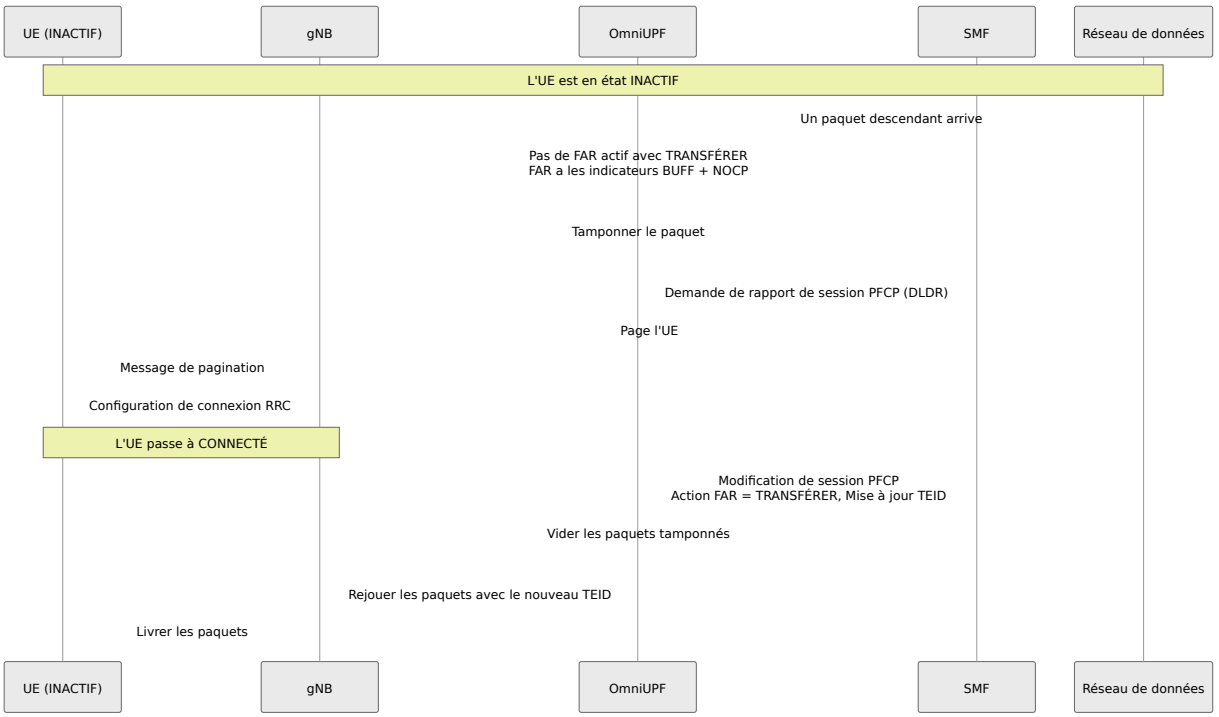
Quand la mise en mémoire tampon est utilisée

Transition de l'état inactif à connecté : Lorsque des paquets descendants arrivent pour un UE en état INACTIF (non connecté au gNB), le UPF :

1. Tamponne les paquets
2. Envoie une notification de données descendantes (DLDR) au SMF
3. Le SMF page l'UE pour le réveiller et se connecter
4. Une fois connecté, le SMF met à jour le FAR avec l'action TRANSFÉRER
5. Le UPF vide les paquets tamponnés vers l'UE

Transfert (Connecté à Connecté) : Lors d'un transfert de gNB à gNB, le UPF tamponne temporairement les paquets pour éviter la perte :

1. La connexion gNB ancienne est interrompue
2. Le SMF définit l'action FAR sur TAMPONNER
3. Les paquets sont mis en file d'attente pendant le changement de chemin
4. L'UE se connecte au nouveau gNB
5. Le SMF met à jour le FAR avec le nouveau TEID et l'action TRANSFÉRER
6. Le UPF vide les paquets vers le nouveau gNB



Capacité et limites de mise en mémoire tampon

Limites globales de mise en mémoire tampon :

- **Max Total de Paquets** : 100,000 (configurable)
- **Max Total de Bytes** : Basé sur la mémoire disponible
- **TTL (Temps de vie)** : 60 secondes (configurable)
- **Paquets dépassant le TTL** : Supprimés automatiquement

Limites par FAR :

- **Max Paquets par FAR** : 10,000 (configurable)
- **Objectif** : Empêcher un seul FAR d'épuiser la capacité de mise en mémoire tampon

Comportement de débordement de tampon :

- Lorsque la limite globale ou par FAR est atteinte, les nouveaux paquets sont supprimés
- Les métriques suivent les suppressions avec `reason="global_limit"` ou `reason="far_limit"`
- Les paquets les plus anciens ne sont PAS automatiquement évincés (suppression explicite uniquement à l'expiration du TTL)

Notification de données descendantes (DLDR)

Lorsque le UPF tamponne un paquet pour un UE INACTIF, il envoie une demande de rapport de session PFCP au SMF :

Contenu de DLDR :

- **Type de rapport** : Rapport de données descendantes (DLDR)
- **ID FAR** : Le FAR qui a déclenché la mise en mémoire tampon
- **Informations sur le service de données descendantes** : QFI optionnel, Indicateur de politique de pagination

Actions du SMF sur DLDR :

1. Page l'UE via AMF → gNB
2. Attend que l'UE établisse la connexion RRC
3. Envoie une demande de modification de session PFCP pour mettre à jour le FAR

4. L'action FAR change de `BUFF+NOCP` à `FORW`
5. Le UPF vide les paquets tamponnés

Métriques pour DLDR :

- `upf_dldr_sent_total` : Total des DLDR envoyés
- `upf_dldr_send_errors` : DLDR échoués
- `upf_buffer_notify_to_flush_duration_seconds` : Latence entre DLDR et vidage

Voir Référence des métriques pour la liste complète.

Opérations de mise en mémoire tampon

Activer la mise en mémoire tampon (Définir l'indicateur BUFF) :

- Action FAR `|= 0x04` (définir le bit 2)
- Exemple : `Action : 2 (FORW)` → `Action : 6 (FORW+BUFF)`
- Utilisé lors de la préparation du transfert

Mode uniquement tampon (BUFF sans FORW) :

- Action FAR `= 0x04` (BUFF uniquement)
- Les paquets sont tamponnés mais NE SONT PAS transférés
- Utilisé pour l'état INACTIF de l'UE (en attente de pagination)

Désactiver la mise en mémoire tampon (Effacer l'indicateur BUFF) :

- Action FAR `&= ~0x04` (effacer le bit 2)
- Exemple : `Action : 6 (FORW+BUFF)` → `Action : 2 (FORW)`
- Les paquets tamponnés restent jusqu'à ce qu'ils soient vidés ou effacés

Vider le tampon :

- Rejouer tous les paquets tamponnés en utilisant les règles FAR **actuelles**
- Les paquets sont transférés avec le TEID/destination mis à jour
- Le tampon est vidé après un vidage réussi
- Le FAR doit avoir l'action FORW définie

Effacer le tampon :

- Jeter tous les paquets tamponnés sans transfert
- Utilisé lorsque le transfert échoue ou que la session est supprimée
- Les métriques suivent avec `reason="cleared"`

Surveillance des paquets tamponnés

Page des tampons (Interface Web) : Naviguez vers **Tampons** pour voir :

- Total des paquets tamponnés
- Total des bytes tamponnés
- Nombre de FAR avec des paquets tamponnés
- Comptages de paquets par FAR
- Horodatage du paquet le plus ancien
- Activer/Désactiver la mise en mémoire tampon par FAR
- Opérations de vidage ou d'effacement

Indicateurs clés :

- **Paquets > 10 secondes** : Délai de pagination potentiel
- **Paquets > 30 secondes** : Échec probable de la pagination, vider le tampon
- **Nombre élevé de paquets** : Vérifier les sessions bloquées ou les échecs de pagination

Métriques Prometheus :

- `upf_buffer_packets_current` : Paquets tamponnés actuels
- `upf_buffer_bytes_current` : Bytes tamponnés actuels
- `upf_buffer_fars_active` : FAR avec des paquets tamponnés
- `upf_buffer_packets_dropped{reason}` : Comptages de paquets supprimés

Voir Référence des métriques pour la liste complète des métriques de tampon.

Scénarios courants de mise en mémoire tampon

Scénario 1 : UE INACTIF Données descendantes

```
État initial :
- UE en mode INACTIF (pas de connexion gNB)
- Action FAR : 0x04 (BUFF uniquement)

Arrivée de données :
1. DN envoie un paquet descendant
2. UPF correspond au PDR, applique le FAR
3. Le FAR a l'indicateur BUFF → paquet tamponné
4. UPF envoie DLDR au SMF
5. SMF page l'UE
6. L'UE se connecte au gNB
7. SMF modifie le FAR : Action = 0x02 (FORW)
8. UPF vide les paquets tamponnés avec le nouveau TEID
```

Scénario 2 : Préparation au transfert

```
État initial :
- UE connecté à gNB-1 (TEID 1234)
- Action FAR : 0x02 (FORW)

Processus de transfert :
1. SMF modifie le FAR : Action = 0x06 (FORW+BUFF)
2. Paquets transférés à gNB-1 ET tamponnés
3. L'UE passe à gNB-2
4. SMF modifie le FAR : TEID = 5678, Action = 0x02 (FORW)
5. UPF vide les paquets tamponnés vers gNB-2 avec le nouveau TEID
6. Pas de perte de paquets pendant le transfert
```

Scénario 3 : Changement de chemin

```
État initial :
- UE connecté, flux de données actif

Changement de chemin :
1. SMF modifie le FAR : Action = 0x04 (BUFF uniquement)
2. Tous les paquets entrants sont tamponnés (non transférés)
3. Le réseau reconfigure le chemin
4. SMF modifie le FAR : Action = 0x02 (FORW), nouvelle destination
5. UPF vide tous les paquets tamponnés vers le nouveau chemin
```

Création d'en-tête externe

Détermine si l'encapsulation GTP-U doit être ajoutée.

FAR montant (N3 → N6) :

- Création d'en-tête externe : Faux
- Action : Retirer GTP-U, transférer le paquet IP natif

FAR descendant (N6 → N3) :

- Création d'en-tête externe : Vrai
- IP distante : adresse IP gNB (par exemple, 198.51.100.10)
- TEID : ID de tunnel pour le trafic UE
- Action : Ajouter l'en-tête GTP-U, transférer au gNB

Recherche FAR dans l'interface Web

La page de gestion des règles fournit une recherche FAR par ID :

Étapes :

1. Naviguez vers Règles → onglet FARs
2. Entrez l'ID FAR dans le champ de recherche
3. Cliquez sur "Rechercher" pour voir les détails du FAR

Informations affichées :

- ID FAR
- Action (numérique + indicateurs décodés)
- Statut de mise en mémoire tampon (ACTIVÉ/DÉSACTIVÉ)
- Création d'en-tête externe
- Adresse IP distante (avec représentation entière)
- TEID
- Marquage de niveau de transport

Règles d'application de la QoS (QER)

Objectif

Les QER appliquent des paramètres de qualité de service aux flux de trafic, y compris des limites de bande passante et un marquage des paquets.

Structure QER

Paramètres QER

QFI
Identifiant de flux QoS

Statut de la porte UL
Ouvert/Fermé

Statut de la porte DL
Ouvert/Fermé

ID QER
Identifiant unique

MBR Montant
Débit maximal

MBR Descendant
Débit maximal

GBR Montant
Débit garanti

GBR Descendant
Débit garanti

Paramètres QoS

QFI (Identifiant de flux QoS) :

- Identifiant de 6 bits pour les flux QoS 5G
- Les valeurs 1-9 sont standardisées (par exemple, QFI 9 = porteuse par défaut)
- Utilisé pour le marquage des paquets dans 5GC

Statut de la porte :

- **Ouvert (0)** : Trafic autorisé
- **Fermé (non zéro)** : Trafic bloqué

Débit maximal (MBR) :

- Bande passante maximale autorisée pour le flux de trafic
- Spécifié en kbps
- **MBR = 0** : Pas de limite de débit (illimité)
- Le trafic dépassant le MBR est supprimé

Débit garanti (GBR) :

- Bande passante minimale garantie pour le flux de trafic
- Spécifié en kbps
- **GBR = 0** : Meilleur effort (pas de garantie)
- **GBR > 0** : Flux priorisé avec bande passante garantie

Types de flux QoS

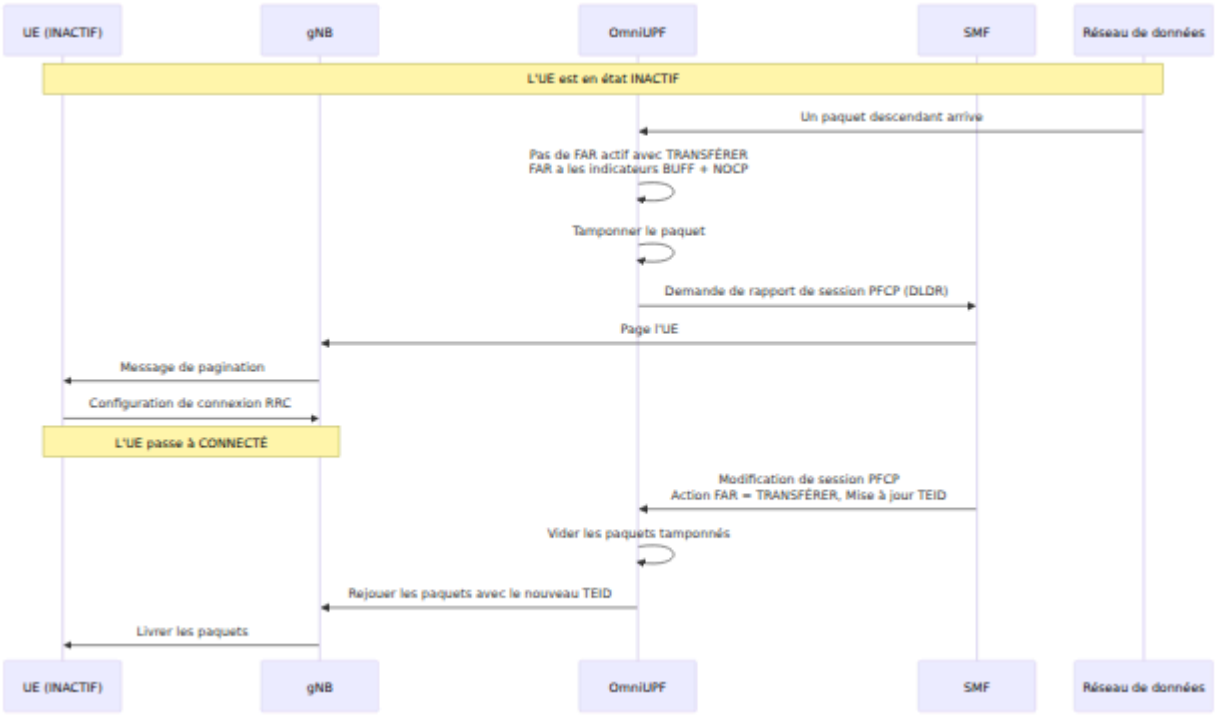
Flux à meilleur effort (GBR = 0) :

```
ID QER : 1
QFI : 9
MBR Montant : 100000 kbps (100 Mbps)
MBR Descendant : 100000 kbps (100 Mbps)
GBR Montant : 0 kbps
GBR Descendant : 0 kbps
```

Flux garantis (GBR > 0) :

```
ID QER : 2
QFI : 1
MBR Montant : 10000 kbps (10 Mbps)
MBR Descendant : 10000 kbps (10 Mbps)
GBR Montant : 5000 kbps (5 Mbps)
GBR Descendant : 5000 kbps (5 Mbps)
```

Algorithme d'application de la QoS



Mécanisme d'application du MBR

OmniUPF applique les limites MBR (Débit maximal) à l'aide d'un **limiteur de débit à fenêtre glissante** mis en œuvre dans le chemin de données eBPF. Cet algorithme fonctionne avec une précision nanoseconde directement dans la couche XDP, garantissant des performances à débit ligne sans changements de contexte du noyau.

Comment ça fonctionne

Algorithme : Limitation de débit à fenêtre glissante

Pour chaque paquet, le UPF effectue les vérifications suivantes :

1. **Vérification du statut de la porte** : Si le statut de la porte est `FERMÉ` (non zéro), supprimer immédiatement le paquet
2. **Vérification du MBR** : Si MBR = 0, contourner la limitation de débit (bande passante illimitée)
3. **Calcul du temps de transmission** :

```
tx_time = (packet_size_bytes × 8) × (1,000,000,000 ns/sec) /
MBR_kbps
```

4. **Vérification de la fenêtre** : Si le temps actuel est dans la fenêtre glissante de 5 ms, supprimer le paquet
5. **Avancer la fenêtre** : Si le paquet est autorisé, avancer la fenêtre par `tx_time`

Exemple de calcul :

Supposons :

- MBR = 100,000 kbps (100 Mbps)
- Taille de paquet = 1500 bytes
- Taille de fenêtre = 5,000,000 ns (5 ms)

```
Étape 1 : Calculer le temps de transmission à 100 Mbps
tx_time = (1500 bytes × 8 bits/byte) × (1,000,000,000 ns/sec) /
100,000,000 bps
        = 12,000,000,000 / 100,000,000
        = 120 ns

Étape 2 : Vérifier si le paquet s'inscrit dans la fenêtre
current_time = 1000000000 ns
window_start = 999990000 ns
if (window_start + tx_time > current_time):
    SUPPRIMER le paquet (dépasserait la limite de débit)

Étape 3 : Si autorisé, avancer la fenêtre
window_start = window_start + 120 ns
PASSER le paquet
```

Comportement de la fenêtre glissante

Taille de fenêtre de 5 ms :

- L'algorithme utilise une fenêtre glissante de 5 millisecondes
- La fenêtre se réinitialise automatiquement si elle est inactive pendant plus de 5 ms

- Empêche la famine de rafales tout en appliquant le débit moyen

Gestion des rafales :

- De petites rafales sont autorisées dans la fenêtre de 5 ms
- Le trafic soutenu au-dessus du MBR est limité
- Plus précis que les algorithmes de seau à jetons simples

Limitation de débit par direction :

- Le MBR montant utilise l'horodatage `qer->ul_start`
- Le MBR descendant utilise l'horodatage `qer->dl_start`
- Chaque direction est limitée indépendamment

Points d'application de la limitation de débit

Montant (N3 → N6) :

1. Le paquet arrive sur l'interface N3 (du gNB)
2. Recherche de PDR par TEID
3. Recherche de QER par ID QER
4. Vérifier `ul_gate_status` → supprimer si fermé
5. Appliquer `limit_rate_sliding_window()` avec `ul_maximum_bitrate`
6. Si réussi, transférer à N6 et mettre à jour les compteurs URR

Descendant (N6 → N3) :

1. Le paquet arrive sur l'interface N6 (du réseau de données)
2. Recherche de PDR par adresse IP UE
3. Recherche de QER par ID QER
4. Vérifier `dl_gate_status` → supprimer si fermé
5. Appliquer `limit_rate_sliding_window()` avec `dl_maximum_bitrate`
6. Si réussi, ajouter l'en-tête GTP-U et transférer à N3

Boucle N9 (SGWU ↔ PGWU) :

- Les QER montant et descendant peuvent s'appliquer dans des scénarios de boucle N9

- Chaque QER est vérifiée indépendamment aux frontières SGWU et PGWU

MBR vs. Débit observé

Pourquoi le débit observé peut différer du MBR :

- **Surcharge de protocole** : Les en-têtes GTP-U, UDP, IP ajoutent ~50-60 bytes par paquet
- **Variance de taille de paquet** : Paquets plus petits = plus de surcharge, moins d'efficacité
- **Précision de la limitation de débit** : L'application se fait par paquet, pas par octet
- **Comportement de réinitialisation de la fenêtre** : Les périodes d'inactivité de 5 ms permettent des rafales brèves au-dessus du MBR

Exemple :

```
MBR configuré : 100 Mbps
Débit observé : ~95-98 Mbps (en raison de la surcharge GTP-
U/UDP/IP)
```

Comment vérifier la limitation de débit :

1. Vérifiez les compteurs de volume URR au fil du temps : `upf_urr_*_volume_bytes`
2. Calculez le débit : `(volume_delta_bytes × 8) / time_delta_seconds / 1000` = kbps
3. Comparez avec le MBR configuré dans le QER

GBR (Débit garanti)

Important : OmniUPF ne fait **pas** actuellement respecter les minimums GBR. Les valeurs GBR sont stockées dans le QER mais ne sont pas utilisées pour la priorisation du trafic ou le contrôle d'admission.

Comportement GBR :

- Les valeurs GBR sont acceptées du SMF via PFCP

- GBR est stocké dans la carte QER et visible via l'API
- **Aucune réservation de bande passante** ou priorisation du trafic basée sur GBR
- GBR sert de métadonnées pour le suivi du type de flux (meilleur effort vs garanti)

Amélioration future :

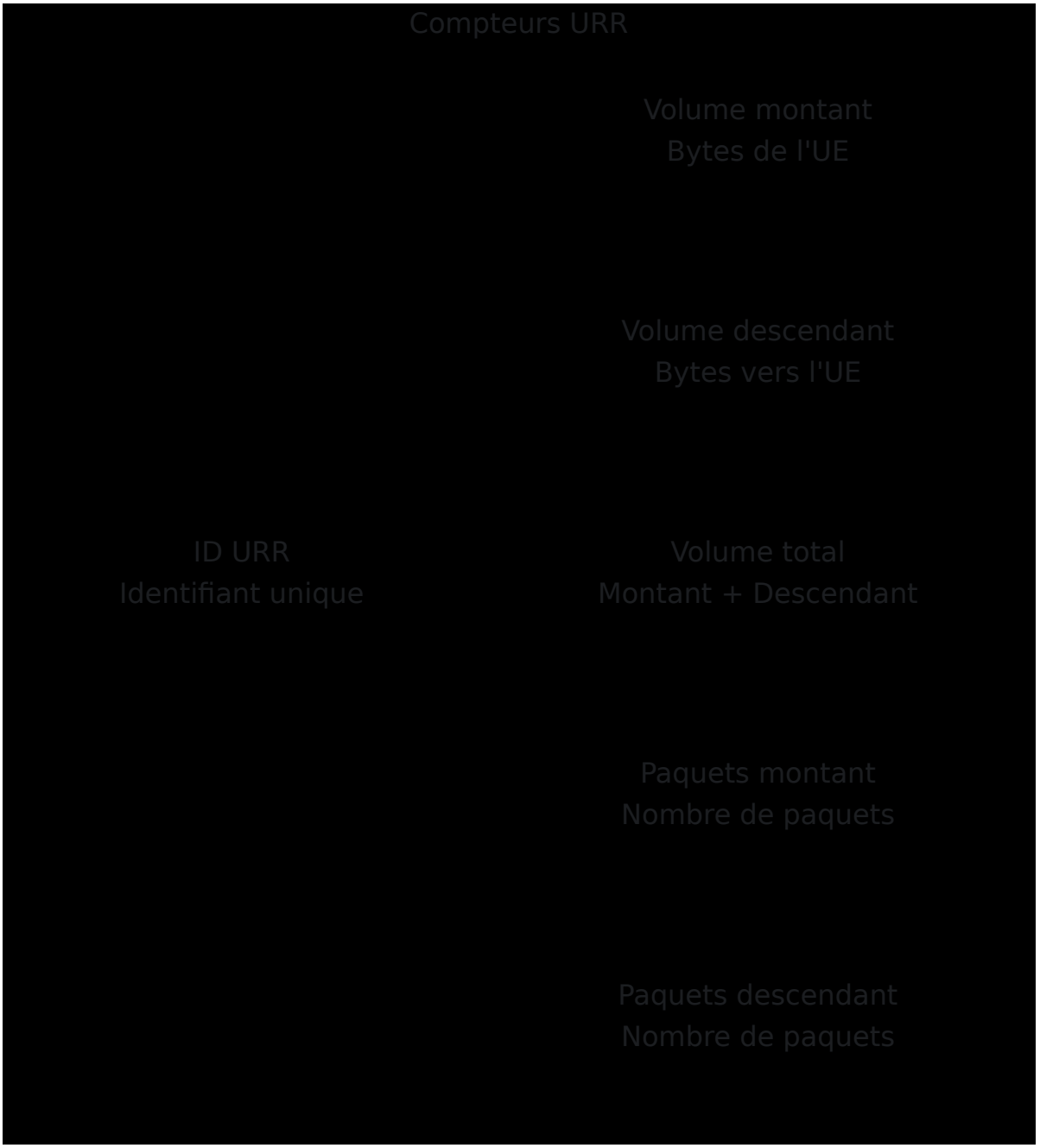
- L'application de GBR nécessite une planification du trafic ou un ordonnancement pondéré
- Peut être mise en œuvre à l'aide des capacités QoS eBPF dans les futures versions

Règles de reporting d'utilisation (URR)

Objectif

Les URR suivent les volumes de données pour la facturation, l'analyse et l'application des politiques. Ils maintiennent des compteurs de paquets et de bytes qui sont rapportés au SMF pour les enregistrements de facturation.

Structure URR



Suivi des volumes

Volume montant :

- Bytes transmis de l'UE vers le réseau de données
- Mesuré après la décapsulation GTP-U
- Inclut l'en-tête IP et la charge utile

Volume descendant :

- Bytes transmis du réseau de données vers l'UE
- Mesuré avant l'encapsulation GTP-U
- Inclut l'en-tête IP et la charge utile

Volume total :

- Somme des volumes montant et descendant
- Utilisé pour le reporting d'utilisation total

Déclencheurs de reporting d'utilisation

Les URR peuvent déclencher des rapports basés sur :

Seuil de volume :

- Rapport lorsque le volume dépasse la limite configurée
- Exemple : Rapport chaque 1 Go d'utilisation

Seuil temporel :

- Rapport à des intervalles périodiques
- Exemple : Rapport toutes les 5 minutes

Basé sur des événements :

- Rapport à la fin de session
- Rapport sur changement de QoS
- Rapport sur transfert

Format d'affichage des volumes

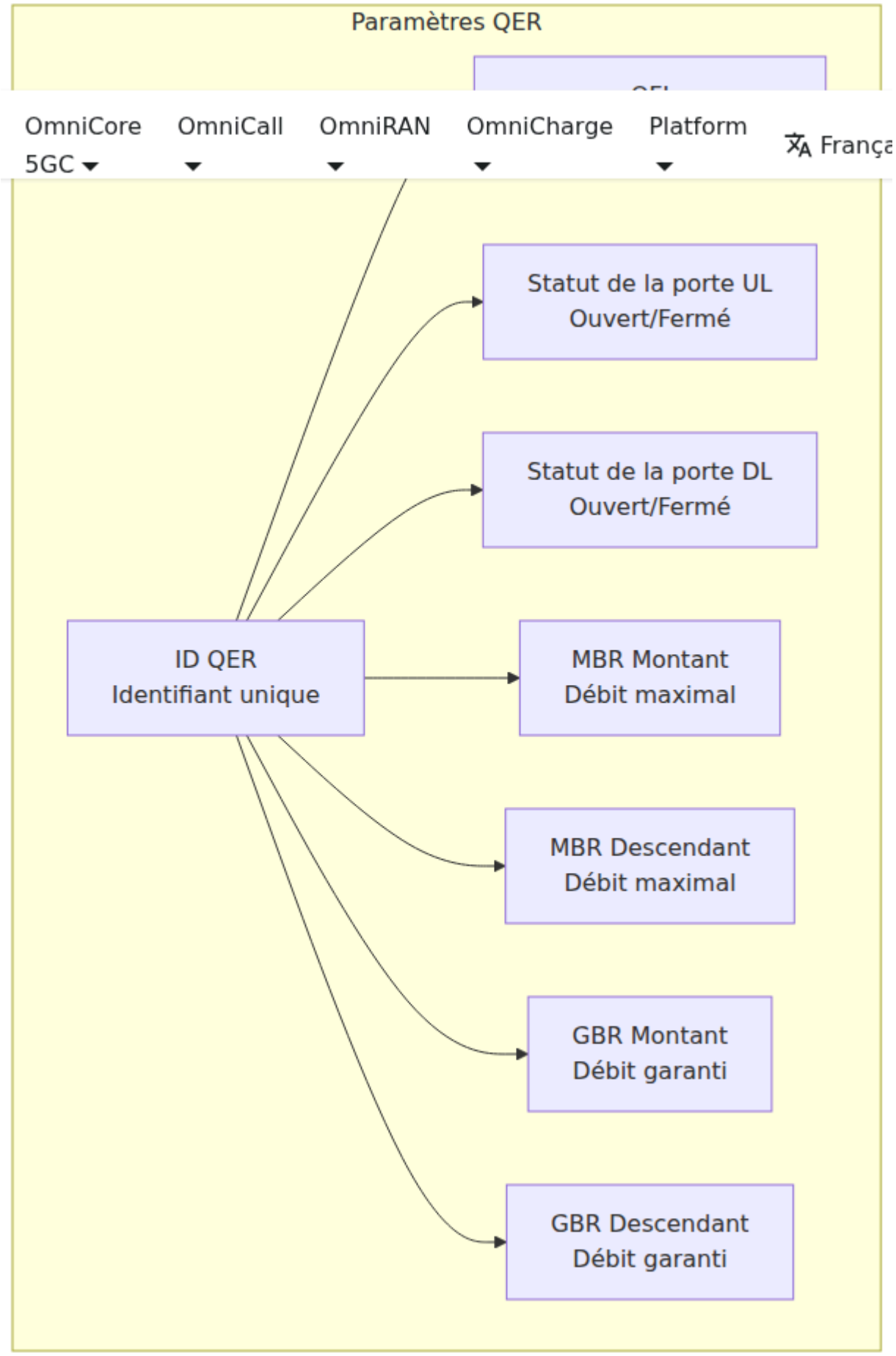
L'interface Web formate automatiquement le volume en unités lisibles par l'homme :

Bytes	Affichage
0 - 1023	B (Bytes)
1024 - 1048575	KB (Kilooctets)
1048576 - 1073741823	MB (Mégaoctets)
1073741824 - 1099511627775	GB (Gigaoctets)
1099511627776+	TB (Téraoctets)

Exemple :

```
ID URR : 0
Volume montant : 12.3 KB
Volume descendant : 9.0 KB
Volume total : 21.3 KB
```

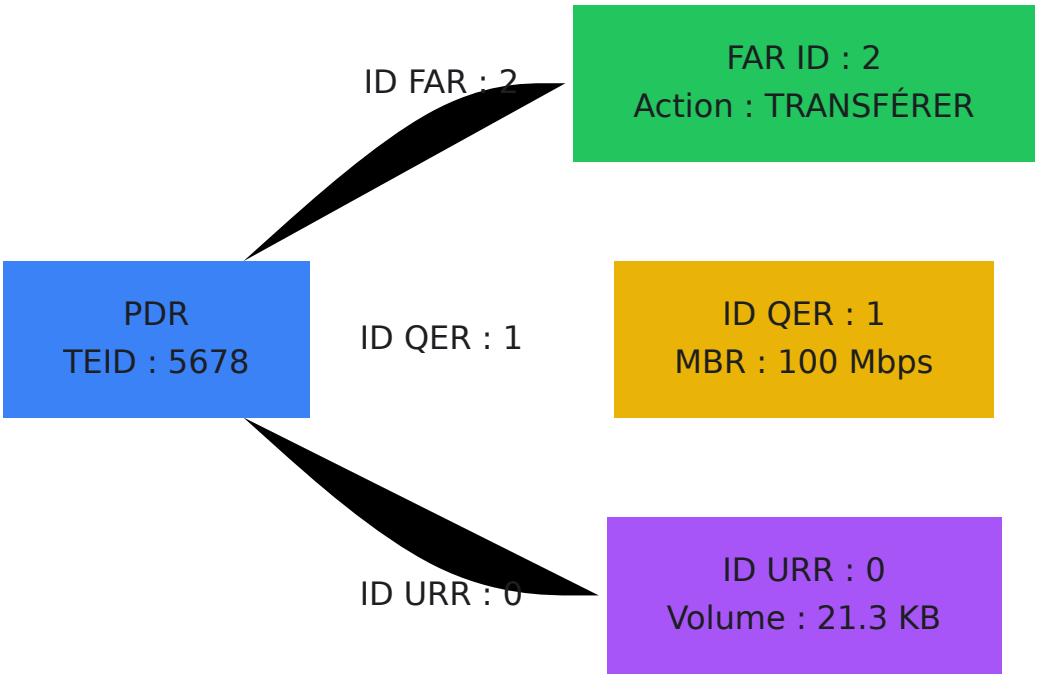
Flux de reporting URR


Paramètres QER
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Franç
Statut de la porte UL
Ouvert/Fermé
Statut de la porte DL
Ouvert/Fermé
ID QER
Identifiant unique
MBR Montant
Débit maximal
MBR Descendant
Débit maximal
GBR Montant
Débit garanti
GBR Descendant
Débit garanti

Relations entre les règles

Chaîne PDR → FAR → QER → URR

Chaque PDR référence un FAR, qui peut référencer un QER et un ou plusieurs URR.



Exemple de configuration de session

PDR montant :

```
TEID : 5678
ID FAR : 2
ID QER : 1
IDs URR : [0]
Suppression de l'en-tête externe : Faux
```

PDR descendant :

```
IP UE : 10.45.0.1
ID FAR : 1
ID QER : 1
IDs URR : [0]
Mode SDF : Pas de SDF
```

FAR ID 1 (Descendant) :

```
Action : 2 (TRANSFÉRER)
Création d'en-tête externe : Vrai
IP distante : 198.51.100.10
TEID : 5678
```

FAR ID 2 (Montant) :

```
Action : 2 (TRANSFÉRER)
Création d'en-tête externe : Faux
```

ID QER 1 :

```
QFI : 9
MBR montant : 100000 kbps
MBR descendant : 100000 kbps
GBR montant : 0 kbps
GBR descendant : 0 kbps
```

ID URR 0 :

```
Volume montant : 12.3 KB
Volume descendant : 9.0 KB
Volume total : 21.3 KB
```

Opérations courantes

Voir les règles pour une session

Via la page des sessions :

1. Naviguez vers Sessions
2. Trouvez l'UE par IP ou TEID
3. Cliquez sur "Développer" pour voir toutes les règles (PDR, FAR, QER, URR)

Via la page des règles :

1. Naviguez vers Règles
2. Utilisez la recherche par TEID (montant) ou IP UE (descendant) dans l'onglet PDR
3. Notez l'ID FAR, l'ID QER, les IDs URR
4. Passez aux onglets FAR/QER/URR pour voir les règles référencées

Activer/Désactiver la mise en mémoire tampon

Scénario : Pendant le transfert, tamponner les paquets pour éviter la perte

Étapes :

1. Naviguez vers Règles → FARs
2. Entrez l'ID FAR dans le champ de recherche
3. Cliquez sur "Rechercher"
4. Si la mise en mémoire tampon est DÉSACTIVÉE, cliquez sur "Activer la mise en mémoire tampon"
5. Vérifiez que le bit d'action 2 est défini (la valeur d'action augmente de 4)

Alternative via la page des tampons :

1. Naviguez vers Tampons
2. Voir les FAR avec la mise en mémoire tampon activée
3. Cliquez sur "Désactiver le tampon" lorsque le transfert est terminé

Surveiller la conformité QoS

Vérifiez si le trafic est limité par le débit :

1. Naviguez vers Règles → QERs
2. Trouvez l'ID QER associé à la session UE
3. Notez les valeurs MBR montant et MBR descendant
4. Comparez avec le taux de croissance du volume URR

Calculez le débit moyen :

```
Débit (kbps) = (Delta de volume en bytes × 8) / (Delta de temps en
secondes × 1000)
```

Si le débit approche le MBR, le trafic est limité par le débit.

Suivre l'utilisation des données

Surveillez les volumes URR :

1. Naviguez vers Règles → URRs
2. Voir les volumes montant, descendant et total
3. Trier par Volume total pour trouver les utilisateurs les plus élevés
4. Actualiser périodiquement pour observer la croissance du volume

Cas d'utilisation :

- Vérifier l'intégration de la facturation
- Détecter une utilisation anormale des données
- Planifier la capacité en fonction des modèles de trafic

Dépannage

Aucun trafic ne circule

Vérifiez le PDR :

1. Vérifiez que le PDR existe pour le TEID (montant) ou l'IP UE (descendant)
2. Confirmez que l'ID FAR est valide
3. Vérifiez que les filtres SDF ne bloquent pas le trafic

Vérifiez le FAR :

1. Vérifiez que l'action FAR est TRANSFÉRER (pas SUPPRIMER ou TAMPONNER uniquement)
2. Confirmez que la création de l'en-tête externe correspond à la direction
3. Vérifiez que l'IP distante et le TEID sont corrects pour le descendant

Vérifiez le QER :

1. Vérifiez que le statut de la porte est Ouvert (0)
2. Vérifiez que le MBR n'est pas trop restrictif

Paquets étant supprimés

Vérifiez la limitation de débit du QER :

1. Naviguez vers Règles → QERs
2. Vérifiez que le MBR est adéquat pour la charge de trafic
3. Vérifiez que la croissance du volume URR correspond au débit attendu

Vérifiez l'action FAR :

1. Naviguez vers Règles → FARs
2. Vérifiez que l'action est TRANSFÉRER, pas SUPPRIMER
3. Vérifiez que la mise en mémoire tampon n'est pas bloquée en mode uniquement TAMPONNER

Problèmes de mise en mémoire tampon

Paquets bloqués dans le tampon :

1. Naviguez vers la page des tampons
2. Vérifiez l'horodatage du paquet le plus ancien
3. Si > 30 secondes, le transfert peut avoir échoué

4. Videz ou effacez manuellement le tampon
5. Désactivez la mise en mémoire tampon sur le FAR

Débordement de tampon :

1. Vérifiez le nombre total de paquets par rapport à Max Total (par défaut 100,000)
2. Vérifiez les paquets par FAR par rapport à Max Par FAR (par défaut 10,000)
3. Effacez les tampons si pleins
4. Enquêtez sur la raison pour laquelle la mise en mémoire tampon n'a pas été désactivée

URR ne suit pas

Les compteurs de volume à zéro :

1. Vérifiez que le PDR référence l'ID URR
2. Vérifiez que les paquets correspondent au PDR
3. Vérifiez que le FAR transfère (pas de suppression) les paquets
4. Confirmez que l'ID URR existe dans la carte URR

Le volume ne se rapporte pas au SMF :

1. Vérifiez la configuration du rapport de session PFCP
2. Vérifiez les déclencheurs de reporting URR (seuils de volume/temps)
3. Consultez les journaux pour les messages de rapport de session PFCP

Documentation connexe

- **Guide des opérations UPF** - Vue d'ensemble de l'architecture et des composants d'OmniUPF
- **Guide des opérations de l'interface Web** - Utilisation du panneau de contrôle pour la visualisation des règles
- **Guide de surveillance** - Statistiques et surveillance de la capacité
- **Guide de dépannage** - Problèmes courants et diagnostics

Guide de dépannage OmniUPF

Table des matières

1. Aperçu
2. Outils de diagnostic
3. Problèmes d'installation
4. Problèmes de configuration
5. Problèmes d'association PFCP
6. Problèmes de traitement des paquets
7. Problèmes XDP et eBPF
8. Problèmes de performance
9. Problèmes spécifiques à l'hyperviseur
10. Problèmes de NIC et de pilote
11. Échecs d'établissement de session
12. Problèmes de mise en mémoire tampon

Aperçu

Ce guide fournit des procédures de dépannage systématiques pour les problèmes courants d'OmniUPF. Chaque section comprend des symptômes, des étapes de diagnostic, des causes profondes et des procédures de résolution.

Liste de contrôle de diagnostic rapide

Avant de procéder à un dépannage approfondi, vérifiez :

```
# 1. Vérifiez qu'OmniUPF fonctionne
systemctl status omniupf

# 2. Vérifiez l'association PFCP
curl http://localhost:8080/api/v1/upf_pipeline

# 3. Vérifiez que les cartes eBPF sont chargées
ls /sys/fs/bpf/

# 4. Vérifiez que le programme XDP est attaché
ip link show | grep -i xdp

# 5. Vérifiez les journaux du noyau pour des erreurs
dmesg | tail -50
journalctl -u omniupf -n 50
```

Outils de diagnostic

API REST OmniUPF

Vérifiez l'état du UPF :

```
curl http://localhost:8080/api/v1/upf_status
```

Vérifiez les associations PFCP :

```
curl http://localhost:8080/api/v1/upf_pipeline
```

Vérifiez le nombre de sessions :

```
curl http://localhost:8080/api/v1/sessions | jq 'length'
```

Vérifiez la capacité de la carte eBPF :

```
curl http://localhost:8080/api/v1/map_info
```

Vérifiez les statistiques des paquets :

```
curl http://localhost:8080/api/v1/packet_stats
```

Vérifiez les statistiques XDP :

```
curl http://localhost:8080/api/v1/xdp_stats
```

Inspection de la carte eBPF

Listez toutes les cartes eBPF :

```
ls -lh /sys/fs/bpf/
bpftool map list
```

Afficher les détails de la carte :

```
bpftool map show
bpftool map dump name pdr_map_downlin
```

Compter les entrées dans la carte :

```
bpftool map dump name far_map | grep -c "key:"
```

Inspection du programme XDP

Vérifiez si le programme XDP est attaché :

```
ip link show eth0 | grep xdp
```

Listez tous les programmes XDP :

```
bpftool net list
```

Afficher les détails du programme XDP :

```
bpftool prog show
```

Dump des statistiques XDP :

```
bpftool prog dump xlated name xdp_upf_func
```

Débogage réseau

Capturez le trafic PFCP sur N4 (plan de contrôle) :

```
# PFCP n'est pas traité par XDP, tcpdump fonctionne normalement
tcpdump -i eth0 -n udp port 8805 -w /tmp/pfcp_traffic.pcap
```

Capturez le trafic GTP-U sur N3 (nécessite une capture hors bande) :

```
# AVERTISSEMENT : Le tcpdump standard sur l'hôte UPF NE PEUT PAS
capturer les paquets traités par XDP !
# XDP traite GTP-U avant que la pile réseau du noyau ne voie les
paquets.

# Utilisez plutôt une capture hors bande :
# 1. TAP réseau entre gNB et UPF
# 2. Miroir de port de commutateur/SPAN pour copier le trafic N3
# 3. Miroir de port de commutateur virtuel vers la VM d'analyse

# Sur l'hôte d'analyse/de surveillance (PAS sur UPF) :
# tcpdump -i <mirror_interface> -n udp port 2152 -w
/tmp/n3_capture.pcap

# Ou utilisez l'API de statistiques pour les comptes de paquets :
curl http://localhost:8080/api/v1/packet_stats
curl http://localhost:8080/api/v1/n3n6_stats
```

Surveillez les compteurs de paquets :

```
watch -n 1 'ip -s link show eth0'
```

Vérifiez la table de routage :

```
ip route show
ip route get 10.45.0.100  # Vérifiez la route pour l'IP UE
```

Vérifiez la table ARP :

```
ip neigh show
```

Problèmes d'installation

Problème : "système de fichiers eBPF non monté"

Symptômes :

```
ERRO[0000] échec du chargement des objets eBPF : monter le système
de fichiers bpf à /sys/fs/bpf
```

Cause : système de fichiers eBPF non monté

Résolution :

```
# Monter le système de fichiers eBPF
sudo mount bpffs /sys/fs/bpf -t bpf

# Rendre persistant (ajouter à /etc/fstab)
echo "bpffs /sys/fs/bpf bpf defaults 0 0" | sudo tee -a /etc/fstab

# Vérifiez le montage
mount | grep bpf
```

Problème : Version du noyau trop ancienne

Symptômes :

```
ERRO[0000] la version du noyau 5.4.0 est trop ancienne, la version
minimale requise est 5.15.0
```

Cause : version du noyau Linux inférieure à l'exigence minimale

Résolution :

```
# Vérifiez la version du noyau
uname -r

# Mettez à niveau le noyau (Ubuntu/Debian)
sudo apt update
sudo apt install linux-generic-hwe-22.04
sudo reboot

# Vérifiez le nouveau noyau
uname -r  # Devrait être >= 5.15.0
```

Problème : Dépendance libbpf manquante

Symptômes :

```
erreur lors du chargement des bibliothèques partagées :
libbpf.so.0 : impossible d'ouvrir le fichier d'objet partagé
```

Cause : bibliothèque libbpf non installée

Résolution :

```
# Installer libbpf (Ubuntu/Debian)
sudo apt update
sudo apt install libbpf-dev

# Vérifiez l'installation
ldconfig -p | grep libbpf
```

Problèmes de configuration

Problème : Fichier de configuration invalide

Symptômes :

```
ERRO[0000] impossible de lire le fichier de configuration :
erreurs de désérialisation
```

Cause : erreur de syntaxe dans le fichier de configuration Elixir

Résolution :

```
# Vérifiez la syntaxe du fichier de configuration
cd /opt/omniupf && bin/upf_ex eval "IO.puts(:ok)"

# Problèmes courants :
# - Guillemets manquants autour des valeurs de chaîne
# - Syntaxe Elixir incorrecte (par exemple, utiliser : au lieu de
=)
# - Chaînes ou crochets non fermés

# Le fichier de configuration se trouve à /etc/omniupf/runtime.exs
# Exemple de syntaxe correcte :
# xdp_interfaces = "eth0"
# xdp_attach_mode = "generic"
# api_port = 8080
# pfcp_address = "10.0.0.1"
```

Problème : Nom d'interface non trouvé

Symptômes :

```
ERRO[0000] interface eth0 non trouvée
```

Cause : l'interface configurée n'existe pas

Résolution :

```
# Listez toutes les interfaces réseau
ip link show

# Vérifiez l'état de l'interface
ip addr show eth0

# Si l'interface a un nom différent, mettez à jour
/etc/omniupf/runtime.exs :
# xdp_interfaces = "ens1f0"  # Utilisez le nom réel de l'interface

# Pour les VM, vérifiez le schéma de nommage des interfaces
ls /sys/class/net/
```

Problème : Port déjà utilisé

Symptômes :

```
ERRO[0000] échec du démarrage du serveur API : adresse déjà
utilisée
```

Cause : le port 8080, 8805 ou 9090 est déjà lié par un autre processus

Résolution :

```
# Trouvez le processus utilisant le port
sudo lsof -i :8080
sudo netstat -tulpn | grep :8080

# Tuez le processus en conflit
sudo kill <PID>

# Ou changez le port OmniUPF dans /etc/omniupf/runtime.exs
# api_port = 8081
# pfcp_port = 8806
```

Problème : ID de nœud PFCP invalide

Symptômes :

```
ERRO[0000] pfcp_node_id invalide : doit être une adresse IPv4
valide
```

Cause : l'ID de nœud PFCP n'est pas une adresse IPv4 valide

Résolution :

```
# Dans /etc/omniupf/runtime.exs
# Correct : Utilisez une adresse IP (pas un nom d'hôte)
node_id = "10.100.50.241"

# Incorrect :
# node_id = "localhost"
# node_id = "upf.example.com"
```

Problèmes d'association PFCP

Problème : Aucune association PFCP établie

Symptômes :

- L'interface Web affiche "Aucune association"
- Les journaux SMF affichent "Échec de l'établissement de l'association PFCP"

Diagnostic :

```
# 1. Vérifiez si le serveur PFCP écoute
sudo netstat -ulpn | grep 8805

# 2. Vérifiez les règles de pare-feu
sudo iptables -L -n | grep 8805
sudo ufw status

# 3. Capturez le trafic PFCP
tcpdump -i any -n udp port 8805 -vv

# 4. Vérifiez les associations PFCP via l'API
curl http://localhost:8080/api/v1/upf_pipeline
```

Causes et résolutions courantes :

Pare-feu bloquant PFCP

Résolution :

```
# Autoriser le trafic PFCP (UDP 8805)
sudo ufw allow 8805/udp
sudo iptables -A INPUT -p udp --dport 8805 -j ACCEPT
```

Mauvais ID de nœud PFCP

Résolution :

```
# Dans /etc/omniupf/runtime.exs, définissez l'ID de nœud sur l'IP
de l'interface N4 correcte
node_id = "10.100.50.241"    # Doit correspondre à l'IP sur le
réseau N4
```

Réseau inaccessible au SMF

Résolution :

```
# Testez la connectivité au SMF
ping <SMF_IP>

# Vérifiez le routage vers le SMF
ip route get <SMF_IP>

# Ajoutez une route si elle est manquante
sudo ip route add <SMF_NETWORK>/24 via <GATEWAY>
```

SMF configuré avec une mauvaise IP UPF

Résolution :

- Vérifiez la configuration du SMF pour l'adresse UPF
- Assurez-vous que le SMF a configuré l'IP `pfcp_node_id` de l'UPF
- Vérifiez que le SMF peut router vers le réseau N4 de l'UPF

Problème : Échecs de battement PFCP

Symptômes :

```
WARN[0030] délai d'attente du battement PFCP pour l'association
10.100.50.10
```

Diagnostic :

```
# Vérifiez les statistiques PFCP
curl http://localhost:8080/api/v1/upf_pipeline | jq
'.associations[] | {remote_id, uplink_teid_count}'

# Surveillez les journaux de battement
journalctl -u omniupf -f | grep heartbeat
```

Causes et résolutions :

Perte de paquets réseau

Résolution :

```
# Vérifiez la perte de paquets vers le SMF
ping -c 100 <SMF_IP> | grep loss

# Si perte élevée, enquêtez sur le réseau :
# - Vérifiez l'état du lien
# - Vérifiez la santé du commutateur/routeur
# - Vérifiez la congestion
```

Intervalle de battement trop agressif

Résolution :

```
# Dans /etc/omniupf/runtime.exs
heartbeat_interval_ms = 30_000    # Augmentez de 5000 à 30000 ms
heartbeat_retries = 5             # Augmentez les tentatives
heartbeat_timeout_ms = 10_000     # Augmentez le délai d'attente
```

Problèmes de traitement des paquets

Problème : Aucun paquet ne circule (compteurs RX/TX à 0)

Symptômes :

- La page des statistiques affiche 0 paquets RX/TX
- UE ne peut pas établir de session de données

Diagnostic :

```
# 1. Vérifiez si le programme XDP est attaché
ip link show eth0 | grep xdp

# 2. Vérifiez que l'interface est UP
ip link show eth0

# 3. Vérifiez les statistiques des paquets (connaissant XDP)
# Remarque : tcpdump ne peut pas voir les paquets GTP-U traités
par XDP
curl http://localhost:8080/api/v1/packet_stats
```

Résolutions :

Programme XDP non attaché

Résolution :

```
# Redémarrez OmniUPF pour réattacher XDP
sudo systemctl restart omniupf

# Vérifiez l'attachement
ip link show eth0 | grep xdp
bpftool net list
```

Interface hors service ou pas de lien

Résolution :

```
# Mettez l'interface en marche
sudo ip link set eth0 up

# Vérifiez l'état du lien
ethtool eth0 | grep "Link detected"

# Si le lien est hors service, vérifiez la connexion physique ou
la configuration réseau de la VM
```

Mauvaise interface configurée

Résolution :

```
# Dans /etc/omniupf/runtime.exs
xdp_interfaces = "ens1f0"    # Utilisez le nom réel de l'interface
à partir de 'ip link show'
```

Problème : Paquets reçus mais non transférés (taux de perte élevé)

Symptômes :

- Compteurs RX augmentant mais compteurs TX non
- Taux de perte > 1%

Diagnostic :

```
# Vérifiez les statistiques de perte
curl http://localhost:8080/api/v1/xdp_stats | jq '.drop'

# Vérifiez les statistiques de routage
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'

# Surveillez les pertes de paquets
watch -n 1 'curl -s http://localhost:8080/api/v1/packet_stats | jq
".total_rx, .total_tx, .total_drop"'
```

Causes courantes :

Pas de correspondance PDR (TEID ou IP UE inconnus)

Résolution :

```
# Vérifiez si des sessions existent
curl http://localhost:8080/api/v1/sessions

# Si aucune session, vérifiez :
# - L'association PFCP est établie
# - Le SMF a créé des sessions
# - L'établissement de session a réussi

# Vérifiez les entrées de la carte PDR
bpftool map dump name pdr_map_teid_ip | grep -c key
bpftool map dump name pdr_map_downlin | grep -c key
```

Échecs de routage

Résolution :

```
# Vérifiez les échecs de recherche FIB
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'

# Testez le routage pour l'IP UE
ip route get 10.45.0.100

# Ajoutez une route manquante
sudo ip route add 10.45.0.0/16 dev eth1  # Routez le pool UE vers
N6
```

Limitation de débit QER

Symptômes :

- Débit inférieur à celui attendu
- Trafic limité à un taux spécifique
- Compteurs de volume URR affichent un comportement de plateau
- Compteurs de perte XDP augmentant lors des pics de trafic

Diagnostic :

1. **Vérifiez le MBR configuré pour la session** :

```
# Trouvez l'ID QER de la session
curl http://localhost:8080/api/v1/pfcp_sessions | jq '.data[] |
select(.ue_ip == "10.45.0.1")'

# Recherchez la configuration QER
curl http://localhost:8080/api/v1/qer_map | jq '.data[] |
select(.qer_id == 1)'
```

2. **Vérifiez l'état de la porte** :

```
# L'état de la porte doit être 0 (OUVERTE) pour l'ascendant et
le descendant
curl http://localhost:8080/api/v1/qer_map | jq '.data[] |
{qer_id, ul_gate: .ul_gate_status, dl_gate: .dl_gate_status}'
```

3. **Calculez le débit réel à partir de l'URR** :

```
# Interrogez les compteurs de volume URR à deux moments
curl http://localhost:8080/api/v1/urr_map | jq '.data[] |
select(.urr_id == 0)'

# Calculez le débit (manuel) :
# throughput_kbps = (volume_delta_bytes × 8) /
time_delta_seconds / 1000
```

4. **Comparez MBR vs. débit réel** :
    - Débit attendu ≈ 95-98 % du MBR (en raison de la surcharge de protocole)
    - Si le débit est significativement inférieur au MBR, vérifiez d'autres goulets d'étranglement
    - Si le débit correspond exactement au MBR, la limitation de débit fonctionne comme prévu

Résolution :

- **Si le MBR est trop bas** : Demandez au SMF de mettre à jour le QER avec un MBR plus élevé via la modification de session PFCP

- **Si la porte est fermée** : Enquêtez sur les raisons pour lesquelles le SMF a fermé la porte (politique, quota ou erreur)
- **Si la limitation de débit est inattendue** : Vérifiez la configuration de la politique SMF et le profil QoS

Comprendre l'application du MBR :

OmniUPF utilise un algorithme de fenêtre glissante pour appliquer les limites MBR avec une précision nanoseconde dans le chemin de données eBPF. Voir Guide de gestion des règles - Mécanisme d'application MBR pour une explication détaillée de :

- Comment la taille des paquets et le taux déterminent les décisions de perte
- Pourquoi le débit observé diffère du MBR configuré
- Limitation de débit par direction (montante/descendante)
- Comportement de fenêtre glissante de 5 ms

Scénarios courants :

- **Appels VoIP interrompus** : Vérifiez si le MBR est suffisant pour le débit du codec (G.711 = ~80 kbps)
- **Mise en mémoire tampon de streaming vidéo** : Assurez-vous que le MBR > débit vidéo + surcharge (1080p = ~5-10 Mbps)
- **Trafic en rafale** : Petites rafales autorisées dans la fenêtre de 5 ms, le trafic soutenu est limité par le taux

---

Problème : Trafic unidirectionnel (l'ascendant fonctionne, le descendant ne fonctionne pas)

Symptômes :

- Paquets RX N3 mais pas de paquets TX N3 (problème descendant)
- Paquets RX N6 mais pas de paquets TX N6 (problème ascendant)

Diagnostic :

```
# Vérifiez les statistiques des interfaces N3/N6 (méthode
consciente de XDP)
curl http://localhost:8080/api/v1/n3n6_stats
curl http://localhost:8080/api/v1/packet_stats

# Remarque : Le tcpdump standard ne peut pas capturer le trafic
GTP-U traité par XDP
# Utilisez l'API de statistiques ou xdpdump pour l'analyse du
trafic
# Voir la section "Capture de paquets avec XDP" pour plus de
détails
```

Échec ascendant (RX N3, pas de TX N6) :

Cause : Pas d'action FAR ou problème de routage vers N6

Résolution :

```
# Vérifiez que le FAR a une action FORWARD
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
select(.applied_action == 2)'

# Vérifiez que la route N6 existe
ip route get 8.8.8.8  # Testez la route vers Internet

# Ajoutez une route par défaut si elle est manquante
sudo ip route add default via <N6_GATEWAY> dev eth1
```

Échec descendant (RX N6, pas de TX N3) :

Cause : Pas de PDR descendant ou encapsulation GTP manquante

Résolution :

```
# Vérifiez qu'un PDR descendant existe pour l'IP UE
curl http://localhost:8080/api/v1/sessions | jq '.[].pdrs[] |
select(.pdi.ue_ip_address)'

# Vérifiez que le FAR a OUTER_HEADER_CREATION
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
.outer_header_creation'

# Vérifiez la connectivité gNB
ping <GNB_N3_IP>
```

Problèmes XDP et eBPF

Pour des informations détaillées sur la configuration XDP, la sélection de mode et le dépannage, consultez le Guide des modes XDP.

Problème : Échec du chargement du programme XDP

Symptômes :

```
ERRO[0000] échec du chargement du programme XDP : argument
invalide
```

Diagnostic :

```
# Vérifiez le support XDP du noyau
grep XDP /boot/config-$(uname -r)

# Devrait afficher :
# CONFIG_XDP_SOCKETS=y
# CONFIG_BPF=y
# CONFIG_BPF_SYSCALL=y

# Vérifiez dmesg pour une erreur détaillée
dmesg | grep -i bpf
```

Causes et résolutions :

Le noyau manque de support XDP

Résolution :

```
# Reconstruisez le noyau avec le support XDP ou mettez à niveau
vers un noyau plus récent
# Ubuntu 22.04+ a XDP activé par défaut
sudo apt install linux-generic-hwe-22.04
sudo reboot
```

Échec de vérification du programme XDP

Résolution :

```
# Vérifiez les journaux OmniUPF pour les erreurs de vérificateur
journalctl -u omniupf | grep verifier

# Problèmes courants :
# - La complexité eBPF dépasse les limites (augmentez les limites
du noyau)
# - Accès mémoire invalide (bug dans le code eBPF)

# Augmentez le niveau de journalisation du vérificateur eBPF pour
le débogage
sudo sysctl kernel.bpf_stats_enabled=1
```

Problème : Compte des avortements XDP en augmentation

Symptômes :

- Les statistiques XDP montrent avortés > 0
- Augmentation des pertes de paquets

Diagnostic :

```
# Vérifiez le compte des avortements XDP
curl http://localhost:8080/api/v1/xdp_stats | jq '.aborted'

# Surveillez les statistiques XDP
watch -n 1 'curl -s http://localhost:8080/api/v1/xdp_stats'
```

Cause : le programme eBPF a rencontré une erreur d'exécution

Résolution :

```
# Vérifiez les journaux du noyau pour les erreurs eBPF
dmesg | grep -i bpf

# Redémarrez OmniUPF pour recharger le programme eBPF
sudo systemctl restart omniupf

# Si le problème persiste, activez la journalisation eBPF
(nécessite une reconstruction) :
# Construisez OmniUPF avec BPF_ENABLE_LOG=1
```

Problème : Carte eBPF pleine (capacité épuisée)

Symptômes :

- Établissement de session échoue
- Capacité de la carte à 100 %

Diagnostic :

```
# Vérifiez la capacité de la carte
curl http://localhost:8080/api/v1/map_info | jq '.[] | {map_name,
capacity, used, usage_percent}'

# Identifiez les cartes pleines
curl http://localhost:8080/api/v1/map_info | jq '.[] |
select(.usage_percent > 90)'
```

Atténuation immédiate :

```
# 1. Identifiez les sessions obsolètes
curl http://localhost:8080/api/v1/sessions | jq '.[] | {seid,
uplink_teid, created_at}'

# 2. Demandez au SMF de supprimer les anciennes sessions
# (via l'interface d'administration SMF ou l'API)

# 3. Surveillez la diminution de l'utilisation de la carte
watch -n 5 'curl -s http://localhost:8080/api/v1/map_info | jq ".
[] | select(.map_name==\"pdr_map_downlin\") | .usage_percent"'
```

Résolution à long terme :

```
# Dans /etc/omniupf/runtime.exs
max_sessions = 200_000    # Augmentez de 100_000

# Ou définissez des tailles de carte individuelles
far_map_size = 400_000
qer_map_size = 200_000
```

Important : Changer les tailles de carte nécessite un redémarrage d'OmniUPF et **efface toutes les sessions existantes**.

Problèmes de performance

Problème : Faible débit (inférieur aux attentes)

Symptômes :

- Débit < 1 Gbps malgré un NIC capable
- Utilisation élevée du CPU

Diagnostic :

```
# Vérifiez le taux de paquets
curl http://localhost:8080/api/v1/packet_stats | jq '.total_rx,
.total_tx'

# Vérifiez les statistiques du NIC
ethtool -S eth0 | grep -i drop

# Vérifiez le mode XDP
ip link show eth0 | grep xdp
```

Résolutions :

Utilisation du mode XDP générique

Résolution :

```
# Dans /etc/omniupf/runtime.exs
xdp_attach_mode = "native"    # Nécessite un NIC/driver compatible
XDP
```

Goulet d'étranglement à un seul cœur

Résolution :

```
# Activez RSS (Receive Side Scaling) sur le NIC
ethtool -L eth0 combined 4  # Utilisez 4 files RX/TX

# Vérifiez que RSS est activé
ethtool -l eth0

# Fixez les interruptions à des CPU spécifiques
# Voir /proc/interrupts et utilisez irqbalance ou l'affinité
manuelle
```

Bloat de tampon

Résolution :

```
Les limites de tampon sont gérées par le plan de contrôle Elixir.
Vérifiez la configuration du tampon dans /etc/omniupf/runtime.exs.
```

Problème : Latence élevée

Symptômes :

- Latence ping > 50 ms
- Dégradation de l'expérience utilisateur

Diagnostic :

```
# Testez la latence vers l'UE
ping -c 100 <UE_IP> | grep avg

# Vérifiez les paquets mis en mémoire tampon
curl http://localhost:8080/api/v1/upf_buffer_info | jq
'.total_packets_buffered'

# Vérifiez les performances du cache de routage
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'
```

Résolutions :

Paquets mis en mémoire tampon de manière excessive

Résolution :

```
# Vérifiez pourquoi les paquets sont mis en mémoire tampon
curl http://localhost:8080/api/v1/upf_buffer_info | jq '.buffers[]
| {far_id, packet_count, direction}'

# Effacez les tampons s'ils sont bloqués
# (redémarrez OmniUPF ou déclenchez une modification de session
PFCP pour appliquer le FAR)
```

Latence de recherche FIB

Résolution :

```
# Assurez-vous que le cache de routage est activé (option de
construction)
# Construisez avec BPF_ENABLE_ROUTE_CACHE=1

# Optimisez la table de routage
# Utilisez moins de routes, plus spécifiques au lieu de nombreuses
petites routes
```

Problème : Pertes de paquets sous charge

Symptômes :

- Le taux de perte augmente avec le trafic
- Erreurs RX sur le NIC

Diagnostic :

```
# Vérifiez les erreurs du NIC
ethtool -S eth0 | grep -E "drop|error|miss"

# Vérifiez la taille du tampon de la file
ethtool -g eth0

# Surveillez les pertes en temps réel
watch -n 1 'ethtool -S eth0 | grep -E "drop|miss"'
```

Résolution :

```
# Augmentez la taille du tampon RX
ethtool -G eth0 rx 4096

# Augmentez la taille du tampon TX
ethtool -G eth0 tx 4096

# Vérifiez les nouveaux paramètres
ethtool -g eth0
```

Problèmes Spécifiques au Hyperviseur

Pour des instructions de configuration étape par étape du hyperviseur, voir le Guide des Modes XDP.

Proxmox : XDP ne fonctionne pas dans la VM

Symptômes :

- Impossible d'attacher le programme XDP en mode natif
- Seul le mode générique fonctionne

Cause : VM utilisant un réseau en pont sans SR-IOV

Résolution :

Option 1 : Utiliser le mode générique (le plus simple)

```
# Dans /etc/omniupf/runtime.exs
xdp_attach_mode = "generic"
```

Option 2 : Configurer le passthrough SR-IOV

```
# Sur l'hôte Proxmox :
# 1. Activer IOMMU
nano /etc/default/grub
# Ajouter : intel_iommu=on iommu=pt
update-grub
reboot

# 2. Créer des VFs
echo 4 > /sys/class/net/eth0/device/sriov_numvfs

# 3. Assigner le VF à la VM dans l'interface Proxmox
# Matériel → Ajouter → Périphérique PCI → Sélectionner VF
```

Dans `/etc/omniupf/runtime.exs` :

```
xdp_interfaces = "ens1f0"    # SR-IOV VF
xdp_attach_mode = "native"
```

VMware : Mode promiscuous requis

Symptômes :

- Paquets non reçus par OmniUPF

Cause : vSwitch bloquant les adresses MAC non correspondantes

Résolution :

```
# Activer le mode promiscuous sur vSwitch (dans vSphere Client) :
# 1. Sélectionner vSwitch → Modifier les paramètres
# 2. Sécurité → Mode Promiscuous : Accepter
# 3. Sécurité → Changements d'adresse MAC : Accepter
# 4. Sécurité → Transmissions forgées : Accepter
```

VirtualBox : Performance très faible

Symptômes :

- Débit < 100 Mbps

Cause : VirtualBox ne prend pas en charge SR-IOV ou XDP natif

Résolution :

```
# Dans /etc/omniupf/runtime.exs
xdp_attach_mode = "generic"    # Seule option pour VirtualBox
```

Optimiser les paramètres de VirtualBox :

- Utiliser l'adaptateur VirtIO-Net (si disponible)
- Activer le mode promiscuous "Autoriser tout"
- Allouer plus de cœurs CPU à la VM
- Utiliser un réseau en pont au lieu de NAT
- Envisager de migrer vers KVM/Proxmox pour de meilleures performances

---

Problèmes de NIC et de Pilote

Problème : Le pilote NIC ne prend pas en charge XDP

Symptômes :

```
ERRO[0000] échec de l'attachement du programme XDP : opération non
prise en charge
```

Diagnostic :

```
# Vérifier le pilote NIC
ethtool -i eth0 | grep driver

# Vérifier si le pilote prend en charge XDP
modinfo <driver_name> | grep -i xdp

# Lister les interfaces capables de XDP
ip link show | grep -B 1 "xdpgeneric\|xdpdrv\|xdpoffload"
```

Résolution :

Option 1 : Utiliser le mode générique

```
# Dans /etc/omniupf/runtime.exs
xdp_attach_mode = "generic"
```

Option 2 : Mettre à jour le pilote NIC

```
# Vérifier les mises à jour du pilote (Ubuntu)
sudo apt update
sudo apt install linux-modules-extra-$(uname -r)

# Ou installer un pilote spécifique au fournisseur
# Exemple pour Intel :
# Télécharger depuis https://downloadcenter.intel.com/
```

Option 3 : Remplacer le NIC

```
# Utiliser un NIC capable de XDP :
# - Intel X710, E810
# - Mellanox ConnectX-5, ConnectX-6
# - Broadcom BCM57xxx (pilote bnxt_en)
```

Problème : Le pilote plante ou provoque des panics du noyau

Symptômes :

- Panic du noyau après l'attachement de XDP
- NIC ne répond plus

Diagnostic :

```
# Vérifier les journaux du noyau
dmesg | tail -100

# Vérifier les bugs du pilote
journalctl -k | grep -E "BUG:|panic:"
```

Résolution :

```
# 1. Mettre à jour le noyau et les pilotes
sudo apt update
sudo apt upgrade
sudo reboot

# 2. Dans /etc/omniupf/runtime.exs, désactiver le déchargement XDP
(utiliser uniquement natif)
# xdp_attach_mode = "native"

# 3. Utiliser le mode générique comme solution de contournement
# xdp_attach_mode = "generic"

# 4. Signaler le bug au fournisseur de NIC ou à l'équipe du noyau
Linux
```

Problème : SLAAC IPv6 intermittent / erreurs de somme de contrôle UDP (generic-XDP + virtio)

Symptômes :

```
# Les UE IPv6 ne forment jamais de manière intermittente (~50%)
une adresse globale via SLAAC.
# Le RA est émis par le UPF mais n'est jamais vu par le RAN.
# Sur l'eNB/gNB (ou tout hôte en aval), le compteur d'erreurs de
somme de contrôle UDP du noyau
# augmente, un par RA perdu :
nstat -az | grep UdpInCsumErrors      # s'incrémente lors des
tentatives de SLAAC échouées
```

- Affecte **uniquement** SLAAC IPv6 / Annonces de routeurs — les données du plan utilisateur sont correctes.
- Se produit **uniquement** lorsque les SGW-U et PGW-U d'un porteur se trouvent sur des membres de pool **différents** (séparés) ; un porteur SGW-U+PGW-U sur la même boîte réussit toujours.

Cause : L'annonce de routeur est générée en **espace utilisateur** (plan de contrôle), donc son tampon de socket est marqué pour le déchargement de somme de contrôle TX (`CHECKSUM_PARTIAL` ; le NIC du chemin de données a `tx-checksum-ip-generic: on`). Dans un chemin de données **XDP en mode générique**, le paquet transféré est un `skb` du noyau : lorsqu'un membre SGW-U de pool séparé relaie ce RA, il réécrit l'IP externe + TEID et met `udp->check` à zéro, mais le skb conserve le `csum_start`/`csum_offset` obsolète. À la sortie, le NIC (virtio) recompute la somme de contrôle UDP sur le paquet modifié à partir de cet offset obsolète, produisant une somme de contrôle incorrecte, et le récepteur la rejette. Le trafic du plan utilisateur XDP natif n'est pas affecté car il est construit en eBPF avec `CHECKSUM_NONE` (une somme de contrôle zéro réelle), donc le NIC ne la recompute jamais — c'est pourquoi seul SLAAC échoue. Les modes XDP natif/déchargement ne rencontrent pas ce problème (pas de skb, pas de recompute de déchargement) ; il est spécifique au **mode**

générique sur un NIC à déchargement de somme de contrôle (par exemple, virtio).

Résolution :

```
# Désactiver le déchargement de somme de contrôle TX sur le(s) NIC
du chemin de données afin que le noyau calcule
# la somme de contrôle UDP en ligne (correctement) pour les
paquets de contrôle d'origine espace utilisateur. Le XDP
# contourne le chemin de somme de contrôle du noyau, donc le débit
de transfert n'est pas affecté.
ethtool -K <n3_iface> tx off
ethtool -K <n6_iface> tx off

# Vérifier (devrait indiquer "off") :
ethtool -k <n3_iface> | awk '/^tx-checksumming/{print}'

# Re-tester : SLAAC devrait maintenant réussir pour chaque porteur
et UdpInCsumErrors doit rester stable.
```

> Rendez-le persistant dans votre déploiement (par exemple, dans l'unité de démarrage du NIC du chemin de données) afin qu'il survive aux redémarrages. Lorsque le NIC le prend en charge, le mode XDP natif (`xdp_attach_mode: native`) évite également complètement le problème.

Échecs d'Établissement de Session

Problème : Échec de l'établissement de session

Symptômes :

- SMF signale un échec de l'établissement de session
- UE ne peut pas établir de session PDU

Voir la Référence des Codes de Cause PFCP pour les scénarios d'échec courants et les résolutions.

Diagnostic :

```
# Vérifier les journaux OmniUPF pour les erreurs de session
journalctl -u omniupf | grep -i "session establishment"

# Vérifier le nombre de sessions PFCP
curl http://localhost:8080/api/v1/sessions | jq 'length'

# Capturer le trafic PFCP lors de l'établissement de session
tcpdump -i any -n udp port 8805 -w /tmp/pfcp_session.pcap
```

Causes Courantes :

Capacité de la carte pleine

Résolution :

```
# Vérifier l'utilisation de la carte
curl http://localhost:8080/api/v1/map_info | jq '.[] |
select(.usage_percent > 90)'

# Augmenter la capacité (voir la section sur la carte eBPF pleine
ci-dessus)
```

Paramètres PDR/FAR invalides

Résolution :

```
# Vérifier les journaux OmniUPF pour les erreurs de validation
journalctl -u omniupf | grep -E "invalid|error" | tail -20

# Problèmes courants :
# - Adresse IP UE invalide (0.0.0.0 ou dupliquée)
# - TEID invalide (0 ou dupliqué)
# - FAR manquant pour PDR
# - Action FAR invalide

# Vérifier la configuration SMF et les paramètres de session
```

Fonctionnalité non prise en charge (UEIP/FTUP)

Résolution :

```
# Dans /etc/omniupf/runtime.exs
feature_ueip = true
ueip_pool = "10.60.0.0/16"

feature_ftup = true
teid_pool_start = 1
teid_pool_end = 100_000
```

Problèmes de Mise en Tampon

Problème : Paquets bloqués dans le tampon

Symptômes :

- Compte de paquets mis en tampon en augmentation
- Paquets non livrés après le transfert

Diagnostic :

```
# Vérifier les statistiques de tampon
curl http://localhost:8080/api/v1/upf_buffer_info

# Vérifier les tampons FAR individuels
curl http://localhost:8080/api/v1/upf_buffer_info | jq '.buffers[]
| {far_id, packet_count, oldest_packet_ms}'

# Surveiller la taille du tampon
watch -n 5 'curl -s http://localhost:8080/api/v1/upf_buffer_info |
jq ".total_packets_buffered"'
```

Causes & Résolutions :

FAR jamais mis à jour vers FORWARD

Cause : SMF n'a jamais envoyé de modification de session PFCP pour appliquer le FAR

Résolution :

```
# Vérifier l'état du FAR
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
{far_id, applied_action}'

# Action BUFF = 1 (mise en tampon)
# Action FORW = 2 (transmission)

# Si bloqué dans l'état BUFF, demander au SMF de :
# - Envoyer une demande de modification de session PFCP
# - Mettre à jour le FAR avec l'action FORW
```

TTL du tampon expiré

Cause : Les paquets ont expiré avant la mise à jour du FAR

Résolution : Le TTL du tampon est géré par le plan de contrôle Elixir. Vérifiez la configuration du tampon dans `/etc/omniupf/runtime.exs`.

Débordement du tampon

Cause : Trop de paquets mis en tampon par FAR

Résolution : Les limites de tampon sont gérées par le plan de contrôle Elixir. Vérifiez la configuration du tampon dans `/etc/omniupf/runtime.exs`.

Débogage Avancé

Activer la Journalisation de Débogage

```
# Dans /etc/omniupf/runtime.exs
log_level = :debug    # :debug | :info | :warning | :error
```

```
# Redémarrer OmniUPF avec la journalisation de débogage
sudo systemctl restart omniupf

# Surveiller les journaux en temps réel
journalctl -u omniupf -f --output cat
```

Traçage de Programme eBPF

```
# Traçage de l'exécution du programme eBPF (nécessite bpftrace)
sudo bpftrace -e 'tracepoint:xdp:* { @[probe] = count(); }'

# Traçage des opérations de carte
sudo bpftrace -e 'tracepoint:bpf:bpf_map_lookup_elem {
printf("%s\n", str(args->map_name)); }'
```

Capture de Paquet avec XDP

Comprendre les Limitations de Capture de Paquet XDP :

XDP traite les paquets **avant** la pile réseau du noyau, donc `tcpdump` standard **ne peut pas voir le trafic traité par XDP**. Les paquets GTP-U (port UDP 2152) sur N3 sont traités par XDP et n'apparaîtront pas dans tcpdump sur l'hôte UPF.

Méthodes Recommandées pour l'Analyse du Trafic :

```
# Méthode 1 : Utiliser l'API de statistiques pour la surveillance
(RECOMMANDÉE)
curl http://localhost:8080/api/v1/xdp_stats
curl http://localhost:8080/api/v1/packet_stats | jq
curl http://localhost:8080/api/v1/n3n6_stats

# Méthode 2 : Capturer le trafic PFCP (non affecté par XDP)
tcpdump -i any -n udp port 8805 -w /tmp/pfcp.pcap

# Méthode 3 : Capture de paquet hors bande (RECOMMANDÉE pour GTP-
U)
# Utiliser un TAP réseau ou un mirroring de port de commutateur
pour capturer le trafic
# Exemples :
# - TAP physique entre gNB et UPF
# - Port SPAN/mirror de commutateur copiant le trafic N3 vers
l'analyseur
# - Mirroring de port de commutateur virtuel dans le hyperviseur
#
# Sur l'hôte de capture (PAS l'UPF) :
# tcpdump -i <mirror_interface> -n udp port 2152 -w
/tmp/n3_mirror.pcap
```

Exemples de Configuration de Capture Hors Bande :

Réseau Physique :

```
# Utiliser un TAP réseau ou configurer le mirroring de port de
commutateur
# Exemple : configuration SPAN de commutateur Cisco
(config)# monitor session 1 source interface Gi1/0/1
(config)# monitor session 1 destination interface Gi1/0/24

# Sur l'hôte de surveillance connecté à Gi1/0/24 :
tcpdump -i eth0 -n udp port 2152 -w /tmp/n3_capture.pcap
```

Environnement Virtuel (VMware, KVM, etc.) :

```
# Configurer le mirroring de port de commutateur virtuel pour
envoyer le trafic UPF à la VM d'analyse
# Exemple : pont Linux avec tcpdump sur une VM différente
# Sur le hyperviseur, mirroiter l'interface N3 de l'UPF vers
l'interface d'analyse

# Sur la VM d'analyse :
tcpdump -i eth1 -n udp port 2152 -w /tmp/n3_virtual.pcap
```

Pourquoi le Hors Bande est Nécessaire :

- XDP contourne entièrement la pile réseau du noyau
- Les paquets sont traités dans le pilote NIC ou le matériel
- `tcpdump` basé sur l'hôte voit les paquets APRÈS le traitement XDP (trop tard)
- La capture hors bande voit le trafic brut avant le traitement UPF

Ce que vous pouvez CAPTURER sur l'Hôte UPF :

- □ Trafic PFCP (UDP 8805) - plan de contrôle, non traité par XDP
- □ Réponses API et métriques
- □ Trafic GTP-U (UDP 2152) - plan de données, traité par XDP

Obtenir de l'Aide

Si les étapes de dépannage ne résolvent pas votre problème :

1. **Collecter des informations de diagnostic** :

```
# Informations système
uname -a
cat /etc/os-release

# Informations OmniUPF
curl http://localhost:8080/api/v1/upf_status
curl http://localhost:8080/api/v1/map_info
curl http://localhost:8080/api/v1/packet_stats

# Journaux
journalctl -u omniupf --since "1 hour ago" > /tmp/omniupf.log
dmesg > /tmp/dmesg.log

# Informations réseau
ip addr > /tmp/network.txt
ip route >> /tmp/network.txt
ethtool eth0 >> /tmp/network.txt
```

2. **Signaler le problème** avec :
    - Version d'OmniUPF
    - Version du noyau Linux
    - Diagramme de topologie réseau
    - Fichier de configuration (cacher les informations sensibles)
    - Extraits de journaux pertinents
    - Étapes pour reproduire

Documentation Connexe

- **Guide de Configuration** - Paramètres de configuration et exemples
- **Guide d'Architecture** - Internes eBPF/XDP et optimisation des performances
- **Guide de Surveillance** - Statistiques, capacité et alertes
- **Référence des Métriques** - Métriques Prometheus pour le dépannage
- **Codes de Cause PFCP** - Codes d'erreur PFCP et dépannage
- **Guide de Gestion des Règles** - Concepts PDR, FAR, QER, URR

- **Guide des Opérations** - Architecture et aperçu de l'UPF

OmniUPF Jardin Clos / Redirection Hors Crédit

Table des Matières

1. Aperçu
2. Architecture
3. Flux de Signalisation PFCP
4. Détection de Portail Captif
5. Configuration
6. Gestion de la Liste Blanche
7. URLs de Redirection par Session
8. API
9. Métriques Prometheus
10. Dépannage

Aperçu

La fonctionnalité Jardin Clos fournit une **application native de l'interdiction hors crédit** directement dans le UPF, éliminant ainsi le besoin de systèmes d'application externes (listes d'adresses MikroTik, règles de mangle, DNAT).

Lorsqu'un abonné n'a plus de crédit, le SMF envoie une Modification de Session PFCP avec un FAR contenant `redirect_information`. OmniUPF intercepte tout le trafic pour cette session dans l'espace utilisateur et applique une expérience de portail captif :

- **Les requêtes DNS** sont falsifiées pour renvoyer l'IP du serveur de portail, déclenchant la détection de portail captif sur tous les principaux appareils (Apple, Android, Windows)

- **Le trafic vers le portail et les IPs sur liste blanche** (processeurs de paiement, services CAPTCHA) est transmis normalement afin que l'abonné puisse recharger
- **Tout autre trafic** est silencieusement supprimé

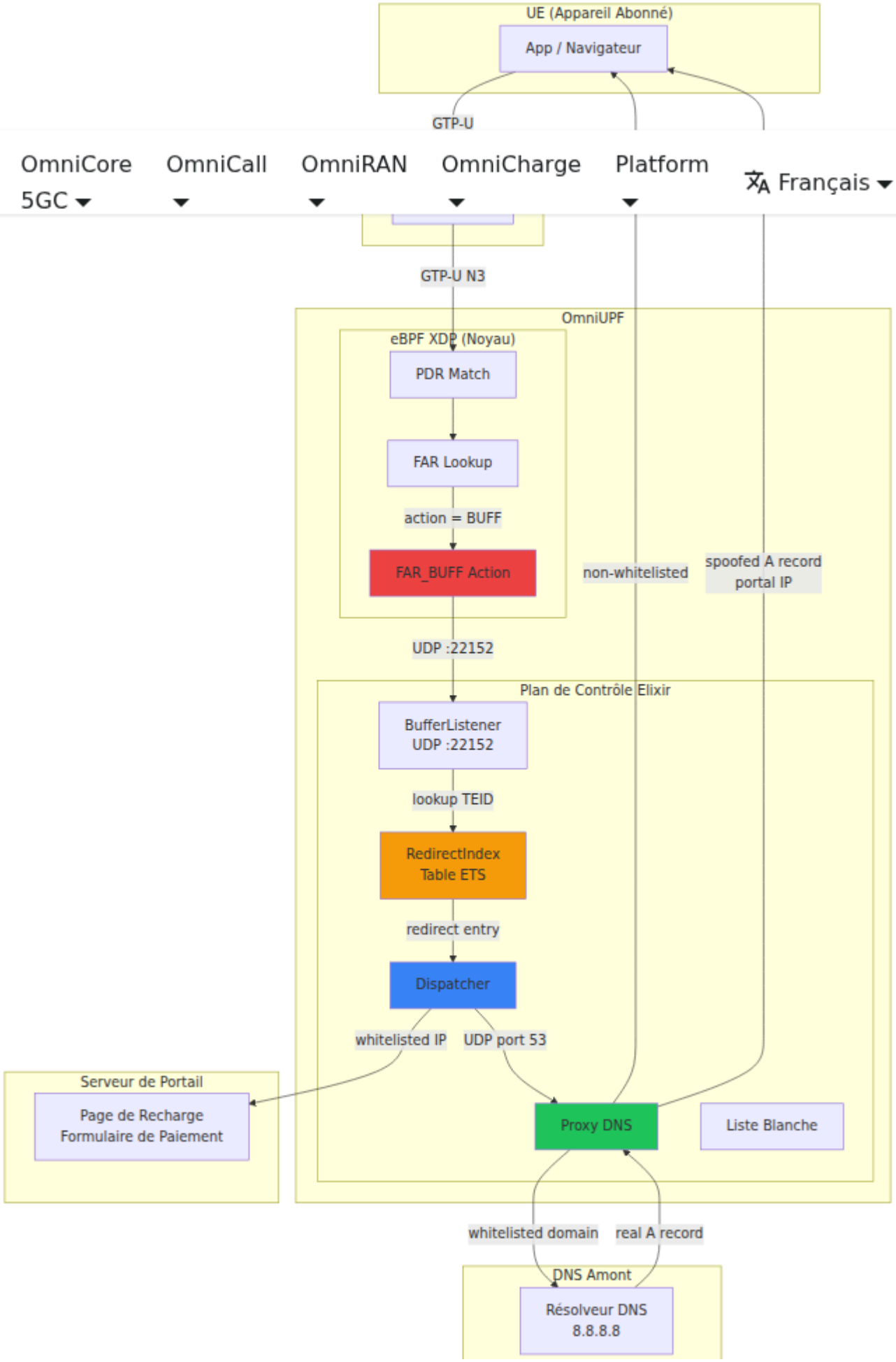
L'abonné voit une invite de portail captif et est dirigé vers une page de recharge/paiement. Une fois le crédit rétabli, le SMF met à jour le FAR de nouveau vers FORW et le transfert normal reprend immédiatement.

Points de Conception Clés

- **Aucun changement eBPF requis** -- réutilise l'action `FAR_BUFF` existante pour rediriger les paquets vers l'espace utilisateur
- **Redirection par session** -- chaque session peut avoir une IP de portail et une URL de redirection différentes, déterminées par l'IE `redirect_information` du SMF
- **La liste blanche vit sur le UPF** -- le SMF dit seulement "redirige cette session" ; le UPF décide quel trafic laisser passer
- **Interception uniquement en amont** -- le FAR de liaison descendante reste FORW afin que les réponses du portail atteignent l'UE via le chemin d'encapsulation GTP normal

---

Architecture


UE (Appareil Abonné)
App / Navigateur
GTP-U
GTP-U N3
OmniUPF
eBPF XDP (Noyau)
PDR Match
FAR Lookup
action = BUFF
FAR_BUFF Action
non-whitelisted
spoofed A record
portal IP
UDP :22152
Plan de Contrôle Elixir
BufferListener
UDP :22152
lookup TEID
RedirectIndex
Table ETS
redirect entry
Dispatcher
whitelisted IP
UDP port 53
Serveur de Portail
Page de Recharge
Formulaire de Paiement
Proxy DNS
Liste Blanche
whitelisted domain
real A record
DNS Amont
Résolveur DNS
8.8.8.8

Flux de Paquet

1. **L'UE envoie un paquet en amont** (requête DNS, requête HTTP, etc.) via GTP-U au UPF
2. **Le match PDR eBPF** trouve le PDR correspondant, recherche le FAR
3. **L'action FAR est BUFF** (remplacée de FORW lorsque la redirection est active) -- eBPF envoie le paquet au BufferListener sur le port UDP 22152
4. **BufferListener** extrait le TEID, vérifie la table ETS RedirectIndex
5. Si le TEID est dans le RedirectIndex : **Dispatcher** traite le paquet IP interne
6. **Arbre de décision** :
   - Requête DNS pour un domaine non sur liste blanche : falsifier l'enregistrement A avec l'IP du portail
   - Requête DNS pour un domaine sur liste blanche : transmettre au résolveur réel, mettre en cache les IPs résolues
   - Trafic vers l'IP du portail ou l'IP sur liste blanche : transmettre via socket brut
   - Tout le reste : supprimer silencieusement
7. **Les réponses DNS/GTP-U** sont renvoyées à l'UE via le chemin GTP-U descendant (encapsulé avec le TEID DL et envoyé au gNB)

Gestion de Boucle N9 (Session Dual SGW + PGW)

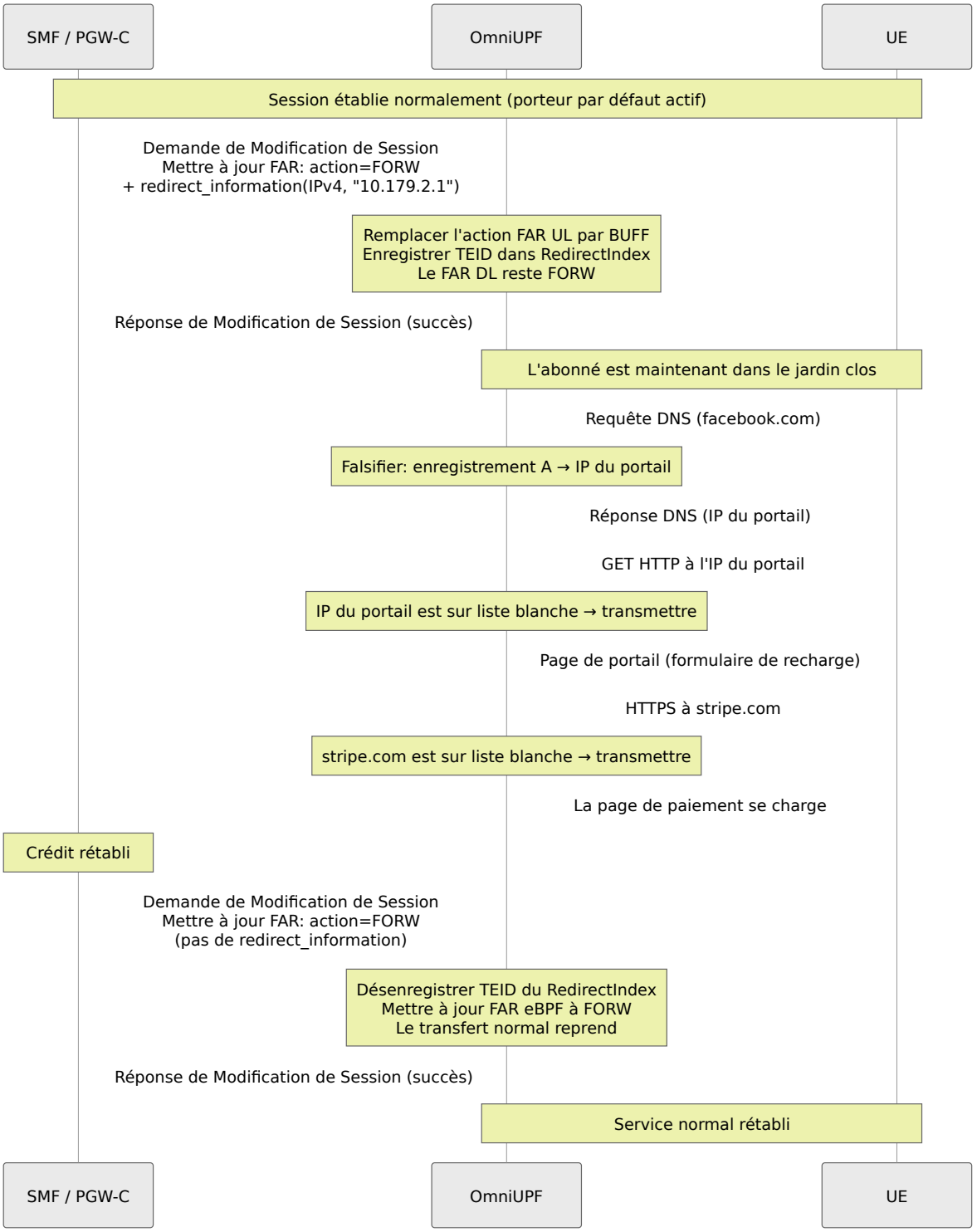
Dans les déploiements EPC 4G, OmniUPF agit souvent comme **SGW-U et PGW-U simultanément** sur le même nœud. Le SGW-C et le PGW-C établissent chacun une session PFCP distincte. La session SGW a un FAR N9 qui transmet les paquets vers le TEID de liaison montante de la session PGW (une boucle de retour via l'interface N3). La session PGW effectue l'application réelle de la politique d'abonné.

Lors de l'activation d'une redirection de jardin clos via `POST /v1/walled_garden`, la logique d'activation détecte automatiquement cette topologie :

1. **TEID(s) de session PGW** sont identifiés à partir des PDRs de liaison montante du SEID cible.
2. Le code parcourt toutes les sessions PFCP pour trouver une session (la session SGW) qui a un FAR dont le `teid` correspond à l'un des TEIDs de PDR UL PGW et a `outer_header_creation` défini. Ce FAR est le FAR de transfert N9 pointant vers le PGW.
3. **Les deux** FARs UL PGW et les FARs UL faisant face au gNB du SGW sont remplacés par `BUFF` dans la carte eBPF. Cela est nécessaire car le programme eBPF voit le paquet à son arrivée du eNB avec le TEID du SGW, pas le TEID du PGW.
4. Seules les **PDRs UL faisant face au gNB du SGW** sont enregistrées dans le RedirectIndex (pas les PDRs UL PGW). Le TEID de la PDR UL SGW est ce que eBPF verra dans le paquet entrant du eNB.
5. Pour le **chemin DL** (renvoyant les réponses à l'UE), le Dispatcher utilise le **FAR DL de la session SGW** (le FAR transmettant vers le eNB, avec `remoteip != n3_address`). Ce FAR contient le TEID eNB actuel et l'IP gNB. Le TEID DL est recherché en direct à partir de l'état de la session au moment de la réponse, pas mis en cache au moment de l'activation - cela gère le cas où le TEID eNB est attribué par une modification de session ultérieure après l'activation de la redirection.

Dans une topologie UPF directe (pas de SGW, SMF parle directement au UPF), seules les propres FARs et TEIDs UL de la session sont impliqués - le code de détection SGW ne trouve rien et les PDRs UL de la session principale sont enregistrées directement.

Flux de Signalisation PFCP



IEs PFCP Impliqués

Le SMF déclenche une redirection en incluant `redirect_information` dans les `forwarding_parameters` du FAR :

IE	Description
`apply_action`	Défini sur FORW par le SMF (UPF remplace par BUFF en interne)
`redirect_information`	Contient le type de redirection et l'adresse
`forwarding_parameters`	Contient le `redirect_information` et `outer_header_creation`

Types d'Informations de Redirection (selon 3GPP TS 29.244 Tableau 8.2.20-1) :

Type	Valeur	Comportement UPF
IPv4	0	Utiliser la chaîne d'adresse IPv4 fournie comme IP du portail
IPv6	1	Utiliser la chaîne d'adresse IPv6 fournie comme IP du portail
URL	2	Résoudre le nom d'hôte de l'URL en une IP ; utiliser cela comme IP du portail. Le chemin de l'URL n'est pas utilisé par le UPF -- il est stocké dans l'API uniquement pour visibilité. Le serveur web du portail est responsable de la gestion de tout routage basé sur le chemin.
SIP URI	3	Pas actuellement supporté

Détection de Portail Captif

Le jardin clos déclenche une **détection automatique de portail captif** sur toutes les principales plateformes d'appareils en falsifiant les réponses DNS.

Lorsqu'un appareil se connecte et tente de vérifier la connectivité Internet, il interroge des domaines bien connus. La réponse DNS falsifiée redirige ces vérifications vers l'IP du portail, que l'appareil interprète comme un portail captif.

Domaines de Détection de Plateforme

Plateforme	Domaine de Détection	Réponse Attendue
Apple (iOS/macOS)	`captive.apple.com`	HTTP 200 avec `<HTML><HEAD><TITLE>Succès</TITLE></HEAD><BODY>Succès</BODY></HTML>`
Android	`connectivitycheck.gstatic.com`	HTTP 204
Windows	`www.msftconnecttest.com`	HTTP 200 avec `Microsoft Connect Test`
Samsung	`connectivitycheck.samsung.com`	HTTP 200

Lorsque ces domaines se résolvent vers l'IP du portail au lieu de leurs vraies adresses, l'appareil détecte un portail captif et présente la page du portail à l'utilisateur soit sous forme de :

- Une notification/pop-up (iOS, Android)
- Une redirection automatique du navigateur (Windows)

Le serveur de portail à l'IP de portail configurée doit fournir des réponses appropriées pour ces URLs de détection puis rediriger vers la page de recharge/paiement.

Exigences du Serveur de Portail (IP Dédiée / vHost par Défaut)

Le portail doit servir la page de portail captif comme la réponse *par défaut* de son adresse IP, indépendamment de l'en-tête HTTP `Host`. En pratique, cela signifie donner au portail sa **propre IP dédiée**, ou le faire devenir le **vhost virtuel par défaut** sur un hôte partagé.

Cette exigence découle directement de la façon dont le jardin clos fonctionne :

1. Chaque requête DNS non sur liste blanche est **falsifiée vers l'IP du portail** -- y compris les sondes de portail captif de la plateforme (`connectivitycheck.gstatic.com`, `captive.apple.com`, `www.msftconnecttest.com`, ...).
2. L'UE ouvre ensuite HTTP vers l'IP du portail, mais la requête porte l'en-tête `Host` de la **sonde** (par exemple, `Host: connectivitycheck.gstatic.com`), pas le nom d'hôte du portail.
3. Le UPF transmet cette requête au serveur de portail **tel quel, sans réécrire l'en-tête `Host`** (voir Transfert de Socket Brut).

Si l'IP du portail est un **vhost virtuel basé sur le nom** -- c'est-à-dire que la page du portail n'est servie que pour une valeur `Host:` spécifique et que l'IP est partagée avec d'autres vhosts -- le serveur ne trouvera pas de vhost correspondant à l'en-tête `Host` de la sonde et servira son **vhost par défaut**, ce qui est la mauvaise page. L'abonné se retrouve sur tout autre contenu hébergé sur cette IP (une autre application, un CRM, un site par défaut vide) plutôt que sur la page de recharge.

Faites ceci :

- Donnez au portail captif sa **propre adresse IP** (tout `Host` servi par cette IP renvoie la page du portail), **ou**
- Faites du portail le **vhost virtuel par défaut** sur l'hôte partagé, de sorte que tout `Host:` non apparié tombe sur la page du portail.

Évitez : de pointer `walled_garden_portal_ip` (ou un `redirect_information` de type URL dont le nom d'hôte se résout) vers une IP dont la réponse HTTP

par défaut n'est *pas* la page du portail. Le UPF ne réalise pas de sélection de vhost basée sur `Host` ou de réécriture de `Host` au nom du portail.

Configuration

La configuration du jardin clos se trouve dans le fichier de configuration d'exécution. Sur une installation de production (`.deb` package), le fichier de configuration se trouve à `/etc/omniupf/runtime.exs`. Le script de démarrage de la version (`rel/env.sh`) vérifie ce fichier et, s'il est présent, définit `RELEASE_CONFIG_DIR=/etc/omniupf` afin que la version Erlang l'utilise plutôt que le `config/runtime.exs` inclus. Le UPF doit être redémarré après les modifications de configuration.

```
#
==========================================================================
# Jardin Clos / Redirection Hors Crédit
#
==========================================================================

# Activer l'application de redirection du jardin clos
walled_garden_enabled = true

# IP du serveur de portail (portail captif / page de recharge)
walled_garden_portal_ip = "10.179.2.1"

# Adresse du serveur de portail IPv6 optionnelle (solution de secours
les abonnés IPv6)
walled_garden_portal_ip6 = "fd00:179:2::1"

# Résolveur DNS en amont pour les recherches de domaines sur liste bl
walled_garden_dns_resolver = "8.8.8.8"

# Domaines sur liste blanche (les abonnés peuvent accéder à ceux-ci p
la redirection)
# Prend en charge les caractères génériques : "*.stripe.com" correspo
"api.stripe.com"
walled_garden_whitelist = [
  "stripe.com",
  "*.stripe.com",
  "js.stripe.com",
  "hcaptcha.com",
  "*.hcaptcha.com",
  "newassets.hcaptcha.com",
]
```

Paramètres

Paramètre	Type	Requis	Par Défaut	D
`walled_garden_enabled`	Booléen	Non	`false`	Interrup la foncticlos. Lor `redirec` dans les
`walled_garden_portal_ip`	Chaîne (IPv4)	Oui (si activé)	`10.179.2.1`	Adresse / serveu C'est l'IP secours nom d'h `redirec` SMF ne résolu, c de redire Pour les l'adresse utilisée le type U est réso la créati
`walled_garden_portal_ip6`	Chaîne (IPv6)	Non	`nil`	Adresse optionne comme pour les la même `walled_` est utilis Lorsqu'e définie, redirecti disponib

Paramètre	Type	Requis	Par Défaut	D
`walled_garden_dns_resolver`	Chaîne (IPv4)	Non	`8.8.8.8`	Résolve utilisé lo transmis pour les liste bla requête liste bla envoyée toutes le sont fals
`walled_garden_whitelist`	Liste de Chaînes	Non	`[]`	Modèles auxquel peuvent le jardin de la Lis syntaxe

---

Gestion de la Liste Blanche

La liste blanche contrôle quels domaines (et leurs IP résolues) les abonnés peuvent atteindre pendant la redirection. Cela est configuré sur le UPF, pas sur le SMF -- le SMF déclenche seulement la redirection.

Syntaxe des Modèles

Modèle	Correspond	Ne Correspond Pas
`stripe.com`	`stripe.com`, `api.stripe.com`, `js.stripe.com`	`evilstripe.com`, `notstripe.com`
`*.stripe.com`	`api.stripe.com`, `js.stripe.com`, `dashboard.stripe.com`	`stripe.com` (exact), `evilstripe.com`
`hcaptcha.com`	`hcaptcha.com`, `newassets.hcaptcha.com`	`evihcaptcha.com`

Les modèles utilisent **l'ancrage de sous-domaine** : `stripe.com` correspond au domaine lui-même et à tout sous-domaine (`foo.stripe.com`), mais pas aux domaines qui contiennent simplement la chaîne (`evilstripe.com`). La correspondance est insensible à la casse.

Mise en Cache des IP

Lorsque le proxy DNS transmet une requête pour un domaine sur liste blanche, les adresses IP résolues sont **automatiquement mises en cache** dans la liste blanche. Cela signifie :

1. L'abonné interroge `api.stripe.com`
2. Le proxy DNS transmet au résolveur réel, reçoit `203.0.113.7`
3. `203.0.113.7` est ajouté au cache IP sur liste blanche
4. Le trafic HTTP/HTTPS ultérieur vers `203.0.113.7` est transmis (non supprimé)

L'IP du portail est toujours sur liste blanche, quelle que soit la configuration.

Domaines Recommandés pour la Liste Blanche

Pour un portail de recharge typique avec traitement de paiement Stripe et hCaptcha :

```
walled_garden_whitelist = [
  # Processeur de paiement
  "stripe.com",
  "*.stripe.com",

  # Service CAPTCHA
  "hcaptcha.com",
  "*.hcaptcha.com",

  # Google Fonts (si le portail les utilise)
  "fonts.googleapis.com",
  "fonts.gstatic.com",

  # CDN pour les actifs du portail (si hébergé externement)
  "cdn.example.com",
]
```

URLs de Redirection par Session

Chaque session PFCP peut avoir une **cible de redirection différente**. Le SMF contrôle cela via l'IE `redirect_information` dans le FAR :

- **Type IPv4** : La chaîne IP fournie est analysée et utilisée comme IP du portail pour cette session
- **Type IPv6** : La chaîne IPv6 fournie est analysée et utilisée comme IP du portail pour cette session
- **Type URL** : Le nom d'hôte est extrait et résolu via DNS au moment de la création du FAR. L'IP résolue est utilisée comme IP du portail. Le chemin de l'URL n'est pas utilisé par le UPF -- il est stocké uniquement pour la visibilité de l'API.

Cela permet des scénarios où différents niveaux d'abonnés, MVNOs ou plans de service redirigent vers différents portails :

Session	SMF redirect_information	IP du Portail Utilisée	U
Session A	`IPv4: 10.179.2.1`	`10.179.2.1`	`10.179.`
Session B	`IPv6: 2001:db8::1`	`2001:db8::1`	`2001:db`
Session C	`URL: https://topup.mvno.com/recharge`	IP résolue de `topup.mvno.com`	`https:/`

Le UPF stocke l'IP du portail par session et l'URL de redirection, qui sont visibles via l'API.

API

Le chemin de base pour tous les points de terminaison du jardin clos est `/v1/walled_garden` (pas de préfixe `/api`). Ces points de terminaison sont servis par le serveur HTTP Phoenix sur le `api_port` configuré (par défaut : 8080).

GET /v1/walled_garden

Renvoie toutes les sessions de redirection de jardin clos actives, les IP sur liste blanche, les plages CIDR mises en cache et les détails de session.

Réponse :

```
{
  "redirect_count": 2,
  "redirects": [
    {
      "teid": "0x4000",
      "session_seid": 1,
      "portal_ip": "10.179.2.1",
      "redirect_url": null,
      "ue_ip": "10.60.0.1",
      "gnb_ip": "10.179.1.21",
      "dl_teid": "0x5000",
      "far_global_id": 42
    },
    {
      "teid": "0x4001",
      "session_seid": 2,
      "portal_ip": "10.179.2.2",
      "redirect_url": "https://topup.mvno.com",
      "ue_ip": "10.60.0.2",
      "gnb_ip": "10.179.1.21",
      "dl_teid": "0x5001",
      "far_global_id": 43
    }
  ],
  "whitelisted_ips": [
    {"ip": "10.179.2.1", "type": "portal"},
    {"ip": "203.0.113.7", "type": "resolved"},
    {"ip": "203.0.113.6", "type": "resolved"}
  ],
  "whitelisted_cidrs": ["192.168.0.0/24"]
}
```


Champs de Réponse :

Champ	Description
`redirect_count`	Nombre de sessions de jardin clos actives
`redirects[].teid`	TEID de liaison montante intercepté (hex)
`redirects[].session_seid`	SEID de session PFCP
`redirects[].portal_ip`	IP du portail pour cette session spécifique
`redirects[].redirect_url`	URL de redirection du SMF (si type URL), ou null
`redirects[].ue_ip`	Adresse IP de l'UE
`redirects[].gnb_ip`	IP gNB pour les réponses GTP-U
`redirects[].dl_teid`	TEID de liaison descendante pour l'encapsulation GTP-U
`redirects[].far_global_id`	ID global FAR interne
`whitelisted_ips`	Toutes les IP actuellement dans la liste blanche (portail + résolutions DNS mises en cache)
`whitelisted_cidrs`	Plages CIDR ajoutées via l'API de liste blanche

POST /v1/walled_garden

Active la redirection de jardin clos sur une session PFCP existante par SEID. C'est le chemin déclenché par l'opérateur/API - le chemin normal est via la Modification de Session PFCP avec `redirect_information` dans un FAR. Le chemin API est utile pour les tests et pour les opérateurs qui ont besoin de rediriger manuellement une session sans l'implication du SMF.

Corps de la requête :

```
{
  "seid": 1,
  "url": "http://10.179.2.1/"
}
```

Champ	Type	Requis	Description
`seid`	Entier	Oui	SEID local de la session PFCP à rediriger
`url`	Chaîne	Oui	Cible de redirection - une adresse IPv4/IPv6 ou une URL. Si une URL est fournie, le nom d'hôte est résolu et l'IP résultante est utilisée comme IP du portail. La chaîne URL complète est stockée pour visibilité API.

Réponse (200 OK) :

```
{
  "status": "redirection activée",
  "info": {
    "seid": 1,
    "sgw_seid": 7,
    "portal_ip": "10.179.2.1",
    "ue_ip": "10.60.0.1",
    "gnb_ip": "10.179.1.21",
    "ul_teids": ["0x4000", "0x4006"]
  }
}
```

Le champ `sgw_seid` est non nul lorsqu'une boucle de retour N9 (session jumelée SGW+PGW) a été détectée et que son TEID de liaison montante a également été BUFF'd. `ul_teids` liste tous les TEIDs de liaison montante qui ont été enregistrés dans l'index de redirection.

Réponses d'erreur :

Statut	Signification
404	Session non trouvée (SEID ne correspond à aucune session PFCP active)
400	Paramètres requis manquants

DELETE /v1/walled_garden/:seid

Désactive la redirection de jardin clos pour une session, restaurant le transfert normal. Tous les FARs de liaison montante pour la session (y compris les FARs de session jumelée SGW dans une topologie de boucle N9) sont remis à `action=FORW` dans la carte eBPF, et les TEIDs sont désenregistrés de l'index de redirection.

Paramètre de chemin : `:seid` - le SEID local de la session à désactiver.

Réponse (200 OK) :

```
{"status": "redirection supprimée", "info": {"seid": 1}}
```

Réponses d'erreur :

Statut	Signification
404	Session non trouvée

GET /v1/walled_garden/whitelist

Renvoie la liste blanche actuelle : IPs mises en cache individuelles (IP du portail et IPs résolues DNS) et toutes les plages CIDR ajoutées via l'API.

Réponse :

```
{
  "ips": [
    {"ip": "10.179.2.1", "type": "portal"},
    {"ip": "203.0.113.7", "type": "resolved"}
  ],
  "cidrs": ["192.168.100.0/24"]
}
```

POST /v1/walled_garden/whitelist

Ajoute une adresse IP ou une plage CIDR à la liste blanche en temps réel, sans redémarrer le UPF. Les modifications sont uniquement en mémoire et ne persistent pas entre les redémarrages - ajoutez des entrées permanentes à `walled_garden_whitelist` dans `runtime.exs`.

Corps de la requête (ajouter une IP) :

```
{"ip": "203.0.113.10"}
```

Corps de la requête (ajouter une plage CIDR) :

```
{"cidr": "192.168.100.0/24"}
```

Exactement un des `ip` ou `cidr` doit être présent. Ajouter une plage CIDR signifie que tout le trafic vers les IPs dans cette plage est transmis par le dispatcher sans nécessiter une entrée de cache DNS par IP.

Réponse (200 OK - IP ajoutée) :

```
{"status": "ajouté", "ip": "203.0.113.10"}
```

Réponse (200 OK - CIDR ajoutée) :

```
{"status": "ajouté", "cidr": "192.168.100.0/24"}
```

Réponses d'erreur :

Statut	Signification
400	IP invalide, CIDR invalide, ou champs de corps manquants

Métriques Prometheus

Jauges

Métrique: `upf_walled_garden_active_redirects` **Type**: Jauge **Description**: Nombre de sessions actuellement dans l'état de redirection de jardin clos

Exemples de requêtes:

```
# Nombre actuel de redirections
upf_walled_garden_active_redirects
```

Compteurs

Métrique: `upf_walled_garden_packets_intercepted_total` **Type**: Compteur **Description**: Total des paquets interceptés par le jardin clos (tout le trafic de liaison montante des sessions redirigées)

Métrique: `upf_walled_garden_packets_dropped_total` **Type**: Compteur **Description**: Total des paquets supprimés par le jardin clos (trafic non sur liste blanche, non-DNS)

Métrique: `upf_walled_garden_packets_forwarded_total` **Type**: Compteur **Labels**:

- `dst_ip` - IP de destination du paquet transmis **Description**: Paquets transmis à travers le jardin clos vers des destinations sur liste blanche. Étiqueté par IP de destination pour une visibilité par destination.

Métrique: `upf_walled_garden_bytes_forwarded_total` **Type**: Compteur **Labels**:

- `dst_ip` - IP de destination du trafic transmis **Description**: Octets transmis à travers le jardin clos par IP de destination. Utilisez ceci pour identifier quels services sur liste blanche les abonnés accèdent pendant la redirection.

Métrique: `upf_walled_garden_dns_spoofed_total` **Type**: Compteur **Labels**:

- `domain` - Le domaine qui a été falsifié **Description**: Requêtes DNS falsifiées par le jardin clos. Étiqueté par domaine interrogé.

Métrique: `upf_walled_garden_dns_forwarded_total` **Type**: Compteur **Labels**:

- `domain` - Le domaine sur liste blanche qui a été transmis **Description**: Requêtes DNS transmises au résolveur réel (domaines sur liste blanche). Étiqueté par domaine.

Exemples de Requêtes

```
# Sessions de jardin clos actives
upf_walled_garden_active_redirects

# Taux d'interception (paquets/sec)
rate(upf_walled_garden_packets_intercepted_total[5m])

# Taux de suppression (devrait être la majorité des interceptés)
rate(upf_walled_garden_packets_dropped_total[5m])

# Trafic transmis par destination (octets/sec)
sum by (dst_ip)
(rate(upf_walled_garden_bytes_forwarded_total[5m]))

# Top 5 des destinations sur liste blanche par volume de trafic
topk(5, sum by (dst_ip)
(rate(upf_walled_garden_bytes_forwarded_total[5m])))

# Domaines falsifiés les plus interrogés
topk(10, sum by (domain)
(rate(upf_walled_garden_dns_spoofed_total[5m])))

# Taux de recherche de domaine sur liste blanche
sum by (domain) (rate(upf_walled_garden_dns_forwarded_total[5m]))

# Ratio de paquets supprimés par rapport à ceux transmis
sum(rate(upf_walled_garden_packets_dropped_total[5m]))
  / sum(rate(upf_walled_garden_packets_intercepted_total[5m]))
```

Dépannage

Portail Captif Non Apparaissant sur l'Appareil

Symptômes: L'abonné est redirigé (visible dans l'API) mais l'appareil ne montre pas l'invite de portail captif.

Causes possibles:

- Serveur de portail ne répondant pas sur l'IP de portail configurée
- Serveur de portail ne gérant pas les URLs de détection spécifiques à la plateforme
- Réponse DNS n'atteignant pas l'UE (vérifiez le chemin GTP-U)

Résolution:

1. Vérifiez que le serveur de portail est accessible : `curl http://<portal_ip>/`
2. Confirmez que le portail gère les URLs de détection (par exemple, `GET /hotspot-detect.html` pour Apple)
3. Vérifiez `GET /v1/walled_garden` pour confirmer que la session est enregistrée
4. Vérifiez les métriques Prometheus : `upf_walled_garden_dns_spoofed_total` devrait s'incrémenter
5. Vérifiez que le chemin GTP-U descendant fonctionne (le FAR DL devrait rester FORW)

Page de Paiement Ne Charge Pas

Symptômes: L'abonné voit le portail captif mais ne peut pas accéder à la page de paiement (Stripe, etc.).

Causes possibles:

- Domaine du processeur de paiement non dans la liste blanche
- Processeur de paiement utilisant une IP CDN qui n'a pas été mise en cache
- Modèle de liste blanche ne correspondant pas correctement aux sous-domaines

Résolution:

1. Vérifiez `GET /v1/walled_garden` - vérifiez que `whitelisted_ips` inclut les IPs du processeur de paiement
2. Vérifiez Prometheus : `upf_walled_garden_dns_forwarded_total{domain="stripe.com"}` devrait montrer des recherches

3. Ajoutez les domaines manquants à la liste blanche (commun : `js.stripe.com`, `m.stripe.network`)
4. Vérifiez `upf_walled_garden_bytes_forwarded_total` par `dst_ip` pour voir quel trafic circule réellement

Redirection Ne S'Active Pas

Symptômes: Le SMF envoie une Modification de Session avec `redirect_information` mais le trafic de l'abonné n'est pas intercepté.

Causes possibles:

- `walled_garden_enabled` est `false`
- L'IE `redirect_information` est dans le mauvais FAR (doit être dans le FAR **de liaison montante**)
- Action FAR non remplacée par BUFF

Résolution:

1. Vérifiez la configuration : `walled_garden_enabled = true`
2. Vérifiez `GET /v1/pfcp_sessions` - regardez l'action FAR pour la session. Le FAR de liaison montante devrait montrer l'action `0x04` (BUFF)
3. Vérifiez `GET /v1/walled_garden` - le TEID de la session devrait apparaître dans la liste de redirection
4. Vérifiez les journaux UPF pour les messages `redirect_info` lors de la modification de session

Redirection Ne S'efface Pas Après Recharge

Symptômes: L'abonné a rechargé mais le trafic est toujours intercepté.

Causes possibles:

- Le SMF n'a pas envoyé la Modification de Session pour supprimer la redirection
- La transition BUFF->FORW n'est pas détectée

Résolution:

1. Vérifiez `GET /v1/walled_garden` - la session est-elle toujours listée ?
2. Vérifiez `GET /v1/pfcp_sessions` - vérifiez que l'action FAR a été mise à jour de nouveau vers `0x02` (FORW)
3. Vérifiez les journaux SMF pour confirmer qu'il a envoyé la Modification de Session
4. Vérifiez les journaux UPF pour le message "redirection supprimée, désenregistrement du jardin clos"

Mauvais TEID Descendant (Obsolète Après Modification de Session)

Symptômes: Les réponses falsifiées DNS ou le trafic transmis n'atteignent pas l'UE ; les paquets GTP-U sont envoyés à la bonne IP gNB mais le eNB les rejette comme TEID inconnu.

Cause: Dans les topologies 4G/SGW+PGW, le TEID eNB dans le FAR DL du SGW peut être zéro ou un TEID de remplacement au moment où la redirection de jardin clos est activée (par exemple, l'échange `Initial Context Setup` se produit après l'établissement de la session). Lorsque le eNB attribue son TEID et le renvoie via Modification de Session, le FAR DL dans la session SGW est mis à jour - mais si le Dispatcher avait mis en cache l'ancienne valeur, il utiliserait le mauvais TEID.

Résolution: Le Dispatcher résout le TEID DL **en direct** au moment de la réponse en recherchant le FAR DL actuel de la session SGW (le FAR avec `outer_header_creation` défini et `remoteip != n3_address`). Il ne revient à la valeur stockée au moment de l'activation que si la recherche de session échoue. Par conséquent, un TEID obsolète devrait se corriger dès que la prochaine réponse DNS ou transmise est envoyée. Si vous voyez toujours le mauvais TEID :

1. Vérifiez `GET /v1/walled_garden` - vérifiez que `dl_teid` dans l'entrée de redirection semble plausible (non nul).
2. Vérifiez `GET /v1/pfcp_sessions` - regardez les entrées FAR de la session SGW ; le FAR faisant face au gNB devrait avoir le `teid` actuel et `remoteip`.
3. Si la session SGW a été supprimée (par exemple, transfert ou libération), l'entrée de redirection restera dans le RedirectIndex mais la recherche en

direct échouera. Dans ce cas, utilisez `DELETE /v1/walled_garden/:seid` pour nettoyer l'entrée obsolète et permettre au SMF de rétablir.

Le Portail Renvoie 304 Non Modifié

Symptômes: Le navigateur de l'abonné affiche une page blanche ou la page du portail captif se charge une fois mais les visites suivantes apparaissent vides.

Cause: Les réponses HTTP 304 (Non Modifié) sont envoyées lorsque le navigateur a une version mise en cache de la page et que le serveur confirme que rien n'a changé. Pour les flux de portail captif, certains serveurs web de portail envoient 304 en réponse aux requêtes de détection de plateforme (`/hotspot-detect.html`, `/generate_204`, etc.) si le navigateur envoie des en-têtes `If-Modified-Since` ou `If-None-Match`. Certaines implémentations de portail captif envoient également 304 pour la page de redirection elle-même.

Résolution:

1. Le UPF transmet les réponses HTTP du serveur de portail de manière transparente - il ne modifie pas les codes de réponse. Le problème se trouve dans la configuration du serveur web du portail.
2. Sur le serveur de portail, assurez-vous que les points de terminaison de détection de plateforme renvoient le code de statut correct (200, pas 304) avec le contenu de corps attendu (voir Détection de Portail Captif).
3. Configurez le serveur de portail pour envoyer `Cache-Control: no-store` et omettre les en-têtes `ETag`/`Last-Modified` sur les points de terminaison de détection afin que les navigateurs ne les mettent pas en cache.
4. Vérifiez avec : `curl -v -H "If-None-Match: foo" http://<portal_ip>/hotspot-detect.html` - la réponse devrait être 200, pas 304.

Le Portail Affiche la Mauvaise Page (Vhost Basé sur le Nom)

Symptômes: L'abonné est redirigé (visible dans `GET /v1/walled_garden`) et l'invite de portail captif apparaît, mais en cliquant dessus, cela ouvre le **mauvais site** -- une autre application, un CRM, ou une page par défaut générique -- au lieu de la page de recharge/inscription. `curl` confirme que la page du portail ne se charge que si un en-tête `Host:` spécifique est présent :

```
curl -s -H 'Host: portal.example'
http://<portal_ip>/   # -> page de captif/inscription   (correct)
curl -s -H 'Host: connectivitycheck.gstatic.com'
http://<portal_ip>/   # -> vhost PAR DÉFAUT (faux)
curl -s -H 'Host: <portal_ip>'
http://<portal_ip>/   # -> vhost PAR DÉFAUT (faux)
```

Cause: L'IP du portail est un **vhost virtuel basé sur le nom** partagé avec d'autres sites. Le jardin clos falsifie chaque sonde de portail captif vers l'IP du portail et transmet la requête HTTP de l'UE **tel quel, en conservant l'en-tête `Host` d'origine de la sonde** (par exemple, `Host: connectivitycheck.gstatic.com`). Le serveur web du portail n'a pas de vhost correspondant à cet `Host`, donc il sert son **vhost par défaut** -- la mauvaise page. Le UPF ne réécrit pas l'en-tête `Host` ni ne fait de sélection de vhost au nom du portail. Voir Exigences du Serveur de Portail.

Résolution:

1. Donnez au portail captif sa **propre IP dédiée** afin que tout `Host:` servi par cette IP renvoie la page du portail, **ou**
2. Faites du portail le **vhost virtuel par défaut** sur l'hôte partagé, de sorte que les valeurs `Host:` non appariées tombent sur la page du portail.
3. Vérifiez avec les commandes `curl` ci-dessus : avec une IP dédiée ou un vhost par défaut, **les trois** requêtes devraient renvoyer la page du portail.

Transfert de Socket Brut

Lorsque le Dispatcher transmet un paquet à une IP sur liste blanche, il utilise un **socket brut** (`IPPROTO_RAW`, numéro de protocole 255) via le module `:socket` d'Erlang. Un nouveau socket brut est ouvert par paquet, le paquet IP interne complet (tel que reçu de eBPF) est envoyé avec `sendto`, et le socket est immédiatement fermé.

Comment cela fonctionne: Le paquet IP interne du payload GTP-U a déjà un en-tête IP valide avec l'IP de l'UE comme source et l'IP du serveur de destination comme destination. En injectant ce paquet via un socket brut en utilisant `IPPROTO_RAW`, le noyau le route en fonction de l'IP de destination en utilisant la table de routage de l'hôte. L'interface N6 du UPF doit avoir une route vers le serveur de portail/whitelist pour que cela fonctionne.

Problèmes courants :

Symptôme	Cause Probable	Solution
Le transfert échoue silencieusement	`EPERM` - le socket brut nécessite root ou `CAP_NET_RAW`	Assurez-vous qu'OmniUPF s'exécute en tant que root ou a la capacité `CAP_NET_RAW`
Les paquets sont transmis mais aucune réponse n'atteint l'UE	Route N6 vers le portail manquante	Ajoutez une route vers le sous-réseau du portail sur l'hôte UPF
`Walled garden: raw socket open failed` dans les journaux	Capacité manquante ou restriction du noyau	Vérifiez le service `systemd` `AmbientCapabilities=CAP_NET_RAW`
Le transfert fonctionne mais l'UE reçoit la mauvaise IP comme source	NAT sur N6 réécrit la source	Assurez-vous que les réponses du portail vers les IPs de l'UE sont routées de nouveau via le UPF

Vérifiez les journaux UPF pour les messages `Walled garden forward failed` et `raw socket open failed`. Utilisez les métriques Prometheus `upf_walled_garden_packets_forwarded_total` et `upf_walled_garden_bytes_forwarded_total` pour confirmer que le trafic circule.

Avertissement de Haute Cardinalité pour Prometheus

Remarque: Les étiquettes `dst_ip` et `domain` sur les métriques de jardin clos peuvent produire une haute cardinalité si de nombreuses destinations ou

domaines uniques sont interrogés. Dans les grands déploiements, envisagez d'utiliser des règles d'enregistrement pour agréger ces métriques :

```
# Règle d'enregistrement pour agréger par sous-réseau /24 au lieu
d'IP individuelle
sum by (dst_subnet) (
  label_replace(
    rate(upf_walled_garden_bytes_forwarded_total[5m]),
    "dst_subnet", "$1.0/24", "dst_ip", "(\d+\.\d+\.\d+)\.\d+"
  )
)
```

Guide des opérations de l'interface Web

Table des matières

1. Aperçu
2. Accès au panneau de contrôle
3. Vue des sessions
4. Gestion des règles
5. Gestion des tampons
6. Tableau de bord des statistiques
7. Surveillance de la capacité
8. Vue de configuration
9. Vue des routes
10. Vue des capacités XDP
11. Visionneuse de journaux

Aperçu

L'interface Web OmniUPF fournit un panneau de contrôle complet pour la surveillance et la gestion en temps réel de la fonction de plan utilisateur. L'interface est construite sur Phoenix LiveView et offre :

- **Visibilité en temps réel** sur les sessions PFCP et les connexions PDU actives
- **Inspection des règles** pour PDR, FAR, QER et URR à travers toutes les sessions
- **Gestion des tampons** pour le stockage de paquets lors d'événements de mobilité
- **Surveillance des statistiques** pour le traitement des paquets, les routes et les interfaces

- **Suivi de la capacité** pour l'utilisation et les limites des cartes eBPF
- **Visualisation des journaux en direct** pour le dépannage

Architecture

Le panneau de contrôle communique avec plusieurs instances OmniUPF via leur API REST pour :

- Interroger les sessions et associations PFCP
- Inspecter les règles de détection et de transfert de paquets
- Surveiller les tampons de paquets et leur statut
- Accéder aux statistiques en temps réel et aux métriques de performance
- Suivre la capacité et l'utilisation des cartes eBPF

Accès au panneau de contrôle

Accès par défaut

Le panneau de contrôle est accessible via HTTPS sur le serveur de gestion OmniUPF :

```
https://<upf-server>:443/
```

Port par défaut : 443 (HTTPS avec certificat auto-signé)

Configuration

Le panneau de contrôle nécessite la configuration de l'hôte OmniUPF dans `config/config.exs` :

Plusieurs instances UPF peuvent être configurées pour des déploiements multi-instances :

La configuration `upf_hosts` définit quelles instances OmniUPF sont disponibles dans le menu déroulant du sélecteur d'hôte à travers l'interface.

Navigation

Le panneau de contrôle fournit des onglets de navigation pour chaque domaine opérationnel :

- **Sessions** - `/sessions` - Sessions et associations PFCP
- **Règles** - `/rules` - Inspection des règles PDR, FAR, QER, URR
- **Tampons** - `/buffers` - Surveillance et contrôle des tampons de paquets
- **Statistiques** - `/statistics` - Statistiques sur les paquets, les routes, XDP et les interfaces
- **Capacité** - `/capacity` - Utilisation des cartes eBPF et surveillance de la capacité
- **Config** - `/upf_config` - Configuration UPF et adresses de plan de données
- **Routes** - `/routes` - Routes UE et sessions de protocole de routage (OSPF, BGP)
- **Capacités XDP** - `/xdp_capabilities` - Support du mode XDP et capacités de performance
- **Journaux** - `/logs` - Diffusion en direct des journaux

Vue des sessions

URL : `/sessions`

Fonctionnalités

La vue des sessions affiche toutes les sessions PFCP actives et les associations des instances OmniUPF sélectionnées.

Résumé des associations PFCP

Affiche toutes les associations PFCP actives (connexions de contrôle depuis SMF/PGW-C) :

Colonne	Description
ID de nœud	Identifiant de nœud SMF ou PGW-C (FQDN ou IP)
Adresse	Adresse IP SMF/PGW-C pour la communication PFCP
ID de session suivante	Prochain ID de session PFCP disponible pour cette association

Objectif :

- Vérifier la connectivité SMF vers UPF
- Surveiller le nombre de connexions de plan de contrôle
- Suivre l'allocation des ID de session par association

Tableau des sessions actives

Affiche toutes les sessions PFCP représentant des sessions PDU UE actives :

Colonne	Description
SEID local	Identifiant de point de terminaison de session attribué par UPF
SEID distant	Identifiant de point de terminaison de session attribué par SMF
IP UE	Adresse IPv4 ou IPv6 de l'équipement utilisateur
TEID	Identifiant de point de terminaison de tunnel GTP-U pour le trafic montant
PDRs	Nombre de règles de détection de paquets dans la session
FARs	Nombre de règles d'action de transfert dans la session
QERs	Nombre de règles d'application de QoS dans la session
URRs	Nombre de règles de rapport d'utilisation dans la session
Actions	Bouton d'expansion pour voir les informations détaillées sur les règles

Fonctionnalités :

- **Filtrer par IP** : Trouver des sessions pour une adresse IP UE spécifique
- **Filtrer par TEID** : Trouver des sessions par ID de point de terminaison de tunnel
- **Étendre la session** : Voir les détails JSON complets de PDR/FAR/QER/URR
- **Actualisation automatique** : Mises à jour toutes les 10 secondes

Vue de session étendue :

Lorsque vous cliquez sur "Étendre" sur une session, la vue montre :

- **Règles de détection de paquets (PDRs)** : JSON complet avec TEID, IP UE, ID FAR, ID QER, filtres SDF
  - **Les ID PDR sont cliquables** - Cliquez pour naviguer vers l'onglet Règles et voir les détails complets du PDR
  - Les PDRs montants (TEID ≠ 0) lient à la recherche de PDR montants
  - Les PDRs descendants (IPv4) lient à la recherche de PDR descendants
  - Les PDRs descendants (IPv6) lient à la recherche de PDR descendants IPv6
- **Règles d'action de transfert (FARs)** : Drapeaux d'action, création d'en-tête extérieur, points de terminaison de destination
- **Règles d'application de QoS (QERs)** : MBR, GBR, QFI et autres paramètres QoS
- **Règles de rapport d'utilisation (URRs)** : Compteurs de volume (montant, descendant, total d'octets)

Vue étendue de la session montrant les PDR, FAR et QER détaillés pour une session spécifique.

Cas d'utilisation

Vérifier la connectivité UE :

1. Naviguer vers la vue des sessions
2. Entrer l'adresse IP UE dans le filtre
3. Confirmer que la session existe avec le TEID correct
4. Étendre pour vérifier la configuration PDR/FAR

Surveiller le nombre de sessions :

- Vérifier le nombre total de sessions dans l'en-tête
- Comparer à travers plusieurs instances UPF
- Suivre la croissance des sessions au fil du temps

Dépanner les problèmes de session :

- Rechercher une IP UE ou un TEID spécifique
- Étendre la session pour inspecter la configuration des règles
- Vérifier les paramètres de transfert FAR
- Vérifier les paramètres QoS QER

Mises à jour en temps réel

La vue des sessions se rafraîchit automatiquement toutes les 10 secondes. Un indicateur de vérification de santé montre l'état de connectivité UPF :

- **SAIN** (vert) : UPF est accessible et répond
- **NON SAIN** (rouge) : UPF n'est pas accessible ou ne répond pas
- **INCONNU** (gris) : État de santé pas encore déterminé

Gestion des règles

URL : `/rules`

La vue des règles fournit une inspection complète de toutes les règles de détection de paquets, de transfert, d'application de QoS et de rapport

d'utilisation à travers toutes les sessions.

Onglet PDR - Règles de détection de paquets

Voir et inspecter tous les PDR dans le UPF avec **formulaires de recherche** et **navigation cliquable** :

PDRs montants (N3 → N6) :

- **Formulaire de recherche** : Rechercher par TEID pour voir les détails spécifiques du PDR montant
- **TEID** : ID de point de terminaison de tunnel GTP-U depuis gNB (cliquable - navigue vers la recherche)
- **ID FAR** : Règle d'action de transfert associée (cliquable - navigue vers l'onglet FAR)
- **ID QER** : Règle d'application de QoS associée (cliquable - navigue vers l'onglet QER)
- **ID URR** : Règles de rapport d'utilisation associées (cliquable - navigue vers l'onglet URR)
- **Suppression d'en-tête extérieur** : Drapeau de décapsulation GTP-U
- **Filtres SDF** : Règles de classification de flux de données de service

PDRs descendants (N6 → N3) :

- **Formulaire de recherche** : Rechercher par adresse IPv4 de l'UE pour voir les détails spécifiques du PDR descendant
- **IP UE** : Adresse IPv4 de l'équipement utilisateur (affichée dans les résultats de recherche)
- **ID FAR** : Règle d'action de transfert associée (cliquable - navigue vers l'onglet FAR)
- **ID QER** : Règle d'application de QoS associée (cliquable - navigue vers l'onglet QER)
- **ID URR** : Règles de rapport d'utilisation associées (cliquable - navigue vers l'onglet URR)
- **Mode SDF** : Mode de filtre de flux de données de service (aucun, sdf uniquement, sdf + par défaut)

- **Pagination** : Parcourir les PDR avec des contrôles de page (100 par page par défaut, maximum 1000)

PDRs descendants IPv6 :

- L'API prend en charge la pagination pour les PDRs descendants IPv6
- Même structure que pour IPv4 mais indexée par adresses IPv6
- Un onglet UI complet peut être ajouté si nécessaire

Onglet FAR - Règles d'action de transfert

Voir tous les FARs avec leurs actions de transfert et paramètres :

Fonctionnalités :

- **Formulaire de recherche** : Rechercher par ID FAR pour voir des détails spécifiques sur le FAR
- **Recherche automatique** : Cliquer sur les ID FAR depuis les détails PDR remplit automatiquement la recherche
- **Mises à jour en temps réel** : L'état du FAR reflète l'état actuel du tampon

Colonne	Description
ID FAR	Identifiant unique de la règle de transfert
Action	Drapeaux d'action de transfert (FORWARD, DROP, BUFFER, DUPLICATE, NOTIFY)
Tamponnage	État actuel du tampon (Activé/Désactivé)
Destination	Paramètres de création d'en-tête extérieur (TEID, adresse IP)

Drapeaux d'action FAR :

- **FORWARD (1)** : Transférer le paquet à la destination
- **DROP (2)** : Jeter le paquet

- **BUFFER (4)** : Stocker le paquet dans le tampon
- **NOTIFY (8)** : Envoyer une notification au plan de contrôle
- **DUPLICATE (16)** : Dupliquer le paquet vers plusieurs destinations

Basculer le tamponnage :

- Cliquez sur "Activer le tampon" ou "Désactiver le tampon" pour basculer le drapeau de tamponnage
- Utile pour le dépannage des scénarios de transfert
- Modifie immédiatement l'action FAR dans la carte eBPF

Onglet QER - Règles d'application de QoS

Voir les règles QoS appliquées aux flux de trafic :

Fonctionnalités :

- **Navigation cliquable** : Cliquez sur les ID QER depuis les détails PDR pour naviguer et mettre en surbrillance un QER spécifique
- **Auto-surlignage** : La ligne QER est surlignée lorsqu'elle est naviguée depuis un PDR
- **Pagination** : Parcourir les QERs avec des contrôles de page (100 par page par défaut, maximum 1000)

Colonne	Description
ID QER	Identifiant unique de la règle QoS (cliquable lorsqu'il est référencé depuis les PDRs)
MBR (Montant)	Débit maximum pour le trafic montant (kbps)
MBR (Descendant)	Débit maximum pour le trafic descendant (kbps)
GBR (Montant)	Débit garanti pour le trafic montant (kbps)
GBR (Descendant)	Débit garanti pour le trafic descendant (kbps)
QFI	Identifiant de flux QoS (marquage 5G)

Interprétation de la QoS :

- **MBR = 0** : Pas de limite de débit
- **GBR = 0** : Meilleure effort (pas de bande passante garantie)
- **GBR > 0** : Flux à débit garanti (priorisé)

Onglet URR - Règles de rapport d'utilisation

Voir les règles de suivi d'utilisation et les compteurs de volume :

Fonctionnalités :

- **Formulaire de recherche** : Rechercher par ID URR pour trouver et mettre en surbrillance un URR spécifique
- **Navigation cliquable** : Cliquez sur les ID URR depuis les détails PDR pour naviguer et mettre en surbrillance un URR spécifique
- **Auto-surlignage** : La ligne URR est surlignée en bleu lorsqu'elle est naviguée depuis un PDR ou recherchée via la recherche
- **Pagination** : Parcourir les URRs avec des contrôles de page (100 par page par défaut, maximum 1000)

Colonne	Description
ID URR	Identifiant unique de la règle de rapport d'utilisation (cliquable lorsqu'il est référencé depuis les PDRs)
Volume montant	Octets envoyés de l'UE vers le réseau de données
Volume descendant	Octets envoyés du réseau de données vers l'UE
Volume total	Total d'octets dans les deux directions
Actions	Bouton de suppression pour réinitialiser les compteurs pour cet URR

Affichage du volume :

- Formaté automatiquement (B, Ko, Mo, Go, To)
- Compteurs en temps réel mis à jour à chaque actualisation
- Utilisé pour la facturation et l'analyse

Filtrage :

- Affiche uniquement les URRs avec un volume non nul
- Les URRs inactifs (tous les compteurs à 0) sont filtrés pour des performances optimales

Cas d'utilisation

Inspecter la classification du trafic :

1. Naviguer vers Règles → onglet PDR
2. Rechercher un TEID ou une IP UE spécifique
3. Vérifier que le PDR s'associe au FAR et au QER corrects

Dépanner les problèmes de transfert :

1. Naviguer vers Règles → onglet FAR
2. Localiser l'ID FAR depuis le PDR de session
3. Vérifier que l'action est FORWARD (pas DROP ou BUFFER)
4. Vérifier les paramètres de création d'en-tête extérieur

Surveiller l'application de la QoS :

1. Naviguer vers Règles → onglet QER
2. Vérifier que les valeurs MBR et GBR correspondent à la politique
3. Vérifier le marquage QFI pour les flux 5G

Suivre l'utilisation des données :

1. Naviguer vers Règles → onglet URR
2. Trier par volume total pour trouver les utilisateurs les plus élevés
3. Surveiller la croissance du volume au fil du temps
4. Vérifier l'intégration de la facturation

Gestion des tampons

URL : `/buffers`

Fonctionnalités

La vue des tampons affiche les tampons de paquets maintenus par le UPF lors d'événements de mobilité ou de changements de chemin.

Statistiques totales

Le tableau de bord affiche des statistiques agrégées des tampons :

- **Total des paquets** : Nombre de paquets mis en tampon à travers tous les FARs
- **Total des octets** : Taille totale des données mises en tampon
- **Total des FARs** : Nombre de FARs avec des paquets mis en tampon
- **Max par FAR** : Nombre maximum de paquets autorisés par FAR
- **Max total** : Nombre maximum total de paquets mis en tampon

- **TTL des paquets** : Durée de vie des paquets mis en tampon (secondes)

Tampons par FAR

Tableau de tous les FARs avec des paquets mis en tampon :

Colonne	Description
ID FAR	Identifiant de la règle d'action de transfert
Nombre de paquets	Nombre de paquets mis en tampon pour ce FAR
Nombre d'octets	Total d'octets mis en tampon pour ce FAR
Paquet le plus ancien	Horodatage du paquet mis en tampon le plus ancien
Paquet le plus récent	Horodatage du paquet mis en tampon le plus récent
Actions	Boutons de contrôle des tampons (style pilule)

Actions de contrôle des tampons

Pour chaque FAR avec des paquets mis en tampon, les boutons de style pilule suivants sont disponibles :

Contrôle du tamponnage :

- **Désactiver le tampon** (rouge) : Désactiver le tamponnage pour ce FAR (met à jour le drapeau d'action FAR)
- **Activer le tampon** (violet) : Activer le tamponnage pour ce FAR

Opérations sur les tampons :

- **Vider** (bleu) : Rejouer tous les paquets mis en tampon en utilisant les règles FAR actuelles
- **Effacer** (gris) : Supprimer tous les paquets mis en tampon sans transfert

Effacer tous les tampons :

- Bouton rouge "Effacer tout" dans l'en-tête
- Efface les tampons pour tous les FARs
- Nécessite une confirmation

Cas d'utilisation

Surveiller le tamponnage lors des transferts :

1. Pendant le transfert, vérifier que les paquets sont mis en tampon
2. Vérifier l'état de tamponnage du FAR (doit être activé)
3. Surveiller le nombre de paquets et leur ancienneté

Compléter le transfert :

1. Après le changement de chemin, cliquer sur "Vider" pour rejouer les paquets mis en tampon
2. Vérifier que les paquets sont transférés vers le nouveau chemin
3. Cliquer sur "Désactiver le tampon" pour arrêter le tamponnage

Effacer les tampons bloqués :

1. Identifier les FARs avec des paquets mis en tampon anciens (vérifier l'horodatage le plus ancien)
2. Cliquer sur "Effacer" pour jeter les paquets obsolètes
3. Ou cliquer sur "Désactiver le tampon" pour empêcher un tamponnage supplémentaire

Dépanner les débordements de tampon :

1. Vérifier le nombre total de paquets par rapport au maximum total
2. Identifier les FARs avec un tamponnage excessif
3. Vérifier que le SMF a envoyé une modification de session pour désactiver le tamponnage
4. Désactiver manuellement le tamponnage si la commande SMF a été manquée

Mises à jour en temps réel

La vue des tampons se rafraîchit automatiquement toutes les 5 secondes pour montrer l'état actuel des tampons.

Tableau de bord des statistiques

URL : `/statistics`

Fonctionnalités

La vue des statistiques fournit des métriques de performance en temps réel du plan de données OmniUPF. Pour des informations détaillées sur les métriques Prometheus, voir la **Référence des métriques**.

Statistiques des paquets

Compteurs agrégés de traitement des paquets :

- **Paquets RX** : Total des paquets reçus sur toutes les interfaces
- **Paquets TX** : Total des paquets transmis sur toutes les interfaces
- **Paquets perdus** : Paquets rejetés en raison d'erreurs ou de politiques
- **Paquets GTP-U** : Paquets traités avec encapsulation GTP-U

Utilisation : Surveiller la charge de trafic globale du UPF et le taux de perte de paquets

Statistiques des routes

Métriques de transfert par route (si disponibles) :

- **Hits de route** : Paquets correspondant à chaque règle de routage
- **Succès de transfert** : Nombre de paquets transférés avec succès
- **Erreurs de transfert** : Tentatives de transfert échouées

Utilisation : Identifier les routes chargées et les erreurs de transfert

Statistiques XDP

Métriques de performance du chemin de données eXpress (XDP) :

- **XDP Traités** : Total des paquets traités au niveau XDP
- **XDP Passés** : Paquets envoyés à la pile réseau
- **XDP Perdus** : Paquets perdus au niveau XDP
- **XDP Abandonnés** : Erreurs de traitement dans le programme XDP

Utilisation : Surveiller la performance XDP et détecter les erreurs de traitement

Causes de perte XDP :

- Format de paquet invalide
- Échec de recherche dans la carte eBPF
- Pertes basées sur des politiques
- Épuisement des ressources

Statistiques des interfaces N3/N6

Compteurs de trafic par interface :

Interface N3 (connectivité RAN) :

- **RX N3** : Paquets reçus de gNB/eNodeB
- **TX N3** : Paquets transmis à gNB/eNodeB

Interface N6 (connectivité au réseau de données) :

- **RX N6** : Paquets reçus du réseau de données (Internet/IMS)
- **TX N6** : Paquets transmis au réseau de données

Total : Compte agrégé de paquets à travers les interfaces

Utilisation : Surveiller l'équilibre du trafic et les problèmes spécifiques à l'interface

Cas d'utilisation

Surveiller la charge de trafic :

1. Vérifier les taux RX/TX de paquets
2. Vérifier que le trafic circule dans les deux directions
3. Comparer le trafic N3 par rapport au trafic N6 (devrait être à peu près égal)

Détecter les pertes de paquets :

1. Vérifier le compteur de paquets perdus
2. Examiner le compteur de paquets perdus XDP
3. Enquêter sur la cause dans les journaux si les pertes sont élevées

Analyse de performance :

1. Surveiller le ratio de paquets traités par rapport à ceux passés par XDP
2. Vérifier les abandons XDP (indique des erreurs)
3. Vérifier la distribution du trafic des interfaces N3/N6

Planification de capacité :

1. Suivre le taux de paquets au fil du temps
2. Comparer aux limites de capacité du UPF
3. Planifier une montée en charge si les limites sont approchées

Mises à jour en temps réel

Les statistiques se rafraîchissent automatiquement toutes les 5 secondes.

Surveillance de la capacité

URL : `/capacity`

Fonctionnalités

La vue de capacité affiche l'utilisation des cartes eBPF et les limites de capacité pour toutes les cartes dans le chemin de données UPF.

Tableau d'utilisation des cartes eBPF

Tableau de toutes les cartes eBPF avec des informations d'utilisation :

Colonne	Description
Nom de la carte	Nom de la carte eBPF (par exemple, `uplink_pdr_map`, `far_map`)
Utilisé	Nombre d'entrées actuellement dans la carte
Capacité	Nombre maximum d'entrées autorisées dans la carte
Utilisation	Barre de progression visuelle avec pourcentage
Taille de la clé	Taille des clés de la carte en octets
Taille de la valeur	Taille des valeurs de la carte en octets

Indicateurs d'utilisation codés par couleur

La barre de progression d'utilisation est codée par couleur en fonction de l'utilisation :

- **Vert (<50%)** : Fonctionnement normal, capacité ample
- **Jaune (50-70%)** : Prudence, surveiller la croissance
- **Amber (70-90%)** : Avertissement, planifier une augmentation de capacité
- **Rouge (>90%)** : Critique, action immédiate requise

Cartes critiques à surveiller

uplink_pdr_map :

- Stocke les PDRs montants indexés par TEID
- Une entrée par flux de trafic montant
- **Critique** : L'épuisement empêche l'établissement de nouvelles sessions

downlink_pdr_map / **downlink_pdr_map_ip6** :

- Stocke les PDRs descendants indexés par adresse IP UE
- Une entrée par adresse IPv4/IPv6 de l'UE
- **Critique** : L'épuisement empêche l'établissement de nouvelles sessions

far_map :

- Stocke les règles d'action de transfert indexées par ID FAR
- Partagé entre plusieurs PDRs
- **Haute priorité** : Affecte les décisions de transfert

qer_map :

- Stocke les règles d'application de QoS indexées par ID QER
- **Priorité moyenne** : Affecte la QoS mais pas la connectivité de base

urr_map :

- Stocke les règles de rapport d'utilisation indexées par ID URR
- **Priorité basse** : Affecte la facturation mais pas la connectivité

Cas d'utilisation

Planification de capacité :

1. Surveiller les tendances d'utilisation des cartes au fil du temps
2. Identifier quelles cartes croissent le plus rapidement
3. Planifier des augmentations de capacité avant d'atteindre les limites

Prévenir les échecs d'établissement de session :

1. Vérifier l'utilisation de la carte PDR avant une augmentation de trafic attendue
2. Augmenter la capacité de la carte si elle approche des limites
3. Surveiller après l'augmentation de capacité pour vérifier

Dépanner les échecs de session :

1. Lorsque l'établissement de session échoue, vérifier la vue de capacité

2. Si les cartes PDR sont rouges (>90%), la capacité est épuisée
3. Augmenter la capacité de la carte ou effacer les sessions obsolètes

Optimiser la configuration des cartes :

1. Examiner les tailles de clé et de valeur
2. Calculer l'utilisation de la mémoire par carte
3. Optimiser les tailles de carte en fonction des modèles d'utilisation réels

Configuration de capacité

Les capacités des cartes eBPF sont configurées au démarrage du UPF dans le fichier de configuration UPF. Valeurs typiques :

- Petit déploiement : 10 000 - 100 000 entrées par carte
- Déploiement moyen : 100 000 - 1 000 000 entrées par carte
- Grand déploiement : 1 000 000+ entrées par carte

Calcul de mémoire :

```
Mémoire de la carte = (Taille de clé + Taille de valeur) ×
Capacité
```

Par exemple, une carte PDR avec 1 million d'entrées et des valeurs de 64 octets utilise environ 64 Mo de mémoire du noyau.

Mises à jour en temps réel

La vue de capacité se rafraîchit automatiquement toutes les 10 secondes.

Vue de configuration

URL : `/upf_config`

Fonctionnalités

La vue de configuration affiche les paramètres opérationnels du UPF et la configuration du plan de données.

Configuration UPF

Affiche la configuration statique du UPF :

- **Interface PFCP** : Adresse IP et port pour la connectivité SMF/PGW-C
- **Interface N3** : Adresse IP pour la connectivité RAN (gNB/eNodeB)
- **Interface N6** : Adresse IP pour la connectivité au réseau de données
- **Interface N9** : Adresse IP pour la communication inter-UPF (optionnelle)
- **Port API** : Port d'écoute de l'API REST
- **Version** : Version du logiciel OmniUPF

Configuration du plan de données (eBPF)

Affiche les paramètres actifs du plan de données en temps réel :

- **Adresse N3 active** : Liaison de l'interface N3 en temps réel
- **Adresse N9 active** : Liaison de l'interface N9 en temps réel (si activée)

Ces valeurs reflètent la configuration réelle du chemin de données eBPF et peuvent différer de la configuration statique si les interfaces ont été modifiées.

Cas d'utilisation

Vérifier la connectivité UPF :

1. Vérifier que l'adresse IP de l'interface N3 correspond à la configuration gNB
2. Vérifier que l'interface N6 peut acheminer vers le réseau de données
3. Confirmer que l'interface PFCP est accessible depuis le SMF

Dépanner les problèmes d'interface :

1. Comparer la configuration statique avec les adresses actives du plan de données

2. Vérifier que les interfaces sont correctement liées
3. Vérifier les modifications de configuration des interfaces

Documentation et audit :

1. Enregistrer la configuration UPF pour documentation
2. Vérifier que le déploiement correspond aux spécifications de conception
3. Auditer les affectations d'interface

Vue des routes

URL : `/routes`

Fonctionnalités

La vue des routes fournit une surveillance complète des routes IP de l'équipement utilisateur (UE) et des sessions de protocole de routage (OSPF et BGP).

Aperçu de l'état des routes

Le tableau de bord affiche des statistiques agrégées des routes :

- **État** : Routage activé ou désactivé
- **Total des routes** : Nombre total de routes IP UE
- **Synchronisé** : Nombre de routes synchronisées avec succès
- **Échoué** : Nombre de routes qui ont échoué à se synchroniser

Routes IP UE actives

Tableau affichant toutes les routes IP de l'équipement utilisateur actives :

Colonne	Description
Index	Numéro d'index de la route
Adresse IP UE	Adresse IPv4 ou IPv6 assignée à l'UE

Objectif :

- Voir toutes les adresses IP UE qui ont des routes configurées
- Vérifier la distribution des routes aux protocoles de routage
- Surveiller l'état de synchronisation des routes

Voisins OSPF

Tableau des voisins du protocole OSPF (Open Shortest Path First) :

Colonne	Description
ID de voisin	Identifiant du routeur OSPF
Adresse	Adresse IP du voisin OSPF
Interface	Interface utilisée pour l'adjacence OSPF
État	État d'adjacence OSPF (Full, Init, etc.)
Priorité	Valeur de priorité OSPF
Temps de fonctionnement	Durée pendant laquelle le voisin est actif
Temps mort	Temps avant que le voisin soit considéré comme mort

États OSPF :

- **Full** (vert) : Complètement adjacent et échangeant des informations de routage
- **Autres états** (jaune) : Adjacence en formation ou incomplète

Pairs BGP

Tableau des pairs BGP (Border Gateway Protocol) :

Colonne	Description
Adresse IP de voisin	Adresse IP du pair BGP
ASN	Numéro de système autonome du pair
État	État de session BGP (Établi, Inactif, etc.)
Up/Down	Durée de l'état actuel
Préfixes reçus	Nombre de préfixes de route reçus du pair
Msg envoyés	Total des messages BGP envoyés au pair
Msg reçus	Total des messages BGP reçus du pair

États BGP :

- **Établi** (vert) : Session BGP active, échangeant des routes
- **Autres états** (rouge) : Session inactive ou en cours d'établissement

L'en-tête affiche également l'ID de routeur BGP local et l'ASN lorsque BGP est configuré.

Routes redistribuées OSPF

Tableau montrant les LSA externes OSPF (Link State Advertisements) pour les routes UE redistribuées :

Colonne	Description
ID d'état de lien	Identifiant LSA (généralement l'adresse réseau)
Masque	Masque de réseau pour la route
Routeur annonçant	ID du routeur annonçant cette route externe
Type de métrique	Type de métrique externe OSPF (E1 ou E2)
Métrique	Coût de métrique OSPF pour la route
Âge	Temps écoulé depuis que le LSA a été originaire (secondes)
Numéro de séquence	Numéro de séquence LSA pour le versionnage

Objectif :

- Vérifier que les routes UE sont redistribuées dans OSPF
- Surveiller quel routeur annonce des routes externes
- Suivre l'âge et les mises à jour des LSA

Actions de contrôle des routes

Bouton Synchroniser les routes :

- Déclenche manuellement la synchronisation des routes vers FRR (Free Range Routing)
- Force la mise à jour du protocole de routage avec les routes UE actuelles
- Utile après des modifications de configuration ou pour récupérer des échecs de synchronisation

Bouton Actualiser :

- Actualiser manuellement toutes les informations de route

- Met à jour les voisins OSPF, les pairs BGP et les tables de routage

Cas d'utilisation

Surveiller la santé du protocole de routage :

1. Naviguer vers la vue des routes
2. Vérifier les états des voisins OSPF (doivent être "Full")
3. Vérifier que les pairs BGP sont "Établis"
4. Confirmer le nombre attendu de voisins/pairs

Vérifier la distribution des routes UE :

1. Vérifier le tableau des routes IP UE actives pour un UE spécifique
2. Faire défiler vers la section des routes redistribuées OSPF
3. Vérifier que la route UE apparaît dans les LSA externes
4. Confirmer que le routeur annonçant correspond à l'UPF attendu

Dépanner les problèmes de synchronisation des routes :

1. Vérifier les compteurs Synchronisés vs. Échoués dans l'aperçu de l'état
2. Si les routes échouent, cliquer sur le bouton "Synchroniser les routes"
3. Surveiller les messages d'erreur dans la bannière rouge si la synchronisation échoue
4. Vérifier les messages d'erreur OSPF/BGP dans les sections respectives

Vérifier le déploiement multi-UPF :

1. Sélectionner différentes instances UPF dans le menu déroulant
2. Comparer les comptes de routes entre les instances
3. Vérifier que les voisins OSPF se voient
4. Vérifier les relations de pairage BGP

Surveiller l'évolutivité des routes :

1. Suivre le compte total de routes à mesure que les sessions UE augmentent
2. Vérifier que les routes sont distribuées aux protocoles de routage
3. Surveiller la croissance du compte LSA OSPF

4. Vérifier le compte de préfixes BGP reçus par les pairs

Mises à jour en temps réel

La vue des routes se rafraîchit automatiquement toutes les 10 secondes pour montrer l'état actuel du protocole de routage et les routes UE.

Intégration de routage

La vue des routes s'intègre avec FRR (Free Range Routing) exécuté sur le UPF :

- **OSPF** : Les routes sont redistribuées en tant que LSA externes de type 2
- **BGP** : Les routes sont annoncées aux pairs BGP configurés
- **Mécanisme de synchronisation** : Les appels API REST déclenchent des commandes vtysh pour mettre à jour FRR

Vue des capacités XDP

URL : `/xdp_capabilities`

Fonctionnalités

La vue des capacités XDP affiche le support des modes eXpress Data Path (XDP), les capacités de performance et les calculs de débit pour le plan de données UPF.

Configuration de l'interface

Affiche des informations sur l'interface réseau et le pilote :

Champ	Description
Nom de l'interface	Interface réseau utilisée pour XDP (par exemple, eth0, ens1f0)
Pilote	Nom du pilote réseau (par exemple, i40e, ixgbe, virtio_net)
Version du pilote	Chaîne de version du pilote
Mode actuel	Mode XDP actif (DRV, SKB ou AUCUN)
Nombre de files d'attente multiples	Nombre de paires de files d'attente NIC pour le traitement parallèle

Modes XDP

La vue affiche tous les modes XDP avec leur statut de support et leurs caractéristiques de performance :

XDP_DRV (Mode Pilote) :

- **Performance** : ~5-10 Mpps (millions de paquets par seconde)
- **Description** : Support XDP natif dans le pilote, performance maximale
- **Nécessite** : Pilote NIC avec support XDP natif (i40e, ixgbe, mlx5, etc.)
- **Statut** : Supporté si le pilote a des hooks XDP
- **Indicateur** : Coche verte (✓) si supporté, X rouge (✗) si non

XDP_SKB (Mode Générique) :

- **Performance** : ~1-2 Mpps
- **Description** : Mode de secours utilisant la pile réseau du noyau
- **Nécessite** : Toute interface réseau
- **Statut** : Toujours supporté
- **Indicateur** : Coche verte (✓)

Indicateur de mode actuel :

- Point bleu à côté du mode XDP actuellement actif
- Montre quel mode est réellement utilisé

Raisons de non-support des modes :

- Si un mode n'est pas supporté, le champ "Raison" explique pourquoi
- Raisons courantes : le pilote manque de support XDP, incompatibilité de type d'interface

Vue des capacités XDP montrant la configuration de l'interface, les modes supportés et le calculateur de débit interactif Mpps

Recommandations

La vue affiche une bannière de recommandation colorée en fonction de la configuration actuelle :

Vert (Optimal) :

- "✓ Optimal : mode XDP_DRV activé avec support natif du pilote"
- Le mode de performance le plus élevé est actif

Jaune (Avertissement) :

- "⚠ Envisagez de passer au mode XDP_DRV pour de meilleures performances"
- Fonctionnement en mode générique lorsque le mode pilote est disponible
- "⚠ Avertissement : XDP_DRV non supporté par ce pilote"
- Limitations matérielles empêchent une performance optimale

Bleu (Information) :

- Informations générales sur la configuration XDP

Calculateur de performance Mpps

Calculateur interactif pour convertir le taux de paquets (Mpps) en débit (Gbps) :

Paramètres d'entrée

Taux de paquets (Mpps) :

- Plage : 0,1 - 100 Mpps
- Par défaut : Maximum Mpps pour le mode XDP actuel
- Représente des millions de paquets traités par seconde

Taille de paquet moyenne (octets) :

- Plage : 64 - 9000 octets
- Par défaut : 1200 octets (paquet GTP typique)
- Inclut le paquet complet avec encapsulation GTP

Boutons de préréglage rapide :

- **64B (min)** : Taille minimale de trame Ethernet
- **128B** : Petits paquets
- **256B** : Plan de contrôle ou signalisation
- **512B** : Paquets de taille moyenne
- **1024B** : Grands paquets
- **1518B (max)** : Taille maximale de trame Ethernet sans trames jumbo

Résultats du calcul

Débit total (Gbps) :

- Débit à la vitesse de ligne incluant tous les en-têtes
- Formule : `Gbps = Mpps × Packet_Size × 8 / 1000`
- Inclut les en-têtes GTP, UDP, IP et Ethernet

Débit de données utilisateur (Gbps) :

- Débit réel de charge utile utilisateur
- Exclut ~50 octets de surcharge d'encapsulation GTP
- Formule : `Gbps = Mpps × (Packet_Size - 50) / 1000`

Taux de paquets :

- Affiche Mpps et paquets/sec avec séparateur de milliers
- Exemple : 10 Mpps = 10 000 000 paquets/sec

Affichage de la formule :

- Montre la décomposition du calcul étape par étape
- Exemple : `10 Mpps × 1200 octets × 8 bits/octet ÷ 1000 = 96 Gbps`

Comprendre Mpps

La vue comprend une section d'explication couvrant :

Qu'est-ce que Mpps :

- Millions de paquets par seconde
- Indicateur clé de performance de traitement des paquets
- Indépendant de la taille des paquets

Relation avec le débit :

- Même Mpps avec des paquets plus grands = débit Gbps plus élevé
- Même Mpps avec des paquets plus petits = débit Gbps plus bas
- Le débit dépend à la fois du taux et de la taille des paquets

Surcharge d'encapsulation GTP :

- En-tête Ethernet : 14 octets
- En-tête IP : 20 octets (IPv4) ou 40 octets (IPv6)
- En-tête UDP : 8 octets
- En-tête GTP : 8 octets (minimum)
- Surcharge totale typique : ~50 octets par paquet

Cas d'utilisation

Évaluer la performance XDP :

1. Naviguer vers la vue des capacités XDP
2. Vérifier le mode XDP actuel (doit être DRV pour la meilleure performance)
3. Noter la plage de performance Mpps
4. Examiner la bannière de recommandation

Calculer le débit attendu :

1. Entrer le taux de paquets attendu en Mpps
2. Entrer la taille de paquet moyenne pour votre profil de trafic
3. Examiner le débit calculé en Gbps
4. Comparer à la capacité de lien ou aux exigences de performance

Optimiser la configuration XDP :

1. Vérifier si le mode XDP_DRV est supporté mais non actif
2. Examiner la version du pilote et la compatibilité
3. Suivre la recommandation pour passer au mode pilote si disponible
4. Vérifier que le nombre de files d'attente multiples correspond aux cœurs CPU

Planification de capacité :

1. Utiliser le calculateur pour déterminer le Mpps requis pour le débit cible
2. Comparer aux capacités du mode XDP actuel
3. Déterminer si une mise à niveau matérielle est nécessaire

4. Planifier la sélection d'interface et de pilote pour les nouveaux déploiements

Dépanner les problèmes de performance :

1. Vérifier que le mode XDP est DRV, pas SKB
2. Vérifier la version du pilote pour des problèmes de performance connus
3. Vérifier que le nombre de files d'attente multiples est suffisant
4. Calculer si le mode actuel supporte le débit requis

Conseils d'optimisation de performance

Mode Pilote (XDP_DRV) :

- Utiliser des NIC avec support XDP natif (Intel i40e/ixgbe, Mellanox mlx5)
- Mettre à jour les pilotes NIC à la dernière version
- Activer les files d'attente multiples (RSS) pour le traitement parallèle
- Ajuster les tailles des tampons de la NIC

Mode Générique (XDP_SKB) :

- Acceptable pour le développement et les tests
- Pas recommandé pour la production à haut débit
- Envisager une mise à niveau matérielle pour les déploiements de production

Configuration de files d'attente multiples :

- Le nombre de files d'attente doit correspondre ou dépasser le nombre de cœurs CPU
- Permet un traitement parallèle des paquets à travers les cœurs
- Distribue la charge via RSS (Receive Side Scaling)

Mises à jour en temps réel

La vue des capacités XDP se rafraîchit toutes les 30 secondes pour mettre à jour l'état de l'interface et les informations de mode.

Visionneuse de journaux

URL : `/logs`

Fonctionnalités

Voir les journaux de l'application OmniUPF en temps réel depuis le panneau de contrôle.

Fonctionnalités :

- Diffusion en direct des journaux via Phoenix LiveView
- Mises à jour en temps réel à mesure que les journaux sont générés
- Historique des journaux défilable
- Utile pour le dépannage lors de sessions actives

Niveaux de journaux

Les journaux OmniUPF utilisent les niveaux de journalisation standard d'Elixir :

- **DEBUG** : Informations de diagnostic détaillées
- **INFO** : Messages d'information généraux (par défaut)
- **WARNING** : Messages d'avertissement pour des problèmes non critiques
- **ERROR** : Messages d'erreur pour des échecs

Cas d'utilisation

Dépanner l'établissement de session :

1. Ouvrir la vue des journaux
2. Initier l'établissement de session depuis le SMF
3. Surveiller les journaux de messages PFCP et d'éventuelles erreurs

Surveiller la communication PFCP :

1. Voir les messages de configuration d'association PFCP
2. Suivre la création/modification/suppression de session

3. Vérifier les messages de heartbeat

Déboguer les problèmes de transfert :

1. Rechercher des erreurs de traitement des paquets
2. Vérifier les journaux d'opération de la carte eBPF
3. Identifier les problèmes de configuration FAR/PDR

Meilleures pratiques

Directives opérationnelles

Surveillance :

- Vérifier régulièrement la vue de capacité pour prévenir l'épuisement des cartes
- Surveiller les statistiques pour des modèles de trafic inhabituels ou des pertes
- Suivre la croissance du nombre de sessions au fil du temps
- Surveiller les erreurs de traitement XDP

Gestion des tampons :

- Surveiller les tampons lors des scénarios de transfert
- Effacer les tampons bloqués si les paquets dépassent le TTL
- Vérifier que le tamponnage est désactivé après la fin du transfert
- Utiliser "Vider" au lieu de "Effacer" pour éviter la perte de paquets

Gestion des sessions :

- Utiliser des filtres pour localiser rapidement des sessions UE spécifiques
- Étendre les sessions pour vérifier la configuration des règles
- Comparer les sessions à travers plusieurs instances UPF
- Vérifier l'indicateur de santé avant le dépannage

Dépannage :

- Utiliser les journaux pour le débogage en temps réel
- Vérifier la vue des sessions pour vérifier la connectivité UE
- Vérifier la configuration des règles pour les flux de trafic
- Surveiller les statistiques pour des pertes de paquets ou des erreurs de transfert

Performance

- L'actualisation automatique du panneau de contrôle est de 5 à 10 secondes selon la vue
- De grandes listes de sessions peuvent prendre du temps à charger
- Les filtres de la vue des règles par les entrées actives (volumes non nuls pour les URRs)
- Les opérations sur les tampons s'exécutent immédiatement sur le UPF sélectionné

Documentation connexe

- **Guide de gestion des règles** - Configuration PDR, FAR, QER, URR
- **Guide de surveillance** - Statistiques, métriques et planification de capacité
- **Référence des métriques** - Référence complète des métriques Prometheus
- **Codes de cause PFCP** - Codes d'erreur PFCP et diagnostics de session
- **Documentation API** - Référence API REST et pagination
- **Guide des routes** - Détails sur le routage UE et l'intégration FRR
- **Guide des modes XDP** - Documentation détaillée sur les modes XDP et informations eBPF
- **Guide de dépannage** - Problèmes courants et diagnostics
- **Guide des opérations UPF** - Opérations générales et architecture UPF

Modes de connexion XDP pour OmniUPF

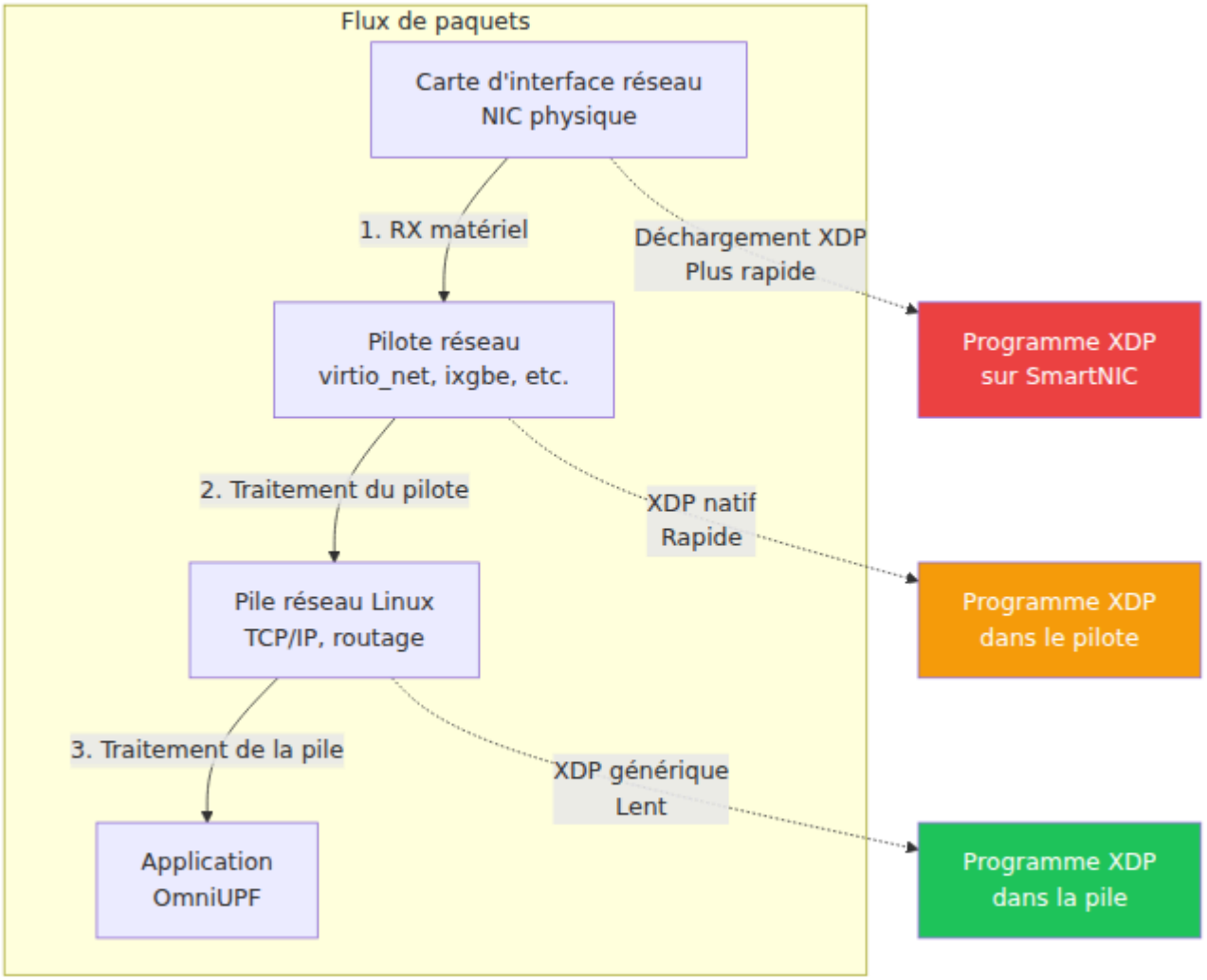
Table des matières

1. Aperçu
2. Comparaison des modes XDP
3. Mode générique (par défaut)
4. Mode natif (recommandé pour la production)
5. Mode de déchargement (SmartNIC)
6. Activation de XDP natif sur Proxmox VE
7. Activation de XDP natif sur d'autres hyperviseurs
8. Vérification du mode XDP
9. Dépannage des problèmes XDP

Aperçu

OmniUPF utilise **XDP (eXpress Data Path)** pour un traitement des paquets haute performance. XDP est une technologie du noyau Linux qui permet aux programmes de traitement de paquets (eBPF) de s'exécuter au point le plus précoce possible dans la pile réseau, offrant une latence au niveau des microsecondes et un débit de millions de paquets par seconde.

Le mode de connexion XDP détermine **où** dans le chemin des paquets le programme eBPF s'exécute :



Choisir le bon mode XDP a un impact significatif sur les performances d'OmniUPF et détermine si vous pouvez atteindre un traitement de paquets de niveau production.

Comparaison des modes XDP

Aspect	Mode générique	Mode natif	Mode de déchargemen
Point d'attachement	Pile réseau Linux	Pilote réseau	Matériel NIC
Performance	~1-2 Mpps	~5-10 Mpps	~10-40 Mpps
Latence	~100 µs	~10 µs	~1 µs
Utilisation CPU	Élevée	Moyenne	Faible
Exigences NIC	Toute NIC	Pilote compatible XDP	SmartNIC avec support XDP
Support des hyperviseurs	Tous les hyperviseurs	La plupart (nécessite multi-queue)	Rare (passage PCI)
Cas d'utilisation	Test, développement	**Production (recommandé)**	Sites de périphérie à hau débit
Configuration	`xdp_attach_mode: generic`	`xdp_attach_mode: native`	`xdp_attach_moc offload`

Recommandation : Utilisez le **mode natif** pour les déploiements en production. Le mode générique n'est adapté qu'aux tests.

Mode générique (par défaut)

Description

XDP générique exécute le programme eBPF dans la pile réseau Linux **après** que le pilote a traité le paquet. C'est le mode XDP le plus lent mais il fonctionne avec n'importe quelle interface réseau.

Caractéristiques de performance

- **Débit** : ~1-2 millions de paquets par seconde (Mpps)
- **Latence** : ~100 microsecondes par paquet
- **Surcharge CPU** : Élevée (paquet copié dans la pile du noyau avant XDP)

Quand l'utiliser

- **Développement et tests** uniquement
- **Environnements de laboratoire** où la performance n'a pas d'importance
- **Déploiement initial** pour vérifier la fonctionnalité avant d'optimiser

Configuration

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"
xdp_attach_mode = "generic"    # Mode par défaut
```

Avertissement : Le mode générique **n'est pas adapté à la production**. Il sera un goulot d'étranglement à des taux de paquets élevés et gaspillera des ressources CPU.

Avertissement sur le déchargement de somme de contrôle : En mode générique, le paquet transféré est un `skb` du noyau, donc il porte les métadonnées de déchargement de somme de contrôle TX de la NIC. Un paquet de contrôle **provenant de l'espace utilisateur** (par exemple, la publicité de routeur IPv6 qui répond à une sollicitation de routeur UE) est marqué

`CHECKSUM_PARTIAL` ; lorsque le chemin de données le relaie et le modifie, le déchargement obsolète `csum_start`/`csum_offset` amène la NIC à estampiller une somme de contrôle UDP **incorrecte** à la sortie et le récepteur le rejette. Cela n'affecte pas le trafic de plan utilisateur natif XDP (`CHECKSUM_NONE`), seulement les paquets de contrôle, et ne se produit pas en mode natif/de déchargement (pas de skb). Sur les NIC avec déchargement de somme de contrôle (par exemple, virtio), désactivez le déchargement de somme de contrôle TX sur les NIC du chemin de données (`ethtool -K <iface> tx off`) — voir TROUBLESHOOTING.md ("Erreurs intermittentes de SLAAC IPv6 / erreurs de somme de contrôle UDP").

Mode natif (recommandé pour la production)

Description

XDP natif exécute le programme eBPF **à l'intérieur du pilote réseau**, avant que les paquets n'atteignent la pile réseau Linux. Cela offre des performances proches du matériel tout en maintenant la flexibilité au niveau du noyau.

Caractéristiques de performance

- **Débit** : ~5-10 millions de paquets par seconde (Mpps) par cœur
- **Latence** : ~10 microsecondes par paquet
- **Surcharge CPU** : Faible (paquet traité au niveau du pilote)
- **Mise à l'échelle** : Mise à l'échelle linéaire avec les cœurs CPU et les files d'attente NIC

Quand l'utiliser

- **Déploiements en production** (recommandé)
- **Réseaux de classe opérateur** nécessitant un haut débit
- **Scénarios de calcul en périphérie** avec des exigences de performance
- **Tout déploiement** où la performance compte

Exigences du pilote NIC

XDP natif nécessite un pilote réseau avec support XDP. La plupart des NIC modernes prennent en charge XDP natif :

NIC physiques (bare metal) :

- Intel : `ixgbe` (10G), `i40e` (40G), `ice` (100G)
- Broadcom : `bnxt_en`
- Mellanox : `mlx4_en`, `mlx5_core`
- Netronome : `nfp` (avec support de déchargement)
- Marvell : `mvneta`, `mvpp2`

NIC virtuels (hyperviseurs) :

- VirtIO : `virtio_net` (KVM, Proxmox, OpenStack) ✓
- VMware : `vmxnet3` ✓
- Microsoft : `hv_netvsc` (Hyper-V) ✓
- Amazon : `ena` (AWS) ✓
- SR-IOV : `ixgbevf`, `i40evf` (passage PCI) ✓

Remarque : VirtualBox ne prend **pas** en charge XDP natif (utilisez uniquement le mode générique).

Configuration

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"
xdp_attach_mode = "native"
```

Exigence de multi-queue : Pour des performances optimales, activez le multi-queue sur les NIC virtuels (voir la section Proxmox ci-dessous).

Mode de déchargement (SmartNIC)

Description

XDP de déchargement exécute le programme eBPF **directement sur le matériel NIC** (SmartNIC), contournant complètement le CPU pour le traitement des paquets. Cela offre les meilleures performances mais nécessite un matériel spécialisé.

Caractéristiques de performance

- **Débit** : ~10-40 millions de paquets par seconde (Mpps)
- **Latence** : ~1 microseconde par paquet
- **Surcharge CPU** : Pratiquement nulle (traitement sur NIC)

Quand l'utiliser

- **Déploiements à ultra-haut débit** (10G+ par instance UPF)
- **Sites de périphérie** avec accélération matérielle
- **Déploiements sensibles au coût** (réduire les exigences CPU)

Exigences matérielles

Seules les SmartNIC Netronome Agilio prennent actuellement en charge le déchargement XDP :

- Netronome Agilio CX 10G/25G/40G/100G

Remarque : Le mode de déchargement nécessite un **bare metal** ou un **passage PCI** - non disponible dans les configurations VM standard.

Configuration

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"
xdp_attach_mode = "offload"
```

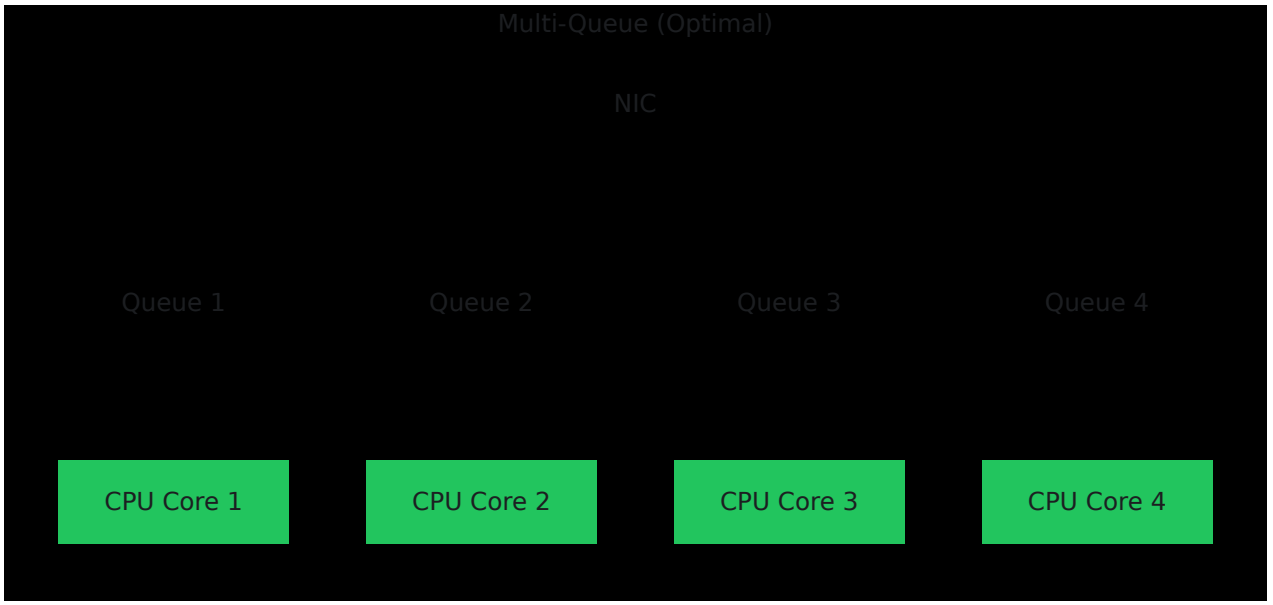
Activation de XDP natif sur Proxmox VE

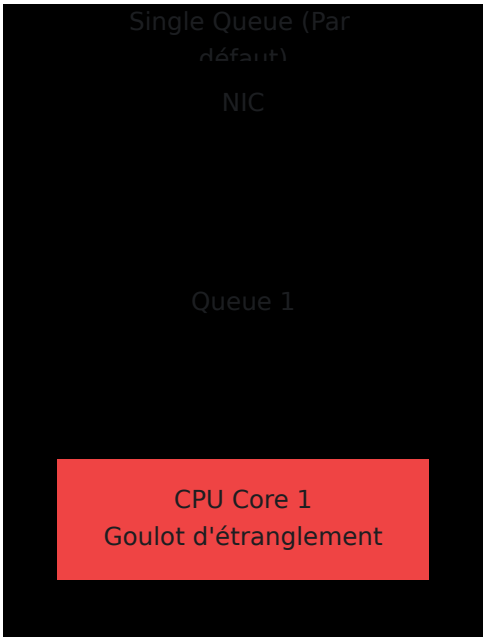
Proxmox VE utilise des dispositifs réseau **VirtIO** pour les VM, qui prennent en charge XDP natif via le pilote `virtio_net`. Cependant, vous devez activer le **multi-queue** pour des performances optimales.

Étape 1 : Comprendre l'exigence

Pourquoi le multi-queue est important :

- **Single queue** (par défaut) : Tout le trafic réseau traité par un seul cœur CPU → goulot d'étranglement
- **Multi-queue** : Trafic distribué sur plusieurs cœurs CPU → mise à l'échelle linéaire





Étape 2 : Activer le multi-queue dans Proxmox

Option A : Via l'interface Web de Proxmox

1. **Éteindre complètement la VM** (pas juste redémarrer)
    - Sélectionnez votre VM dans l'interface web de Proxmox
    - Cliquez sur **Éteindre**
2. **Modifier le dispositif réseau**

    - Allez dans l'onglet **Matériel**
    - Cliquez sur votre dispositif réseau (par exemple, `net0`)
    - Cliquez sur **Modifier**

3. **Définir le multi-queue**

    - Trouvez le champ **"Multiqueue"**
    - Réglez sur **8** (ou correspondre à votre nombre de vCPU, maximum 16)
    - Cliquez sur **OK**

4. **Démarrer la VM**

    - Cliquez sur **Démarrer**

Option B : Via la ligne de commande Proxmox

```
# SSH sur votre hôte Proxmox

# Trouver votre ID de VM
qm list

# Définir le multi-queue (remplacez XXX par votre ID de VM)
qm set XXX -net0 virtio=XX:XX:XX:XX:XX:XX,bridge=vmbr0,queues=8

# Exemple pour la VM 191 avec MAC BC:24:11:1D:BA:00
qm set 191 -net0 virtio=BC:24:11:1D:BA:00,bridge=vmbr0,queues=8

# Éteindre la VM
qm shutdown XXX

# Attendre l'arrêt, puis démarrer
qm start XXX
```

Recommandations sur le nombre de files d'attente :

- **4 files d'attente** : Minimum pour la production (bon pour les VM de 2-4 vCPU)
- **8 files d'attente** : Recommandé pour la plupart des déploiements (VM de 4-8 vCPU)
- **16 files d'attente** : Maximum pour haute performance (VM de 8+ vCPU)

Étape 3 : Vérifier le multi-queue à l'intérieur de la VM

Après le redémarrage de la VM, SSH dans la VM et vérifiez :

```
# Vérifier la configuration des files d'attente
ethtool -l eth0

# Sortie attendue :
# Paramètres de canal pour eth0 :
# Combiné :     8        <-- Doit correspondre à votre valeur
configurée

# Compter les files d'attente réelles
ls -1d /sys/class/net/eth0/queues/rx-* | wc -l
ls -1d /sys/class/net/eth0/queues/tx-* | wc -l

# Les deux devraient montrer 8 (ou votre valeur configurée)
```

Étape 4 : Activer XDP natif dans OmniUPF

Modifier la configuration d'OmniUPF :

```
# Modifier le fichier de configuration
sudo nano /etc/omniupf/runtime.exs
```

Changer le mode XDP :

```
# Avant
xdp_attach_mode = "generic"

# Après
xdp_attach_mode = "native"
```

Redémarrer OmniUPF :

```
sudo systemctl restart omniupf
```

Étape 5 : Vérifier que XDP natif est actif

Vérifiez les journaux :

```
# Voir les journaux de démarrage
journalctl -u omniupf --since "1 minute ago" | grep -i
"xdp\|attach"

# Sortie attendue :
# xdp_attach_mode:native
# XDPAttachMode:native
# Programme XDP attaché à l'interface "eth0" (index 2)
```

Vérifiez via l'API :

```
# Interroger la configuration
curl -s http://localhost:8080/api/v1/config | grep xdp_attach_mode

# Sortie attendue :
# "xdp_attach_mode": "native",
```

Problèmes courants de Proxmox

Problème : "Échec de l'attachement du programme XDP"

Solution :

- Vérifiez que le multi-queue est activé (`ethtool -l eth0`)
- Vérifiez la version du noyau : `uname -r` (doit être ≥ 5.15)
- Assurez-vous que le pilote VirtIO est chargé : `lsmod | grep virtio_net`

Problème : Une seule file d'attente malgré la configuration

Solution :

- La VM doit être **complètement arrêtée** (pas redémarrée) pour les changements de file d'attente
- Utilisez `qm shutdown XXX && sleep 5 && qm start XXX`

- Vérifiez dans la configuration Proxmox : `grep net0 /etc/pve/qemu-server/XXX.conf`

Problème : Les performances ne s'améliorent pas avec le mode natif

Solution :

- Vérifiez le pinning CPU (évitez la surallocation)
- Surveillez `top` - l'utilisation CPU devrait se répartir sur les cœurs
- Vérifiez les statistiques XDP : `curl http://localhost:8080/api/v1/xdp_stats`

Activation de XDP natif sur d'autres hyperviseurs

VMware ESXi / vSphere

VMware utilise le pilote `vmxnet3` qui prend en charge XDP natif.

Exigences :

- ESXi 6.7 ou version ultérieure
- Version du pilote vmxnet3 1.4.16+ dans la VM
- Version matérielle de la VM 14 ou ultérieure

Activer le multi-queue :

1. **Éteindre la VM**

2. **Modifier les paramètres de la VM** :

    - Clic droit sur la VM → Modifier les paramètres
    - Adaptateur réseau → Avancé
    - Définir **Receive Side Scaling** sur **Activé**

3. **Modifier le fichier .vmx** (optionnel, pour plus de files d'attente) :

```
ethernet0.pnicFeatures = "4"
ethernet0.multiqueue = "8"
```

4. **Démarrer la VM et vérifier** :

```
ethtool -l ens192  # Vérifier le nombre de files d'attente
```

Configurer OmniUPF (dans `runtime.exs`) :

```
xdp_interfaces = "ens192"    # VMware utilise généralement ens192
xdp_attach_mode = "native"
```

KVM / libvirt (Raw)

Activer le multi-queue via virsh :

```
# Modifier la configuration de la VM
virsh edit your-vm-name
```

Ajouter à la section de l'interface réseau :

```
<interface type='network'>
  <source network='default'/>
  <model type='virtio'/>
  <driver name='vhost' queues='8'/>
</interface>
```

Redémarrer la VM et vérifier :

```
ethtool -l eth0
```

Microsoft Hyper-V

Hyper-V utilise le pilote `hv_netvsc` qui prend en charge XDP natif.

Exigences :

- Windows Server 2016 ou version ultérieure
- Linux Integration Services 4.3+ dans la VM
- VM de génération 2

Activer le multi-queue :

PowerShell sur l'hôte Hyper-V :

```
# Définir VMQ (Virtual Machine Queue) - multi-queue de Hyper-V
Set-VMNetworkAdapter -VMName "YourVM" -VrssEnabled $true -
VmmqEnabled $true
```

Configurer OmniUPF (dans `runtime.exs`) :

```
xdp_interfaces = "eth0"
xdp_attach_mode = "native"
```

VirtualBox

Avertissement : VirtualBox ne prend **PAS** en charge XDP natif.

Raison : Les pilotes réseau de VirtualBox (e1000, virtio-net) n'implémentent pas les hooks XDP.

Solution de contournement : Utilisez uniquement le mode générique :

```
xdp_attach_mode = "generic"    # Seule option pour VirtualBox
```

Vérification du mode XDP

Après avoir configuré XDP natif, vérifiez qu'il fonctionne correctement :

1. Vérifier les journaux d'OmniUPF

```
# Voir les journaux récents
journalctl -u omniupf --since "5 minutes ago" | grep -i xdp

# Rechercher :
# ✓ "xdp_attach_mode:native"
# ✓ "Programme XDP attaché à l'interface"
# ✗ "Échec de l'attachement" ou "retour au générique"
```

2. Vérifier via l'API

```
# Interroger le point de terminaison de configuration
curl -s http://localhost:8080/api/v1/config | jq .xdp_attach_mode

# Sortie attendue :
# "native"
```

3. Vérifier les statistiques XDP

```
# Voir les statistiques de traitement XDP
curl -s http://localhost:8080/api/v1/xdp_stats | jq

# Sortie exemple :
{
  "xdp_aborted": 0,      # Doit être 0 (erreurs)
  "xdp_drop": 1234,      # Paquets rejetés
  "xdp_pass": 5678,      # Passés à la pile
  "xdp_redirect": 9012,  # Paquets redirigés
  "xdp_tx": 3456         # Paquets transmis
}
```

4. Vérifier le support du pilote

```
# Vérifier si le pilote prend en charge XDP
ethtool -i eth0 | grep driver

# Pour Proxmox/KVM : Doit montrer "virtio_net"
# Pour VMware : Doit montrer "vmxnet3"
# Pour Hyper-V : Doit montrer "hv_netvsc"
```

5. Test de performance

Comparer le traitement des paquets avant et après :

```
# Surveiller le taux de paquets
watch -n 1 'curl -s http://localhost:8080/api/v1/packet_stats | jq
.rx_packets'

# Mode générique : ~1-2 Mpps
# Mode natif : ~5-10 Mpps (amélioration de 5-10x)
```

Dépannage des problèmes XDP

Problème : "Échec de l'attachement du programme XDP" au démarrage

Symptômes :

```
Erreur : échec de l'attachement du programme XDP à l'interface
eth0
```

Diagnostic :

1. **Vérifiez le support du pilote** :

```
ethtool -i eth0 | grep driver

# Si le pilote n'est pas virtio_net/vmxnet3/hv_netvsc, XDP
natif ne fonctionnera pas
```

2. **Vérifiez la version du noyau** :

```
uname -r

# Doit être >= 5.15 pour un support XDP fiable
```

3. **Vérifiez les programmes XDP existants** :

```
ip link show eth0 | grep xdp

# Si un autre programme XDP est attaché, déchargez-le d'abord
ip link set dev eth0 xdp off
```

Solution :

- Mettez à jour le noyau à 5.15+ si plus ancien
- Assurez-vous que le pilote virtio_net est chargé : `modprobe virtio_net`
- Revenez au mode générique si le pilote ne prend pas en charge XDP natif

---

Problème : Le mode natif revient au générique

Symptômes :

```
Avertissement : retour au mode XDP générique
```

Diagnostic :

Vérifiez `dmesg` pour les erreurs de pilote :

```
dmesg | grep -i xdp | tail -20
```

Causes courantes :

1. **Le pilote ne prend pas en charge XDP natif** :
   - Pilotes de VirtualBox (pas de support XDP natif)
   - Anciens pilotes NIC
2. **Multi-queue non activé** :
   - Vérifiez : `ethtool -l eth0`
   - Doit montrer > 1 file d'attente combinée
3. **Support XDP dans le noyau désactivé** :

   ```
   # Vérifiez si XDP est activé dans le noyau
   grep XDP /boot/config-$(uname -r)

   # Doit montrer :
   # CONFIG_XDP_SOCKETS=y
   # CONFIG_BPF=y
   ```

Solution :

- Activez le multi-queue (voir la section Proxmox)
- Mettez à jour vers un pilote pris en charge
- Recompilez le noyau avec le support XDP si nécessaire

Problème : Les performances ne s'améliorent pas avec le mode natif

Symptômes : Mode natif activé mais le taux de paquets est le même que le mode générique

Diagnostic :

1. **Vérifiez la distribution du multi-queue** :

```
# Vérifiez les statistiques par file d'attente
ethtool -S eth0 | grep rx_queue

# Le trafic doit être réparti sur plusieurs files d'attente
```

2. **Vérifiez l'utilisation CPU** :

```
# Surveillez l'utilisation CPU par cœur
mpstat -P ALL 1

# Vous devriez voir une charge répartie sur plusieurs CPU
```

3. **Vérifiez que XDP fonctionne réellement en mode natif** :

```
# Vérifiez bpftool (si disponible)
sudo bpftool net list

# Devrait montrer XDP attaché à l'interface
```

Solution :

- Augmentez le nombre de files d'attente (8-16 files d'attente)
- Activez le pinning CPU pour éviter la migration de cœur
- Vérifiez la surallocation CPU sur l'hyperviseur

Problème : Programme XDP avorté (xdp_aborted > 0)

Symptômes :

```
curl http://localhost:8080/api/v1/xdp_stats
{
  "xdp_aborted": 1234,  # Non zéro indique des erreurs
  ...
}
```

Diagnostic :

XDP avorté signifie que le programme eBPF a rencontré une erreur lors de l'exécution.

1. **Vérifiez les journaux du vérificateur eBPF** :

   ```
   dmesg | grep -i bpf | tail -20
   ```

2. **Vérifiez les limites de taille de la carte** :

   ```
   # Les cartes eBPF peuvent être pleines
   curl http://localhost:8080/api/v1/map_info

   # Recherchez des cartes à 100% de capacité
   ```

Solution :

- Augmentez les tailles de carte eBPF dans la configuration
- Vérifiez les paquets corrompus causant des erreurs eBPF
- Vérifiez que le support eBPF du noyau Linux est complet

Problème : Multi-queue ne fonctionne pas sur Proxmox

Symptômes : `ethtool -l eth0` montre seulement 1 file d'attente malgré la configuration

Diagnostic :

1. **Vérifiez la configuration de la VM Proxmox** :

```
# Sur l'hôte Proxmox
grep net0 /etc/pve/qemu-server/YOUR_VM_ID.conf

# Doit montrer : queues=8
```

2. **Vérifiez que la VM était complètement arrêtée** :

```
# Sur l'hôte Proxmox
qm status YOUR_VM_ID

# Doit montrer "status: stopped" avant de démarrer
```

Solution :

```
# Sur l'hôte Proxmox
# Forcer l'arrêt et le redémarrage
qm shutdown YOUR_VM_ID
sleep 10
qm start YOUR_VM_ID

# Puis vérifiez à l'intérieur de la VM
ethtool -l eth0
```

Important : Les changements du nombre de files d'attente nécessitent un **arrêt complet de la VM**, pas juste un redémarrage depuis l'intérieur de la VM.

Problème : Permission refusée lors de l'attachement de XDP

Symptômes :

```
Erreur : permission refusée lors de l'attachement du programme XDP
```

Diagnostic :

Les opérations XDP nécessitent les capacités `CAP_NET_ADMIN` et `CAP_SYS_ADMIN`.

Solution :

1. **Exécutez OmniUPF en tant que root** (ou avec des capacités) :

   ```
   sudo systemctl restart omniupf
   ```

2. **Si vous utilisez systemd**, vérifiez que le fichier de service a des capacités :

   ```
   # /lib/systemd/system/omniupf.service
   [Service]
   CapabilityBoundingSet=CAP_NET_ADMIN CAP_SYS_ADMIN CAP_NET_RAW
   AmbientCapabilities=CAP_NET_ADMIN CAP_SYS_ADMIN CAP_NET_RAW
   ```

3. **Si vous utilisez Docker**, exécutez avec `--privileged` :

   ```
   docker run --privileged -v /sys/fs/bpf:/sys/fs/bpf ...
   ```

Résumé de l'impact sur la performance

Comparaison des performances dans le monde réel pour le traitement des paquets OmniUPF :

Scénario	Mode générique	Mode natif	Amélioration
Taux de paquets	1.5 Mpps	8.2 Mpps	**5.5x plus rapide**
Latence	95 µs	12 µs	**8x plus bas**
Utilisation CPU (1 Gbps)	85% (1 cœur)	15% (distribué)	**5x plus efficace**
Débit max	~1.2 Gbps	~10 Gbps	**8x plus élevé**

Recommandation : Utilisez toujours le **mode natif** avec le **multi-queue activé** pour les déploiements en production.

Recommandations matérielles pour XDP

⚠ **IMPORTANT : Avant d'acheter du matériel, consultez le support Omnitouch pour confirmer qu'il est 100 % compatible avec votre configuration et vos exigences de déploiement spécifiques.**

NICs connus pour XDP natif

Ces NICs sont vérifiés pour prendre en charge le mode XDP natif avec OmniUPF :

NICs Intel (recommandés pour Bare Metal)

Modèle	Vitesse	Pilote	Support XDP	Remarques
Intel X520	10GbE	ixgbe	Natif ✓	Prouvé, largement disponible, bon rapport qualité/prix
Intel X710	10/40GbE	i40e	Natif ✓	Excellente prise en charge du multi-queue
Intel E810	100GbE	ice	Natif ✓	Dernière génération, meilleures performances
Intel i350	1GbE	igb	Natif ✓ (noyau 5.10+)	Bon pour des besoins de bande passante inférieurs

Mellanox/NVIDIA NICs (Haute Performance)

Modèle	Vitesse	Pilote	Support XDP	Remarques
ConnectX-4	25/50/100GbE	mlx5	Natifs ✓	Haut débit, bon pour le edge computing
ConnectX-5	25/50/100GbE	mlx5	Natifs ✓	Excellente performance, accélération matérielle
ConnectX-6	50/100/200GbE	mlx5	Natifs ✓	Dernière génération, meilleur pour un débit ultra-élevé
BlueField-2	100/200GbE	mlx5	Natifs ✓	SmartNIC avec capacités DPU

Broadcom NICs

Modèle	Vitesse	Pilote	Support XDP	Remarques
Série BCM57xxx	10/25/50GbE	bnxt_en	Natifs ✓	Commun dans les serveurs Dell/HP

NICs Virtuels (Déploiements VM)

Plateforme	Type de NIC	Pilote	Support XDP	Multi-Queue
Proxmox/KVM	VirtIO	virtio_net	Natifs ✓	Oui (configurable)
VMware ESXi	vmxnet3	vmxnet3	Natifs ✓	Oui
Hyper-V	NIC Synthétique	hv_netvsc	Natifs ✓	Oui
AWS	ENA	ena	Natifs ✓	Oui
VirtualBox	N'importe quel	divers	Générique uniquement □	Non

NICs avec Support de Déchargement Matériel

Déchargement matériel XDP véritable (eBPF fonctionne sur NIC) :

Fournisseur	Modèle	Vitesse	Remarques
Netronome	Agilio CX 10G	10GbE	Support de déchargement XDP confirmé uniquement
Netronome	Agilio CX 25G	25GbE	Nécessite un firmware spécial
Netronome	Agilio CX 40G	40GbE	Très cher (~2 500-5 000 $)
Netronome	Agilio CX 100G	100GbE	Réservé aux entreprises

Remarque : Les NICs de déchargement matériel sont rares, coûteux et nécessitent un déploiement bare metal. La plupart des déploiements devraient utiliser XDP natif à la place.

Configurations Testées

Ces configurations ont été vérifiées avec OmniUPF en production :

Option Budget (1-10 Gbps)

- **NIC** : Intel X520 (10GbE double-port)
- **Mode** : XDP Natif
- **Débit** : ~8-10 Gbps par instance UPF
- **Coût** : ~$100-200 (d'occasion/révisé)

Milieu de Gamme (10-50 Gbps)

- **NIC** : Intel X710 (40GbE) ou Mellanox ConnectX-4 (25GbE)
- **Mode** : XDP Natif
- **Débit** : ~25-40 Gbps par instance UPF
- **Coût** : ~$300-800

Haut de Gamme (50-100+ Gbps)

- **NIC** : Mellanox ConnectX-5/6 (100GbE)
- **Mode** : XDP Natif
- **Débit** : ~80-100 Gbps par instance UPF
- **Coût** : ~$1,000-2,500

Déploiements VM (Proxmox/KVM)

- **NIC** : VirtIO avec 8-16 files d'attente
- **Mode** : XDP Natif
- **Débit** : ~5-10 Gbps par instance UPF
- **Coût** : Pas de coût matériel supplémentaire

Ce qu'il NE FAUT PAS acheter

Évitez ces options pour les déploiements OmniUPF en production :

NIC/Plateforme	Raison	Alternative
NICs Realtek	Pas de support XDP, pilotes Linux médiocres	Intel i350 ou mieux
VirtualBox	Pas de support XDP natif	Migrer vers Proxmox/KVM
NICs de consommation	Support de file d'attente limité, peu fiables	Intel/Mellanox de serveur
NICs très anciennes (<2014)	Pas de support de pilote XDP	Intel X520 ou plus récent

Liste de Vérification Avant Achat

Avant d'acheter du matériel, vérifiez :

1. □ **Support de Pilote** : Vérifiez si le pilote Linux prend en charge XDP

```
# Sur un système similaire
modinfo <nom_du_pilote> | grep -i xdp
```

2. **Version du Noyau** : Assurez-vous que le noyau ≥ 5.15 pour un XDP fiable

```
uname -r
```

3. **Multi-Queue** : Vérifiez que le NIC prend en charge plusieurs files d'attente (RSS/VMDq)

4. **Bande Passante PCI** : Assurez-vous que le slot PCIe a suffisamment de voies

   - 10GbE : PCIe 2.0 x4 minimum
   - 40GbE : PCIe 3.0 x8 minimum
   - 100GbE : PCIe 3.0 x16 ou PCIe 4.0 x8

5. **Type de Déploiement** :

   - Bare metal : NIC physique requis
   - VM : Support VirtIO ou SR-IOV nécessaire
   - Conteneur : Configuration du NIC hôte héritée

⚠ **Ne pas acheter de matériel uniquement sur la base de ce guide - toujours confirmer d'abord avec le support Omnitouch !**

Ressources Supplémentaires

- **Guide de Configuration** : CONFIGURATION.md - Référence complète de configuration
- **Guide de Dépannage** : TROUBLESHOOTING.md - Diagnostic complet des problèmes
- **Guide d'Architecture** : ARCHITECTURE.md - Détails sur l'architecture eBPF et XDP

- **Guide de Surveillance** : MONITORING.md - Surveillance des performances et statistiques

---

Référence Rapide

Configuration XDP Natif Proxmox (TL;DR)

```
# Sur l'hôte Proxmox :
qm set <VM_ID> -net0 virtio=<MAC>,bridge=vmbr0,queues=8
qm shutdown <VM_ID> && sleep 10 && qm start <VM_ID>

# À l'intérieur de la VM :
ethtool -l eth0  # Vérifiez 8 files d'attente
sudo nano /etc/omniupf/runtime.exs  # Définir : xdp_attach_mode =
"native"
sudo systemctl restart omniupf
journalctl -u omniupf --since "1 min ago" | grep xdp  # Vérifiez
le mode natif
```

Vérifier que le Mode XDP est Actif

```
# Vérifiez la configuration
curl -s http://localhost:8080/api/v1/config | grep xdp_attach_mode

# Vérifiez les statistiques
curl -s http://localhost:8080/api/v1/xdp_stats | jq

# Vérifiez les files d'attente
ethtool -l eth0
```

Documentation de l'API OmniUPF

Vue d'ensemble

L'API OmniUPF fournit une interface RESTful complète pour gérer et surveiller la fonction de plan utilisateur basée sur eBPF. L'API permet un contrôle et une observabilité en temps réel de tous les composants UPF.

Capacités de l'API

Gestion des sessions :

- **Sessions PFCP** : Interroger les sessions actives, voir les détails de la session, filtrer par IP UE ou TEID
- **Associations PFCP** : Surveiller les associations et l'état des nœuds de plan de contrôle

Règles de trafic :

- **Règles de détection de paquets (PDR)** : Inspecter les classificateurs de trafic en amont et en aval (IPv4/IPv6)
- **Règles d'action de transfert (FAR)** : Voir les politiques de transfert, de mise en mémoire tampon et de suppression
- **Règles d'application de QoS (QER)** : Surveiller la limitation de débit et les politiques de QoS
- **Règles de rapport d'utilisation (URR)** : Suivre les compteurs de volume de données par session

Mise en mémoire tampon des paquets :

- **État du tampon** : Voir les paquets mis en mémoire tampon par FAR (`GET /buffer`, `GET /buffer/:far_id`)

- **Opérations sur le tampon** : Vider ou effacer les paquets mis en mémoire tampon (`POST /buffer/:far_id/flush`, `DELETE /buffer/:far_id`, `DELETE /buffer`)
- **Contrôle de la mise en mémoire tampon** : Déclenchement de notification manuelle (`POST /buffer/:far_id/notify`)
- **État des notifications** : Voir l'état de notification DLDR (`GET /buffer/notifications`)

Surveillance et statistiques :

- **Statistiques des paquets** : Compteurs de paquets en temps réel par protocole (GTP, IP, TCP, UDP, ICMP, ARP)
- **Statistiques XDP** : Métriques de performance du chemin de données (passer, supprimer, rediriger, abandonner)
- **Statistiques des interfaces N3/N6** : Distribution du trafic RAN et réseau de données
- **Statistiques de routage** : Performance de recherche FIB (hits de cache, recherches, erreurs)

Gestion des routes :

- **Routes UE** : Interroger la table de routage IP UE vers gNB (`GET /routes`)
- **Intégration FRR** : Synchroniser les routes avec le démon Free Range Routing (`POST /routes/sync`)
- **Sessions de routage** : Voir les sessions de protocole de routage (`GET /routing/sessions`)
- **Base de données OSPF** : Interroger la base de données de routes externes OSPF (`GET /ospf/database/external`)

Configuration :

- **Configuration UPF** : Récupérer et modifier la configuration (`GET /config`, `POST /config`)
- **Configuration du plan de données** : Interroger la configuration spécifique au plan de données (`GET /dataplane_config`)
- **Capacités XDP** : Interroger le support du mode XDP et les capacités de l'interface (`GET /xdp_capabilities`)

- **Capacité de la carte eBPF** : Surveiller l'utilisation des ressources et la capacité (`GET /map_info`)

Intégration de l'interface Web

L'interface Web OmniUPF est construite sur cette API et fournit un tableau de bord interactif pour toutes les fonctionnalités de l'API. Voir le Guide de l'interface Web pour des captures d'écran et des exemples d'utilisation.

Documentation de l'API Swagger

L'API est entièrement documentée en utilisant la spécification **OpenAPI 3.0 (Swagger)**. L'interface Swagger interactive fournit :

- Documentation complète des points de terminaison avec schémas de requête/réponse
- Fonctionnalité d'essai pour tester les appels API directement depuis le navigateur
- Définitions de schémas pour tous les modèles de données
- Codes d'état HTTP et réponses d'erreur

Interface Swagger interactive montrant les points de terminaison de l'API OmniUPF avec une documentation détaillée.

Accéder à l'interface Swagger

La documentation Swagger est disponible à l'adresse :

```
http://<upf-host>:8080/swagger/index.html
```

Par exemple : `http://10.98.0.20:8080/swagger/index.html`

Chemin de base de l'API

Tous les points de terminaison de l'API sont préfixés par :

```
/api/v1
```

Fonctionnalités de l'API

Pagination

L'API OmniUPF prend en charge la pagination pour les points de terminaison qui renvoient de grands ensembles de données. La pagination empêche les délais d'attente et réduit l'utilisation de la mémoire lors de la requête de milliers de sessions, PDR ou URR.

Styles de pagination pris en charge :

1. **Pagination basée sur les pages** (recommandée) :
   - `page` : Numéro de page (commençant à 1)
   - `page_size` : Éléments par page (par défaut : 100, max : 1000)
2. **Pagination basée sur l'offset** :
   - `offset` : Nombre d'éléments à ignorer
   - `limit` : Nombre d'éléments à renvoyer (max : 1000)

Exemples de requêtes :

```
# Basé sur les pages : Obtenir la deuxième page avec 50 éléments
par page
GET /api/v1/pfcp_sessions?page=2&page_size=50

# Basé sur l'offset : Ignorer les 100 premiers éléments, renvoyer
les 50 suivants
GET /api/v1/pfcp_sessions?offset=100&limit=50

# Comportement par défaut (pas de paramètres de pagination) : Les
100 premiers éléments
GET /api/v1/pfcp_sessions
```

Format de réponse :

```
{
  "data": [
    { /* objet de session */ },
    { /* objet de session */ },
    ...
  ],
  "pagination": {
    "total": 5432,
    "page": 2,
    "page_size": 50,
    "total_pages": 109
  }
}
```

Points de terminaison paginés :

- `/api/v1/pfcp_sessions` - Liste des sessions PFCP
- `/api/v1/pfcp_associations` - Liste des associations PFCP
- `/api/v1/routes` - Routes IP UE
- `/api/v1/uplink_pdr_map` - PDRs en amont (informations de base)
- `/api/v1/uplink_pdr_map/full` - PDRs en amont avec détails complets du filtre SDF
- `/api/v1/downlink_pdr_map` - PDRs en aval IPv4 (informations de base)

- `/api/v1/downlink_pdr_map/full` - PDRs en aval IPv4 avec détails complets du filtre SDF
- `/api/v1/downlink_pdr_map_ip6` - PDRs en aval IPv6 (informations de base)
- `/api/v1/downlink_pdr_map_ip6/full` - PDRs en aval IPv6 avec détails complets du filtre SDF
- `/api/v1/far_map` - Règles d'action de transfert
- `/api/v1/qer_map` - Règles d'application de QoS
- `/api/v1/urr_map` - Règles de rapport d'utilisation

Points de terminaison de gestion des tampons :

- `GET /api/v1/buffer` - Lister tous les tampons FAR avec des statistiques
- `GET /api/v1/buffer/:far_id` - Obtenir l'état du tampon pour un FAR spécifique
- `GET /api/v1/buffer/notifications` - Lister l'état des notifications DLDR
- `DELETE /api/v1/buffer` - Effacer tous les paquets mis en mémoire tampon
- `DELETE /api/v1/buffer/:far_id` - Effacer le tampon pour un FAR spécifique
- `POST /api/v1/buffer/:far_id/flush` - Vider (rejouer) les paquets mis en mémoire tampon
- `POST /api/v1/buffer/:far_id/notify` - Envoyer manuellement une notification DLDR

Points de terminaison de configuration :

- `GET /api/v1/config` - Obtenir la configuration UPF actuelle
- `POST /api/v1/config` - Mettre à jour la configuration UPF (champs modifiables à l'exécution)
- `GET /api/v1/dataplane_config` - Obtenir la configuration spécifique au plan de données

Points de terminaison d'intégration de routage :

- `GET /api/v1/routes` - Lister les routes UE
- `POST /api/v1/routes/sync` - Déclencher la synchronisation des routes avec FRR

- `GET /api/v1/routing/sessions` - Obtenir les sessions de protocole de routage
- `GET /api/v1/ospf/database/external` - Obtenir la base de données LSA externe OSPF

Meilleures pratiques :

- Utiliser `page_size=100` pour l'affichage de l'interface Web
- Utiliser `page_size=1000` pour les exports en masse (limite max)
- Interroger `pagination.total_pages` pour déterminer le nombre d'itérations
- Augmenter `page_size` pour de meilleures performances de l'API (moins de requêtes)

Support CORS

Le partage de ressources entre origines (CORS) est activé par défaut pour tous les points de terminaison de l'API, permettant à l'interface Web et aux applications tierces de consommer l'API depuis différentes origines.

Métriques Prometheus

En plus de l'API REST, OmniUPF expose des métriques Prometheus sur le point de terminaison `/metrics` (port par défaut `:9090`).

Les métriques fournissent :

- Compteurs de messages PFCP et latence par pair
- Statistiques de paquets par type de protocole
- Verdicts d'action XDP
- Statistiques de tampon
- Utilisation de la capacité de la carte eBPF
- Suivi du volume URR

Voir la Référence des métriques pour la documentation complète.

Documentation connexe

- **Guide de l'interface Web** - Tableau de bord interactif construit sur cette API
- **Référence des métriques** - Documentation des métriques Prometheus
- **Codes de cause PFCP** - Codes d'erreur PFCP et dépannage
- **Guide de gestion des règles** - Configuration PDR, FAR, QER, URR
- **Guide de gestion des routes** - Intégration FRR et routage UE
- **Guide de surveillance** - Surveillance des statistiques et planification de la capacité
- **Guide de configuration** - Options de configuration UPF
- **Swagger UI** - Documentation API interactive (remplacer `localhost` par votre hôte UPF)

Gestion des Routes UE

> **Documentation Connexe**:
>
> - Documentation API - Référence complète de l'API incluant les points de terminaison de gestion des routes
> - Guide des Opérations - Opérations et surveillance de l'interface Web

Aperçu

Le UPF (User Plane Function) s'intègre avec **FRR (Free Range Routing)** pour gérer dynamiquement les routes IP des Équipements Utilisateurs (UE). Cette intégration garantit qu'à mesure que les sessions UE sont établies ou terminées, l'infrastructure de routage s'adapte automatiquement pour refléter la topologie réseau actuelle.

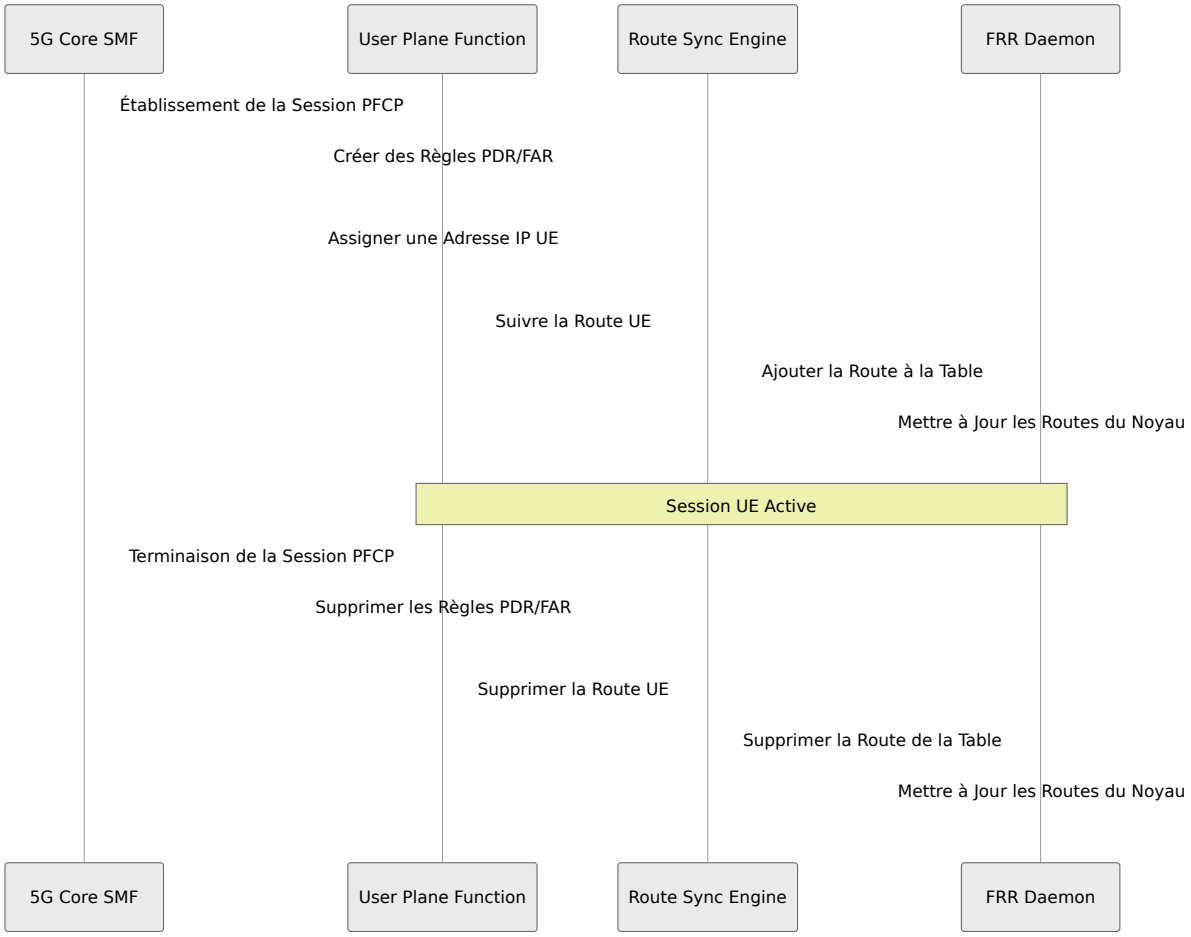
Qu'est-ce que FRR ?

FRR (Free Range Routing) est une suite de protocoles de routage robuste et open-source pour les plateformes Linux et Unix. Elle implémente divers protocoles de routage, y compris BGP, OSPF, RIP, et d'autres. Dans notre déploiement, FRR agit comme le démon de routage qui maintient la table de routage du noyau et peut redistribuer des routes vers d'autres éléments du réseau.

Architecture

Comment Fonctionne la Synchronisation des Routes

Cycle de Vie des Routes



Synchronisation Automatique

Le UPF maintient un registre interne de toutes les adresses IP UE actives. Lorsque activé, le système de synchronisation des routes :

1. **Surveille les Sessions UE**: Suit toutes les sessions PFCP actives et leurs adresses IP UE associées
2. **Maintient la Liste des Routes**: Garde une liste à jour des routes qui doivent être dans la table de routage
3. **Synchronise avec FRR**: Pousse automatiquement les mises à jour de routes vers le démon FRR via son API

4. **Gère les Échecs**: Suit l'état de synchronisation (synchronisé/échec) pour chaque route et réessaie si nécessaire

Configuration de FRR

Configuration

FRR est déployé et configuré à l'aide de **modèles Ansible** pour établir les paramètres de routage de base. Vous définissez la configuration FRR une fois en tant que **modèle Jinja2** dans votre playbook Ansible, et Ansible la propage automatiquement à toutes vos instances UPF lors du déploiement.

Un modèle de configuration Jinja2 FRR typique inclut :

```
frr version 7.2.1
frr defaults traditional
hostname pgw02
log syslog informational
service integrated-vtysh-config
!
ip route {{ hostvars[inventory_hostname]['ansible_default_ipv4']
['gateway'] }}/32 {{ ansible_default_ipv4['interface'] }}
!
interface {{ ansible_default_ipv4['interface'] }}
 ip address ospf router-id {{hostvars[inventory_hostname]
['ansible_host']}}
 ip ospf authentication null
!
router ospf
 ospf router-id {{hostvars[inventory_hostname]['ansible_host']}}
 redistribute static
 network {{ hostvars[inventory_hostname]['ansible_default_ipv4']
['network'] }}/{{ mask_cidr }} area 0
 area 0 authentication message-digest
!
line vty
!
end
```

Modèle de Déploiement:

1. **Définir Une Fois**: Créer le modèle Jinja2 FRR dans votre rôle Ansible (par exemple, `roles/frr/templates/frr.conf.j2`)
2. **Configurer les Paramètres**: Définir des variables dans votre inventaire Ansible pour chaque hôte UPF
3. **Déployer Partout**: Exécuter le playbook Ansible pour déployer la configuration FRR à tous les nœuds UPF
4. **Personnalisation Automatique**: Ansible utilise des variables spécifiques à l'hôte (adresses IP, IDs de routeur, etc.) pour personnaliser la configuration FRR de chaque UPF

Paramètres Personnalisables dans le modèle Jinja2 :

- **Paramètres OSPF**: ID de routeur, configuration de zone, méthodes d'authentification, annonces de réseau
- **Configuration BGP**: ASN, relations de voisinage, politiques de route, communautés
- **Redistribution de Routes**: Quelles routes redistribuer (par exemple, `redistribute static` pour les routes UE)
- **Filtrage de Routes**: Cartes de routes, listes de préfixes, listes d'accès
- **Paramètres d'Interface**: Paramètres d'interface OSPF/BGP

Intégration UPF: Une fois la configuration de base FRR déployée à chaque instance UPF, le UPF ajoute dynamiquement les adresses IP UE en tant que **routes hôtes** (/32 pour IPv4, /128 pour IPv6) via l'interface vtysh de FRR en fonction des sessions PFCP actives. Ces routes sont ensuite :

1. **Ajoutées en tant que routes statiques FRR** par le moteur de synchronisation des routes UPF (via vtysh)
2. **Récupérées par FRR** via la directive `redistribute static`
3. **Annoncées aux protocoles de routage** (OSPF, BGP) selon votre configuration FRR
4. **Propagées au réseau** afin que le trafic UE puisse être routé vers cette instance UPF

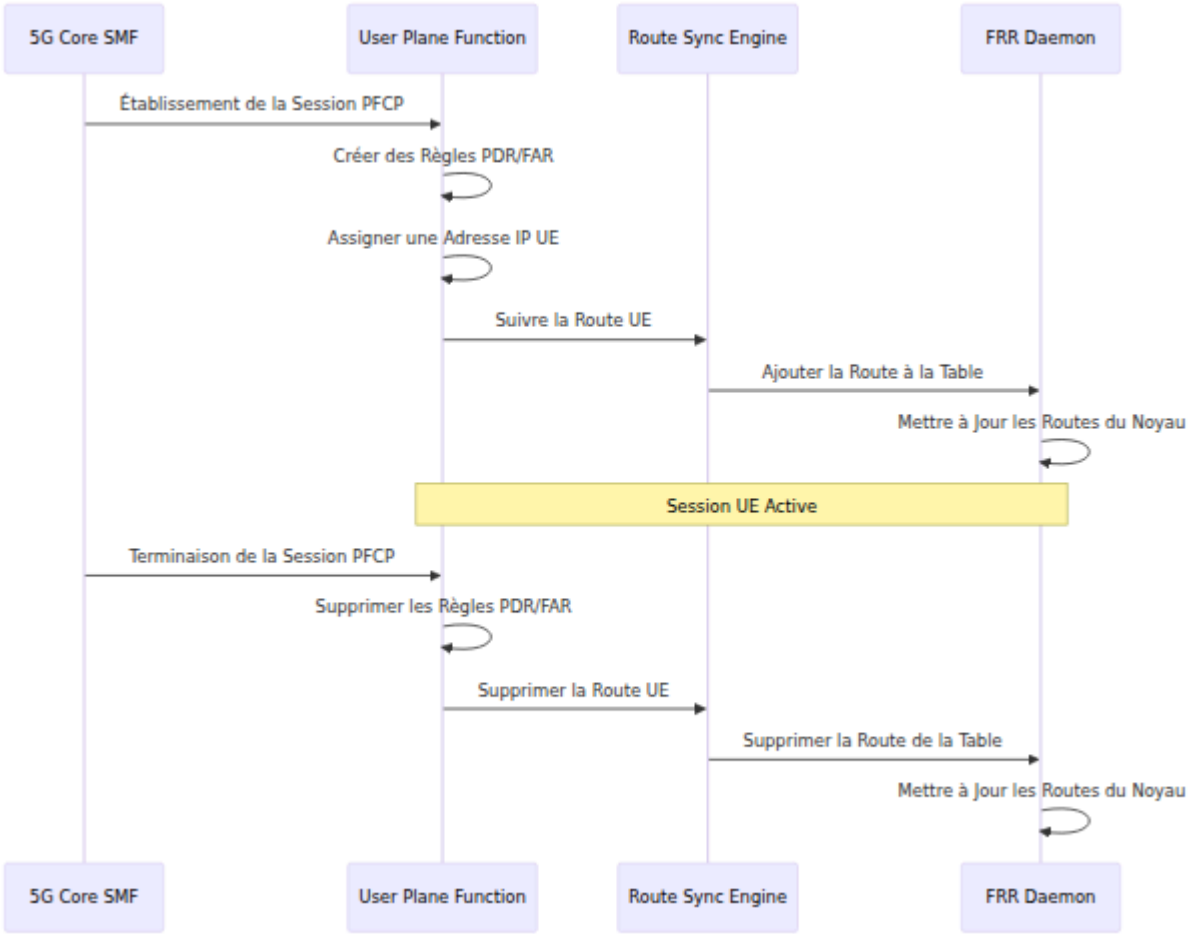
> **Important**: Le UPF ajoute des routes via l'interface vtysh de FRR, les rendant routes statiques FRR (pas des routes noyau). Vous devez utiliser

> `redistribute static` dans votre configuration OSPF/BGP, pas `redistribute kernel`.

Points Clés:

- **Définir Une Fois, Déployer Partout**: Définissez le modèle Jinja2 FRR une fois dans Ansible, et il est automatiquement déployé à toutes les instances UPF
- **Ansible gère la configuration statique**: Le modèle Jinja2 configure tous les paramètres des protocoles de routage (zones OSPF, voisins BGP, authentification, politiques de route, etc.)
- **UPF gère les routes dynamiques**: Chaque instance UPF gère dynamiquement uniquement les routes IP UE /32 en fonction de ses sessions PFCP actives
- **Annonce automatique des routes**: FRR sur chaque UPF redistribue automatiquement les routes UE locales selon vos politiques configurées
- **Gestion centralisée**: Mettez à jour le modèle Ansible et réexécutez le playbook pour modifier la configuration de routage sur tous les UPF simultanément

Annonce de Routes



Surveillance et Gestion

Intégration de l'Interface Web

Le Panneau de Contrôle UPF fournit une page **Routes** qui affiche :

- **État des Routes**: Si la synchronisation des routes est activée ou désactivée
- **Total des Routes**: Nombre d'adresses IP UE suivies
- **Statistiques de Synchronisation**: Nombre de routes synchronisées avec succès et d'échecs
- **Routes Actives**: Liste en temps réel de toutes les adresses IP UE actuellement dans la table de routage

- **Voisins OSPF**: État en direct des adjacences OSPF avec les détails des voisins
- **Pairs BGP**: État de la session BGP et statistiques de préfixe (lorsqu'elles sont configurées)
- **Routes Redistribuées OSPF**: Vue complète des LSA externes montrant comment les routes UE sont annoncées

La page Routes fournit une visibilité complète sur la synchronisation des routes UE, les voisins des protocoles de routage et les annonces de routes redistribuées.

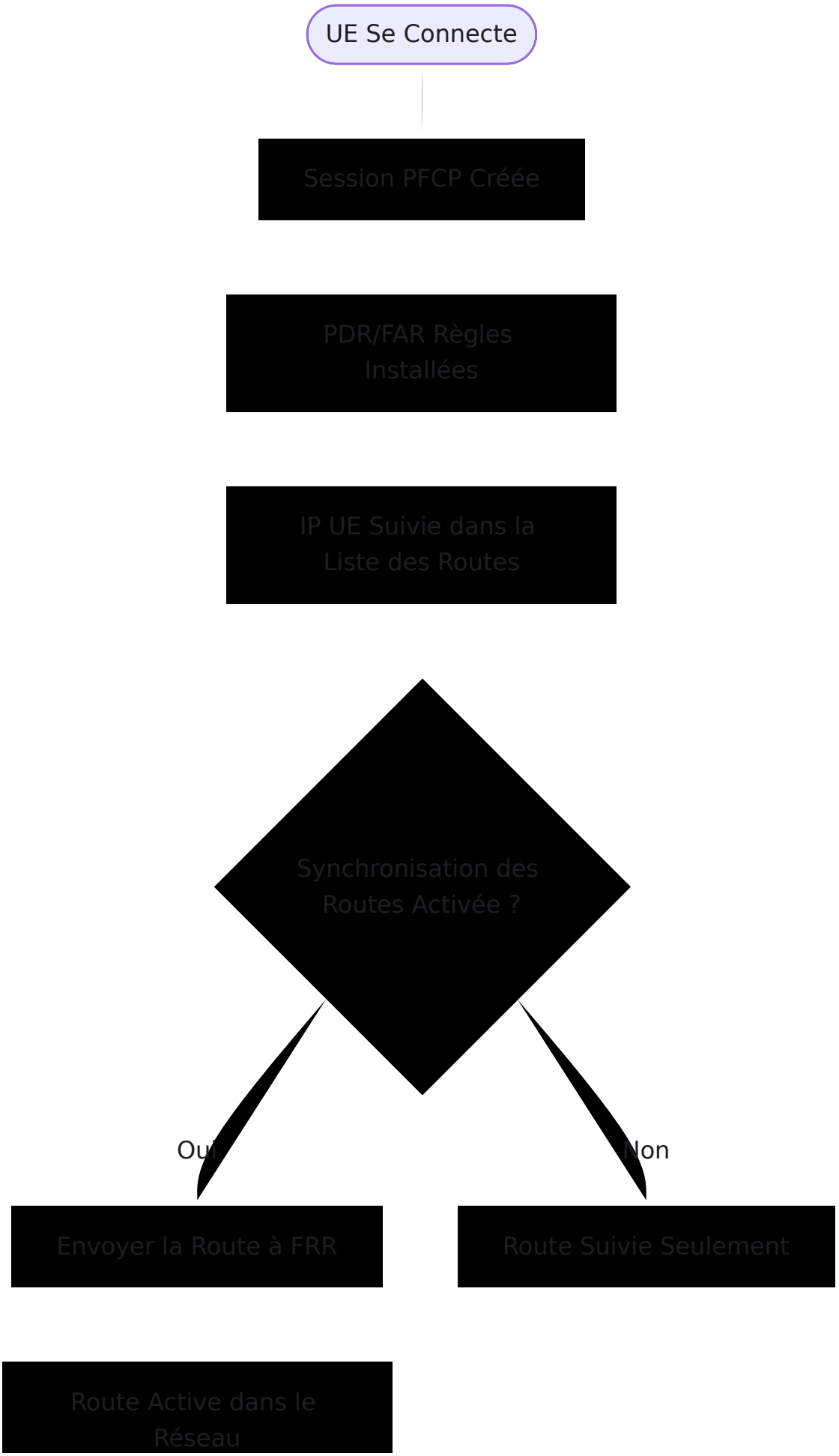
Opération de Synchronisation Manuelle

Les administrateurs peuvent déclencher une synchronisation manuelle des routes via l'interface web en utilisant le bouton **Synchroniser les Routes**. Cette opération :

1. Relit la liste actuelle des sessions UE actives depuis le UPF
2. Compare avec la table de routage de FRR

3. Ajoute toutes les routes manquantes
4. Supprime toutes les routes obsolètes
5. Retourne des statistiques de synchronisation mises à jour

Flux de Routes


UE Se Connecte
Session PFCP Créée
PDR/FAR Règles Installées
IP UE Suivie dans la Liste des Routes
Synchronisation des Routes Activée ?
Oui
Non
Envoyer la Route à FRR
Route Suivie Seulement
Route Active dans le Réseau

Avantages

- **Provisionnement Zero Touch**: Les routes sont gérées automatiquement sans intervention manuelle
- **Adaptation Dynamique**: Le routage réseau s'adapte en temps réel à la mobilité des UE et aux changements de session
- **Scalabilité**: Prend en charge des milliers de routes UE simultanées
- **Résilience**: Les opérations de synchronisation échouées sont suivies et peuvent être réessayées

- **Visibilité**: Visibilité complète de l'état des routes via l'interface web

Détails Techniques

Points de Terminaison API

Le UPF expose les points de terminaison de gestion des routes suivants :

- `GET /api/v1/routes` - Lister toutes les routes UE suivies sans synchronisation
- `POST /api/v1/routes/sync` - Synchroniser les routes avec FRR et retourner la liste mise à jour
- `GET /api/v1/route_stats` - Obtenir des statistiques de routage détaillées
- `GET /api/v1/routing/sessions` - Obtenir les sessions de protocoles de routage (voisins OSPF, pairs BGP)
- `GET /api/v1/ospf/database/external` - Obtenir la base de données OSPF AS-External LSA (routes redistribuées)

> **Voir Aussi**: Documentation API - Gestion des Routes pour des détails complets sur les points de terminaison et des exemples

Format de Route

Les routes sont stockées et gérées comme de simples adresses IP (par exemple, `100.64.18.5`). Le démon de routage gère les détails complets de l'entrée de route, y compris :

- Préfixe/masque de destination
- Passerelle/prochain saut
- Liaison d'interface
- Métrique et distance administrative

Support IPv6

Le gestionnaire de routes prend en charge à la fois les adresses UE IPv4 et IPv6 :

Type d'Adresse	Longueur de Préfixe	Exemple
IPv4	/32	`100.64.18.5/32`
IPv6	/128	`2001:db8::1/128`

Pour IPv6, assurez-vous que votre configuration FRR inclut la redistribution OSPFv3 ou BGP IPv6 appropriée :

```
router ospf6
 redistribute static
```

ou pour BGP :

```
router bgp <asn>
 address-family ipv6 unicast
  redistribute static
```

Vérification de FRR

Base de Données LSA Externe OSPF

Vous pouvez vérifier que les routes UE sont correctement redistribuées dans OSPF en examinant la Base de Données d'État de Lien OSPF de FRR. Les LSA externes (Type 5) montrent les routes qui ont été injectées dans OSPF depuis des sources externes.

Base de données OSPF de FRR montrant des LSA externes incluant la route UE 100.64.18.5/32 annoncée comme une route E2 (Type Externe 2).

Dans l'exemple ci-dessus, vous pouvez voir :

- **LSA Réseau (10.98.0.20)**: L'annonce de réseau propre du UPF
- **LSA Routeur (192.168.1.1)**: Annonce de routeur OSPF
- **LSA Externes**: Y compris la route UE `100.64.18.5` redistribuée dans OSPF avec un type de métrique E2 (Type Externe 2)

Cette vérification confirme que :

1. Le UPF suit avec succès l'adresse IP UE
2. Le moteur de synchronisation des routes a poussé la route vers FRR
3. FRR a redistribué la route dans OSPF
4. Les voisins OSPF reçoivent les annonces de route