Guide des opérations OmniUPF

Table des matières

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, une application de la QoS et une gestion du trafic pour les réseaux mobiles. Construit 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 de 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** basés sur des filtres et des règles de paquets
- **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 la fonctionnalité complète d'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.

Principales capacités 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 de paquets sans copie via eBPF
- Transfert de paquets au niveau du noyau (sans surcharge d'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 de plan utilisateur unifiée fournissant un transfert de paquets de qualité opérateur pour les réseaux 5G autonomes (SA), 5G NSA et 4G

LTE/EPC. **OmniUPF est un produit unique** qui peut fonctionner simultanément comme :

- **UPF (Fonction de plan utilisateur)** - plan utilisateur 5G/NSA (contrôlé par OmniSMF via N4/PFCP)
- **PGW-U (PDN Gateway User Plane)** - passerelle EPC 4G vers des réseaux externes (contrôlé par OmniPGW-C via Sxc/PFCP)
- **SGW-U (Serving Gateway User Plane)** - 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 pur 5G
- **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 situe au niveau du 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 SGW-U et PGW-U séparés (Roaming/Multi-site)

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

Mode de boucle N9 (Instance unique SGWU+PGWU)

Pour des déploiements simplifiés, OmniUPF peut fonctionner **à la fois comme**

SGWU et PGWU sur une seule instance avec un traitement de boucle N9 entièrement en eBPF.

Caractéristiques clés :

- 🔀 **Latence N9 sub-microseconde** - Traitée entièrement en eBPF, ne touche jamais le réseau
- 🔀 **Réduction de 40-50 % du CPU** - Une seule passe XDP contre deux instances séparées
- 🔀 **Déploiement simplifié** - Une instance, un fichier de configuration
- 🔀 **Détection automatique** - Lorsque `n3_address` = `n9_address`, la boucle est activée
- 🔀 **Conformité totale à la 3GPP** - Protocoles PFCP et GTP-U standard

Configuration :

```
# OmniUPF config.yml
interface_name: [eth0]
n3_address: "10.0.1.10"      # IP de l'interface S1-U
n9_address: "10.0.1.10"      # La même IP active la boucle N9
pfcp_address: ":8805"         # Les deux SGWU-C et PGWU-C se
connectent ici
```

Quand l'utiliser :

- Déploiements de calcul en périphérie (minimiser la latence)
- Environnements à budget limité (serveur unique)
- Laboratoire/test (configuration simplifiée)
- Déploiements petits à moyens (< 100K abonnés)

Quand NE PAS l'utiliser :

- Redondance géographique requise (SGWU et PGWU dans des emplacements différents)
- Mandats réglementaires pour des passerelles séparées
- Échelle massive (> 1M abonnés)

Pour des détails complets, des exemples de configuration, des dépannages et des métriques de performance, voir **Guide des opérations de boucle N9**.

Comment les fonctions de plan utilisateur fonctionnent 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 en amont** (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 aux PDR en amont basés sur 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 retiré, les paquets transférés à l'interface N6 (5G) ou SGi (4G)
   - URR suit les comptes de paquets et d'octets pour la facturation

3. **Traitement des paquets en aval** (Réseau de données → UE)

   - **5G** : Les paquets arrivent sur l'interface N6 sous forme d'IP natif
   - **4G** : Les paquets arrivent sur l'interface SGi sous forme d'IP natif
   - Le plan utilisateur fait correspondre les paquets aux PDR en aval 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
   - L'encapsulation GTP-U est ajoutée avec le TEID approprié
   - **5G** : Les paquets sont transférés à l'interface N3 vers gNB
   - **4G** : Les paquets sont transférés à S1-U (SGW-U) ou S5/S8 (PGW-U) vers eNodeB

4. **Mobilité et transfert**

   - **5G** : OmniSMF met à jour les règles PDR/FAR lors de scénarios de transfert
   - **4G** : OmniSGW-C/OmniPGW-C met à jour les règles lors de transfert inter-eNodeB ou TAU (Mise à jour de la 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 3GPP standard :

Interfaces 5G

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

Interfaces 4G/EPC

Interface	De → À	Objectif	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 du 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 natif)	TS 23.401

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

Pour la gestion des sessions PFCP, voir Opérations PFCP.

Composants UPF

Chemin de données eBPF

Le **chemin de données eBPF** est le moteur de traitement de paquets central qui s'exécute 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 aux 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 selon 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 dans la mémoire du noyau) pour le stockage des règles :

Nom de la carte	Objectif	Clé	Valeur
`uplink_pdr_map`	PDR en amont	TEID (32 bits)	Infos PDR (ID FAR, ID QER, IDs URR)
`downlink_pdr_map`	PDR en aval (IPv4)	Adresse IP de l'UE	Infos PDR
`downlink_pdr_map_ip6`	PDR en aval (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 (amont, aval, total)
`sdf_filter_map`	Filtres SDF	ID PDR	Filtres d'application (ports, protocoles)

Caractéristiques de performance :

- **Zéro copie** : Paquets traités entièrement dans l'espace noyau
- **Support XDP** : Attacher au niveau du pilote réseau pour une latence sub-microseconde
- **Multi-cœurs** : Se répartit sur 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 vie et configuration de l'association PFCP
- **Cycle de vie de session** : Créer, modifier et supprimer des sessions PFCP
- **Installation des règles** : Traduire les IE PFCP en entrées de carte eBPF
- **Rapport d'événements** : Notifier SMF des seuils d'utilisation, des erreurs ou des événements de session

Support des messages PFCP :

Type de message	Direction	Objectif
Établissement d'association	SMF → UPF	Établir l'association de contrôle PFCP
Libération d'association	SMF → UPF	Détruire l'association PFCP
Cœur de vie	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	Rapport d'utilisation, erreurs ou 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)

Pour des opérations PFCP détaillées, voir **Guide des opérations PFCP**.

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 route, 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 l'état et de la santé
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 des cartes eBPF
Plan de données	`/dataplane_config`	Adresses d'interface N3/N9

Pour des détails sur l'API et son utilisation, voir **Guide des opérations PFCP** et **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 UE, TEID et comptes de règles
- **Gestion des règles** : Voir et gérer les PDR, FAR, QER et URR 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 des statistiques** : Statistiques en temps réel sur les paquets, les routes, XDP et les interfaces N3/N6
- **Surveillance de la capacité** : Utilisation des cartes eBPF avec des indicateurs de capacité codés par couleur
- **Vue de configuration** : Afficher la configuration de l'UPF et les adresses de plan de données
- **Visionneuse de journaux** : Diffusion en direct des journaux pour le dépannage

Pour des opérations détaillées de l'interface utilisateur, 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 de vie 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.

Création de session PFCP

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

Flux d'établissement de session :

Contenu typique de la session :

- **PDR en amont** : Correspondre sur N3 TEID, 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)

Pour la surveillance des sessions, voir Opérations PFCP.

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 un nouveau point 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 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, voir Guide de gestion des règles.

Suppression de session PFCP

Lorsqu'une session PDU est libérée, le SMF supprime la session PFCP à 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 vidés
- 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 des sessions PDU actives de l'UE (5G) ou des contextes PDP (LTE). Chaque session contient :

- SEID local et distant (Identificateurs de point de session)
- PDR pour la classification des paquets
- FAR pour les décisions de transfert
- QER pour l'application de la QoS (optionnel)
- URR pour le suivi de l'utilisation (optionnel)

Opérations clés sur les sessions :

- **Voir toutes les sessions** avec adresses IP 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, voir 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 PDR en amont** indexés par TEID de l'interface N3
- **Voir les PDR en aval** indexés par adresse IP de l'UE (IPv4 et IPv6)
- **Inspecter les filtres SDF** pour une classification spécifique aux applications
- **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, METTRE EN MÉMOIRE TAMPON, 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 retard de paquet)
- **Surveiller les QER actifs** par session
- **Inspecter les marquages QFI** pour les flux QoS 5G

Règles de rapport d'utilisation (URR) :

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

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

Pour les opérations de règles, voir Guide de gestion des règles.

Mise en mémoire tampon des paquets

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

Le problème : Perte de paquets pendant la mobilité

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

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 lacunes** ou se coupent entièrement (appels téléphoniques interrompus)
- **Des téléchargements qui échouent** et doivent être redémarrés

Avec mise en mémoire tampon : OmniUPF maintient temporairement les paquets jusqu'à ce que le nouveau chemin soit établi, puis les transfère sans interruption. L'utilisateur connaît une **absence d'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 un UE se déplace entre des tours cellulaires :

Chronologie :

- **T+0ms** : Ancien chemin toujours actif
- **T+10ms** : SMF dit à UPF de mettre en mémoire tampon (ancien chemin se fermant, 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** : Aucune perte de paquets, transfert transparent.

---

2. Modification de session (changement de QoS, mise à jour de chemin)

Lorsque le réseau doit changer les paramètres de session :

- **Mise à niveau/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 d'entreprise (changement d'acheminement 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 en aval (récupération en mode veille)

Lorsque un UE est en mode veille (écran éteint, économie de batterie) et que des données en aval 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. Handover inter-RAT (4G ↔ 5G)

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

- Changements d'architecture (eNodeB ↔ gNB)
- Changements de points de tunnel (nouvelle allocation de TEID)
- 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 à deux niveaux** :

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

Processus de mise en mémoire tampon :

Détails clés :

- **Port de tampon** : Port UDP 22152 (paquets envoyés de eBPF à l'espace utilisateur)
- **Encapsulation** : Paquets enveloppés en GTP-U avec ID FAR comme TEID
- **Stockage** : Tampons en mémoire par FAR avec des métadonnées (horodatage, direction, taille de paquet)
- **Limites** :
  - Limite par FAR : 10 000 paquets (par défaut)
  - Limite globale : 100 000 paquets à travers tous les FAR
  - 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 en espace utilisateur** : 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** : L'espace utilisateur rejoue les paquets mis en mémoire tampon en utilisant les nouvelles règles FAR (nouveau point de tunnel)
6. **Reprendre normalement** : 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 1-2 secondes, peut se couper	Transparent, aucune interruption
Téléchargement de fichier à la limite de la cellule	Le téléchargement échoue, doit redémarrer	Le téléchargement continue sans interruption
Jeu en ligne en se déplaçant	La connexion se coupe, expulsé du jeu	Jeu fluide, aucune déconnexion
Appel VoIP dans la voiture	L'appel se coupe à chaque transfert	Clair comme du cristal, aucune coupure
Streaming vidéo dans un train	La vidéo se met en mémoire tampon, qualité diminue	Lecture fluide
Point d'accès mobile pour un ordinateur portable	La session SSH se coupe, l'appel vidéo échoue	Toutes les connexions maintenues

Avantages pour l'opérateur de réseau :

- **Taux de coupure d'appel réduit (CDR)** : KPI critique pour la qualité du réseau
- **Satisfaction client accrue** : Les utilisateurs ne remarquent pas les transferts
- **Coûts de support réduits** : Moins de plaintes concernant les connexions coupées
- **Avantage concurrentiel** : Marketing "Meilleur réseau pour la couverture"

Opérations de gestion des tampons

Les opérateurs peuvent surveiller et contrôler la mise en mémoire tampon via l'interface Web et l'API :

Surveillance :

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

Opérations de contrôle :

- **Vider les tampons** : Déclencher manuellement la lecture du tampon (dépannage)
- **Effacer les tampons** : Discarter les paquets mis en mémoire tampon

(nettoyer les tampons bloqués)
- **Ajuster le TTL** : Changer le temps d'expiration des paquets
- **Modifier les limites** : Augmenter la capacité de tampon par FAR ou globale

Dépannage :

- **Tampon ne se vidant pas** : Vérifiez si le SMF a envoyé une mise à jour FAR pour désactiver la mise en mémoire tampon
- **Débordement du tampon** : Augmenter les limites ou enquêter sur les raisons pour lesquelles la durée de mise en mémoire tampon est excessive
- **Paquets anciens dans le tampon** : Le TTL peut être trop élevé, ou la mise à jour FAR retardée
- **Mise en mémoire tampon excessive** : Peut indiquer des problèmes de mobilité ou des problèmes avec le SMF

Pour des opérations détaillées sur les tampons, voir Guide de gestion des tampons.

Configuration du tampon

Configurez le comportement de mise en mémoire tampon dans `config.yml` :

```yaml
# Paramètres de tampon
buffer_port: 22152                # Port UDP pour les paquets mis en mémoire tampon (par défaut)
buffer_max_packets: 10000         # Max paquets par FAR (prévenir l'épuisement de la mémoire)
buffer_max_total: 100000          # Max paquets totaux à travers tous les FAR
buffer_packet_ttl: 30             # TTL en secondes (discarter les paquets anciens)
buffer_cleanup_interval: 60       # Intervalle de nettoyage en secondes
```

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 des options de configuration complètes, voir Guide de configuration.

Statistiques et surveillance

Statistiques de paquets :

Métriques de traitement de paquets en temps réel incluant :

- **Paquets RX** : Total reçu de toutes les interfaces
- **Paquets TX** : Total transmis à toutes les interfaces
- **Paquets supprimés** : Paquets rejetés en raison d'erreurs ou de politiques
- **Paquets GTP-U** : Comptes de 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 eXpress Data Path :

- **XDP traités** : Paquets gérés au niveau XDP
- **XDP passés** : Paquets envoyés à la pile réseau
- **XDP supprimés** : Paquets supprimés au niveau XDP
- **XDP abandonnés** : Erreurs de traitement

Statistiques des interfaces N3/N6 :

Compteurs de trafic par interface :

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

Pour des détails de surveillance, voir Guide de surveillance.

Gestion de la capacité

Surveillance de la capacité des cartes eBPF :

La performance d'UPF dépend de la capacité des cartes eBPF. Les opérateurs peuvent :

- **Surveiller l'utilisation des cartes** avec des indicateurs de pourcentage

en temps réel
- **Voir les limites de capacité** pour chaque carte eBPF
- **Alertes codées par couleur** :
    - Vert (<50%) : Fonctionnement normal
    - Jaune (50-70%) : Prudence
    - Ambre (70-90%) : Avertissement
    - Rouge (>90%) : Critique

Cartes critiques à surveiller :

- `uplink_pdr_map` : Classification du trafic en amont
- `downlink_pdr_map` : Classification du trafic en aval IPv4
- `far_map` : Règles de transfert
- `qer_map` : Règles de QoS
- `urr_map` : Suivi de l'utilisation

Planification de la capacité :

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

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

Gestion de la configuration

Configuration de l'UPF :

Voir et vérifier les paramètres opérationnels de l'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 d'interface N3 en temps réel
- **Adresse N9 active** : Liaison d'interface N9 en temps réel (si activée)

Pour la visualisation de 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 profondes courantes :

1. **Association PFCP non établie**
    - Vérifiez que le SMF peut atteindre l'interface PFCP de l'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 l'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 de l'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 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 des pertes de paquets ou aucun flux de trafic

Causes profondes courantes :

1. **Mauvaise configuration PDR**
    - Vérifiez que le PDR en amont TEID correspond au TEID attribué par gNB
    - Vérifiez que le PDR en aval IP de l'UE correspond à l'IP attribuée
    - Inspectez les filtres SDF pour des règles trop restrictives
2. **Problèmes d'action FAR**
    - Vérifiez que l'action FAR est TRANSFÉRER (pas SUPPRIMER ou METTRE EN MÉMOIRE TAMPON)
    - Vérifiez les paramètres de création d'en-tête externe pour GTP-U
    - Assurez-vous que le point de destination est correct
3. **Limites de QoS dépassées**
    - Vérifiez les paramètres MBR (Débit maximal) du QER
    - Vérifiez l'allocation GBR (Débit garanti)
    - Surveillez les pertes de paquets dues à la limitation de débit
4. **Problèmes d'MTU d'interface**
    - Vérifiez que la surcharge GTP-U (40-50 octets) ne provoque pas de fragmentation
    - Vérifiez la configuration de l'MTU des interfaces N3/N6
    - Surveillez les messages ICMP indiquant la fragmentation nécessaire

---

Problèmes liés aux tampons

Symptômes : Paquets mis en mémoire tampon indéfiniment, débordement de tampon

Causes profondes courantes :

1. **Mise en mémoire tampon non désactivée après le transfert**
    - Vérifiez le drapeau de mise en mémoire tampon FAR (bit 2)
    - Vérifiez que le SMF a envoyé la modification de session pour désactiver la mise en mémoire tampon
    - Désactivez manuellement la mise en mémoire 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 du TTL du tampon (par défaut peut être trop

long)
    - Effacez les tampons expirés manuellement

3. **Capacité du tampon épuisée**

    - Surveillez l'utilisation totale du tampon et les limites par FAR
    - Vérifiez les règles mal configurées provoquant une mise en mémoire tampon excessive
    - Ajustez les limites max_per_far et max_total

Pour le dépannage des tampons, voir Opérations sur les tampons.

Anomalies de statistiques

Symptômes : Comptes de paquets inattendus, statistiques manquantes

Causes profondes courantes :

1. **Dépassement de compteur**

    - Les cartes eBPF utilisent des compteurs 64 bits (ne devraient pas déborder)
    - Vérifiez les événements de réinitialisation des compteurs 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. **Incohérence des statistiques d'interface**

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

Dégradation des performances

Symptômes : Latence élevée, faible débit, saturation du CPU

Diagnostic :

1. **Surveillez les statistiques XDP** : Vérifiez les suppressions ou abandons XDP
2. **Vérifiez le temps d'accès aux cartes eBPF** : Les recherches de hachage devraient être sub-microsecondes
3. **Passez en revue l'utilisation du CPU** : eBPF doit 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** : Millions de PDRs limités uniquement par la mémoire du noyau
- **Nombre de sessions** : Des milliers de sessions concurrentes par instance UPF
- **Débit** : Débit multi-gigabits avec un bon déchargement de NIC

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

Documentation supplémentaire

Guides d'opérations spécifiques aux composants

Pour des opérations détaillées et un dépannage pour chaque composant de l'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 de fixation 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 la capacité

Guide des modes XDP

Configuration et optimisation détaillées de XDP incluant :

- Modes de fixation XDP expliqués (générique/natif/déchargement)
- Comparaison de performance et benchmarks
- Guide étape par étape pour la configuration XDP native sur Proxmox VE
- Configuration multi-file pour des performances optimales

- 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 de la technologie eBPF et cycle de vie du programme
- Pipeline de traitement de paquets XDP avec appels de queue
- Mise en œuvre du protocole PFCP
- Architecture de mise en mémoire tampon (encapsulation GTP-U vers port 22152)
- Limitation de débit de fenêtre glissante QoS (fenêtre de 5 ms)
- Caractéristiques de performance (latence de 3,5 µs, 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 indicateurs d'action
- Règles d'application de la 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 en amont et en aval
- Logique de traitement des règles et priorité

Guide de surveillance

Statistiques et gestion de la capacité incluant :

- Statistiques d'interface N3/N6 et distribution du trafic
- Statistiques de traitement XDP (passer/supprimer/rediriger/abandonner)
- Surveillance de la capacité des cartes eBPF avec zones codées par couleur
- Métriques de performance (taux de paquets, débit, taux de perte)
- Formules de planification de capacité et estimation de session
- 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 des tampons et état de mise en mémoire tampon
- Tableau de bord des statistiques en temps réel
- Visualisation de la capacité des cartes eBPF
- Affichage de la configuration

Documentation API

Référence complète de l'API REST incluant :

- Documentation interactive OpenAPI/Swagger
- 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 la QoS (QER)
- Règles de rapport d'utilisation (URR)
- Gestion des tampons
- Statistiques et points de terminaison 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

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

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 des performances

- 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 SGWU+PGWU sur une instance unique (Boucle N9)

1. Guide des opérations de boucle N9 - Guide complet pour le déploiement combiné SGWU+PGWU
2. Boucle N9 - Configuration - Configuration réseau et PFCP
3. Boucle N9 - Surveillance - Vérifiez que la boucle est active
4. Boucle N9 - Dépannage - Problèmes courants et solutions

Déploiement sur Proxmox

1. Guide des modes XDP - Configuration XDP native sur Proxmox - **Commencez ici pour des performances**
2. Guide de configuration - Compatibilité avec les hyperviseurs
3. Guide de configuration - Configuration SR-IOV Proxmox
4. Dépannage - Problèmes Proxmox

Optimisation des performances

1. Guide des modes XDP - **Activez XDP natif pour un gain de performance de 5-10x**
2. Guide d'architecture - Optimisation des performances
3. Guide de configuration - Modes de fixation XDP
4. Guide de surveillance - Métriques de performance
5. Dépannage - Problèmes de performance

Compréhension du 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 la capacité

Guide d'Architecture OmniUPF

Table des Matières

Aperçu

OmniUPF tire parti de eBPF (Extended Berkeley Packet Filter) et de XDP (eXpress Data Path) pour atteindre des performances de niveau opérateur pour le traitement des paquets 5G/LTE. En exécutant la logique de traitement des paquets directement dans le noyau Linux, OmniUPF élimine le surcoût du traitement en espace utilisateur et atteint un débit multi-gigabit avec une latence en microsecondes.

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 surcharge de mutex/spinlock

Prêt pour le Déchargement Matériel:

- Le mode de déchargement XDP prend en charge l'exécution sur SmartNIC
- Compatible avec les cartes réseau prenant en charge XDP

- Repli sur des 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 isolés de s'exécuter dans l'espace noyau sans modifier le code source du noyau ou charger des modules du 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 surcharge d'interprétation)
- **Flexibilité**: Peut être mis à jour à l'exécution sans redémarrage du noyau
- **Observabilité**: Tracage et statistiques intégrés

Cycle de Vie des Programmes eBPF

Cartes eBPF

Les cartes eBPF sont des structures de données du 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 vidages 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:

```
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
- Les paquets sont 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:

```
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 a alloué SKB
- Émulation logicielle du comportement XDP
- Fonctionne sur n'importe quelle interface réseau
- Utile pour les tests et le développement

Configuration:

```
xdp_attach_mode: generic
```

Cas d'Utilisation:

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

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	Pas actuellement utilisé

Code de Retour	Signification	Utilisation dans OmniUPF
	nouveau par la même interface	
`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 met en œuvre 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

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);  // Depuis l'en-tête IP
   ```

2. **Étend l'En-tête du Paquet**:

   ```
   // Ajouter de l'espace pour : IP Externe + 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 Externe**:

   ```
   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 tampons 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 Tampons

Paramètre	Par Défaut	Description
Max par FAR	10,000 paquets	Nombre maximum de paquets mis en tampon par FAR
Max Total	100,000 paquets	Nombre maximum total de paquets mis en tampon
TTL du Paquet	30 secondes	Temps avant que les paquets mis en tampon expirent
Port de Tampon	22152	Port UDP pour la livraison de tampons
Intervalle de Nettoyage de Tampon	60 secondes	Fréquence de vérification des paquets expirés

Application de la QoS

Algorithme de Limitation de Débit

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

Mise en Œuvre de la Fenêtre Glissante

Algorithme (depuis `qer.h`):

```c
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 uplink et downlink
- **Mises à Jour Atomiques**: Utilise des pointeurs volatils pour un accès concurrent
- **MBR = 0**: Considéré comme une bande passante illimitée

Exemple de Calcul de QoS

Scénario: MBR = 100 Mbps, Taille du 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 a été 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 de Paquet Maximum**:

```
Max PPS = (100 Mbps ÷ 8) ÷ 1500 octets = 8,333 paquets/seconde
Intervalle entre 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
NIC RX	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 du 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
NIC TX	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-cœurs: presque linéaire jusqu'à 8 cœurs

Utilisation du CPU par Taux de Paquet:

```
CPU % ≈ (Taux de Paquet / 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 cache)
- Recherche par hachage: ~300 ns (miss cache)
- Recherche par tableau: ~50 ns (toujours hit cache)

Bande Passante Mémoire Requise:

```
Bande Passante = Taux de Paquet × (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:

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. Épingler les programmes eBPF à des cœurs spécifiques

Réglage du NIC:

1. Augmenter la taille du tampon RX
2. Activer les NIC multi-files d'attente (RSS)
3. Utiliser le directeur de flux pour le routage 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

# Régler 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 et déploiement UPF
- **Guide de Gestion des Règles** - Détails sur PDR, FAR, QER, URR
- **Guide de Surveillance** - Surveillance de performance et métriques
- **Guide d'Opérations de l'Interface Web** - Utilisation du panneau de contrôle
- **Guide de Dépannage** - Problèmes courants et diagnostics

Guide de Configuration OmniUPF

Table des Matières

Aperçu

OmniUPF est une fonction de plan utilisateur polyvalente qui peut fonctionner dans plusieurs modes pour prendre en charge à la fois les réseaux de cœur 4G (EPC) et 5G. La configuration est gérée via des fichiers de configuration YAML.

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 des 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, voir le Guide des Modes XDP.

Comparaison des Modes

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

Mode Générique (Par Défaut)

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

Avantages :

- Fonctionne avec **n'importe** quelle interface réseau
- Aucune exigence spéciale 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 ont déjà passé par le pilote avant le traitement XDP

Configuration :

```
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 VMs avec SR-IOV

Inconvénients :

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

Configuration :

```
xdp_attach_mode: native
```

Meilleur pour :

- Déploiements de production sur bare metal
- VMs avec passthrough 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)
- Zéro surcharge 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 :

```
xdp_attach_mode: offload
```

Meilleur pour :

- Déploiements de production à très haut débit
- Informatique en périphérie avec des exigences de latence strictes
- 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
`interface_name`	Interfaces réseau pour le trafic N3/N6/N9 (points de rattachement XDP)	Liste	`[lo]`
`n3_address`	Adresse IPv4 pour l'interface N3 (GTP-U de RAN)	IP	`127.0.0.1`
`n9_address`	Adresse IPv4 pour l'interface N9 (UPF-à-UPF pour ULCL)	IP	Identique à `n3_address`

Exemple :

```
interface_name: [eth0, eth1]
n3_address: 10.100.50.233
n9_address: 10.100.50.234
```

Configuration PFCP

Paramètre	Description	Type	Par Défaut
`pfcp_address`	Adresse locale pour le serveur PFCP (interface N4/Sxb/Sxc)	Host:Port	`:8805`
`pfcp_node_id`	ID de nœud local pour le	IP	`127.0.0.1`

Paramètre	Description	Type	Par Défaut
	protocole PFCP		
`pfcp_remote_node`	Pairs PFCP distants (SMF/ PGW-C/SGW-C) à connecter	Liste	`[]`
`association_setup_timeout`	Délai entre les demandes de configuration d'association (secondes)	Entier	5
`heartbeat_retries`	Nombre de tentatives de heartbeat avant de déclarer le pair mort	Entier	3
`heartbeat_interval`	Intervalle de heartbeat PFCP (secondes)	Entier	5
`heartbeat_timeout`	Délai d'attente pour le heartbeat PFCP (secondes)	Entier	5

Exemple :

```yaml
pfcp_address: :8805
pfcp_node_id: 10.100.50.241
pfcp_remote_node:
  - 10.100.50.10  # OmniSMF
  - 10.100.60.20  # OmniPGW-C
heartbeat_interval: 10
heartbeat_retries: 5
```

API et Surveillance

Paramètre	Description	Type	Par Défaut
`api_address`	Adresse locale pour le serveur API REST	Host:Port	`:8080`
`metrics_address`	Adresse locale pour le point de terminaison des métriques Prometheus	Host:Port	`:9090`
`logging_level`	Niveau de journalisation (`trace`, `debug`, `info`, `warn`, `error`)	Chaîne	`info`

Exemple :

```yaml
api_address: :8080
metrics_address: :9090
logging_level: debug
```

Gestion du Chemin GTP

Paramètre	Description	Type	Par Défaut
`gtp_peer`	Liste des pairs GTP pour les keepalives Echo Request	Liste	`[]`
`gtp_echo_interval`	Intervalle entre les Echo Requests GTP (secondes)	Entier	`10`

Exemple :

```yaml
gtp_peer:
  - 10.100.50.50:2152  # gNB
  - 10.100.50.60:2152  # Un autre UPF pour N9
gtp_echo_interval: 15
```

Capacité de Carte eBPF

Paramètre	Description	Type	Par Défaut	Auto-calculé
`max_sessions`	Nombre maximum de sessions simultanées	Entier	`65535`	Utilisé pour calculer les tailles de carte
`pdr_map_size`	Taille de la carte eBPF PDR	Entier	`0`	`max_sessions × 2`
`far_map_size`	Taille de la carte eBPF FAR	Entier	`0`	`max_sessions × 2`
`qer_map_size`	Taille de la carte eBPF QER	Entier	`0`	`max_sessions`
`urr_map_size`	Taille de la carte eBPF URR	Entier	`0`	`max_sessions × 2`

Remarque : Définir les tailles de carte à `0` (par défaut) permet le calcul automatique basé sur `max_sessions`. Remplacez par des valeurs spécifiques si un dimensionnement personnalisé est nécessaire.

Exemple :

```yaml
max_sessions: 100000
# Les cartes seront dimensionnées automatiquement :
# PDR : 200 000 entrées
# FAR : 200 000 entrées
# QER : 100 000 entrées
# URR : 200 000 entrées
```

Exemple de dimensionnement personnalisé :

```yaml
max_sessions: 50000
pdr_map_size: 131070  # Taille personnalisée
far_map_size: 131070
```

```
qer_map_size: 65535
urr_map_size: 131070
```

Configuration de Tampon

Paramètre	Description	Type	Par Défaut
`buffer_port`	Port UDP pour les paquets tamponnés provenant de l'eBPF	Entier	`22152`
`buffer_max_packets`	Nombre maximum de paquets à tamponner par FAR	Entier	`10000`
`buffer_max_total`	Nombre total maximum de paquets tamponnés (0=illimité)	Entier	`100000`
`buffer_packet_ttl`	TTL pour les paquets tamponnés en secondes (0=pas d'expiration)	Entier	`30`
`buffer_cleanup_interval`	Intervalle de nettoyage du tampon en secondes (0=pas de nettoyage)	Entier	`60`

Exemple :

```
buffer_port: 22152
buffer_max_packets: 20000
buffer_max_total: 200000
buffer_packet_ttl: 60
buffer_cleanup_interval: 30
```

Drapeaux de Fonctionnalité

Paramètre	Description	Type	Par Défaut
`feature_ueip`	Activer l'allocation d'IP UE par OmniUPF	Booléen	`false`
`ueip_pool`	Pool d'IP pour l'allocation d'IP UE (nécessite `feature_ueip`)	CIDR	`10.60.0.0/24`
`feature_ftup`	Activer l'allocation de F-TEID par OmniUPF	Booléen	`false`
`teid_pool`	Taille du pool TEID pour l'allocation de F-TEID (nécessite `feature_ftup`)	Entier	`65535`

Exemple (allocation d'IP UE) :

```
feature_ueip: true
ueip_pool: 10.45.0.0/16  # Allouer des IP UE à partir de ce pool
```

Exemple (allocation de F-TEID) :

```
feature_ftup: true
teid_pool: 1000000  # Autoriser jusqu'à 1M d'allocations de TEID
```

Méthodes de Configuration

Fichier de Configuration YAML (Recommandé)

Fichier : `config.yml`

```yaml
# Configuration Réseau
interface_name: [eth0]
n3_address: 10.100.50.233
n9_address: 10.100.50.233
xdp_attach_mode: native

# Configuration PFCP
pfcp_address: :8805
pfcp_node_id: 10.100.50.241
pfcp_remote_node:
  - 10.100.50.10

# API et Surveillance
api_address: :8080
metrics_address: :9090
logging_level: info

# Capacité
max_sessions: 100000

# Pairs GTP
gtp_peer:
  - 10.100.50.50:2152
gtp_echo_interval: 10

# Fonctionnalités
feature_ueip: true
ueip_pool: 10.45.0.0/16
feature_ftup: false

# Tampon
buffer_max_packets: 15000
buffer_packet_ttl: 45
```

Démarrer OmniUPF :

```
./eupf --config /path/to/config.yml
```

Compatibilité Hyperviseur

Aperçu

OmniUPF est compatible avec tous les principaux hyperviseurs et plateformes de virtualisation. Le mode de rattachement 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 Bridge (XDP Générique)

Cas d'utilisation : Réseau VM standard

Configuration :

- Dispositif Réseau : VirtIO ou E1000
- Mode XDP : `generic`
- Performance : ~1-2 Mpps

Paramètres VM Proxmox :

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

Configuration OmniUPF :

```
interface_name: [eth0]
xdp_attach_mode: generic
```

---

2. Passthrough SR-IOV (XDP Natif)

Cas d'utilisation : Production à 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 VM Proxmox :

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

Configuration OmniUPF :

```
interface_name: [ens1f0]  # Nom de la VF SR-IOV
xdp_attach_mode: native
```

---

3. Passthrough 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 de la NIC)

Paramètres 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 :

```
interface_name: [ens1f0]
xdp_attach_mode: native  # ou 'offload' pour SmartNIC
```

KVM/QEMU

Mode Bridge :

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

Passthrough 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/Bridge (XDP Générique uniquement) :

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

Compatibilité NIC

Comprendre Mpps vs Débit

Paquets par seconde (Mpps) et débit (Gbps) ne sont pas directement équivalents - la relation dépend entièrement de la taille des paquets. Le trafic des réseaux mobiles varie considérablement en taille de paquet, des petits paquets VoIP aux grandes 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 minimale** : 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 × 1 000 000 pps × 8 bits/octet = 520 Mbps
```

Paquet GTP-U Maximum (N3 avec MTU de 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 × 1 000 000 pps × 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 minimale** : 1 octet
- **Total** : 29 octets

Débit à 1 Mpps avec des paquets N6 minimum :

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

Paquet N6 Maximum (MTU de 1500) :

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

Débit à 1 Mpps avec des paquets N6 maximum :

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

Exemples de Performance dans le Monde Réel

NIC Intel X710 (capacité 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 volumineux

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 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 volumineux

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

Pourquoi cela importe-t-il pour les réseaux mobiles :

Le trafic mobile est **hautement variable** en taille de paquet et la surcharge GTP-U (36 octets) impacte significativement la performance des petits paquets :

Taille de paquet interne (données utilisateur réelles) :

- **VoIP (codec AMR-WB)** : 65-80 octets → Avec GTP-U : 101-116 octets
- **Données de capteur IoT** : 50-200 octets → Avec GTP-U : 86-236 octets
- **Navigation Web (HTTP/3)** : 400-800 octets → Avec GTP-U : 436-836 octets
- **Streaming vidéo** : 1200-1450 octets → Avec GTP-U : 1236-1486 octets
- **Téléchargements volumineux** : 1500 octets → Avec GTP-U : 1536 octets

Impact de la surcharge GTP-U :

- Petits paquets (< 200 octets) : **~35-70% de surcharge** - Mpps est le facteur limitant
- Paquets moyens (200-800 octets) : **~5-20% de surcharge** - Limitation mixte
- Grands paquets (> 1200 octets) : **~3% de surcharge** - La vitesse de lien est le facteur limitant

Planification de Performance :

Une NIC évaluée à **10 Mpps** atteindra sur l'interface N3 :

- **Trafic lourd en VoIP** (paquets internes de 100 octets) : ~1,0 Gbps (la surcharge GTP-U domine)
- **Mélange mobile moderne** (paquets internes moyens de 1200 octets) : ~9,9 Gbps
- **Trafic lourd en vidéo** (paquets internes de 1400 octets) : ~11,5 Gbps
- **Débit maximum** (paquets internes de 1500 octets) : ~12,3 Gbps

Sur l'interface N6 (pas de surcharge GTP-U) :

- **Mélange mobile moderne** (paquets de 1200 octets) : ~9,6 Gbps à 10 Mpps
- **Débit maximum** (paquets de 1500 octets) : ~12,0 Gbps à 10 Mpps

Règle de Pouce pour UPF Mobile :

- **Trafic de petits paquets** (VoIP, IoT, signalisation) : Mpps est limitant - prévoyez 1-2 Gbps par 10 Mpps
- **Trafic mobile moderne** (moyenne de 1200 octets) : Prévoyez ~9-10 Gbps par 10 Mpps de capacité
- **Trafic lourd en vidéo** (streaming, téléchargements) : Prévoyez ~10-12 Gbps par 10 Mpps de capacité
- **Considérez toujours à la fois N3 et N6** - N3 a une surcharge GTP-U, N6 n'en a pas

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)	Scénario de Déploiement Réaliste
1 Mpps	~1,0 Gbps	~1,0 Gbps	Petite cellule, passerelle IoT
5 Mpps	~4,9 Gbps	~4,8 Gbps	Cellule moyenne, entreprise
10 Mpps	~9,9 Gbps	~9,6 Gbps	Grande cellule, petite ville
20 Mpps	~19,7 Gbps	~19,2 Gbps	Zone métropolitaine, ville

Capacité Mpps NIC	Débit N3 (GTP-U)	Débit N6 (IP natif)	Scénario de Déploiement Réaliste
			moyenne
40 Mpps	~39,4 Gbps	~38,4 Gbps	Grande métropole, hub régional

Remarque : Ces estimations supposent une taille de charge utile moyenne de 1200 octets, qui est représentative du trafic mobile moderne dominé par le streaming vidéo, les réseaux sociaux et les applications cloud. Le débit réel variera en fonction du mélange de trafic.

Pilotes Réseau Compatibles XDP

OmniUPF nécessite des pilotes réseau avec support XDP pour les modes **natif** et **offload**. Le mode générique fonctionne avec **n'importe** quelle 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

NIC 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 + Offload	~40 Mpps
Mellanox ConnectX-4	mlx4	Limité	Générique	~2 Mpps

NIC 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	Offload	~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
```

NIC Recommandées par Cas d'Utilisation

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

Cas d'Utilisation	NIC Recommandée	Mode	Capacité Mpps	Débit (N3)	Scénario de Déploiement
Test/ Développement	Toute NIC (VirtIO, E1000)	Générique	1-2 Mpps	1-2 Gbps	Test en laboratoire, PoC
Site de Petite Cellule	Intel X710, Mellanox CX-5	Natif	5-10 Mpps	5-10 Gbps	Cellule rurale, entreprise
Cellule Moyenne/ Métropolitaine	Intel E810, Mellanox CX-6	Natif	10-20 Mpps	10-20 Gbps	Cellule urbaine, petite ville
Grande Métropole	Mellanox CX-6, Intel E810 (dual)	Natif	20-40 Mpps	20-40 Gbps	Zone métropolitaine, ville moyenne

Cas d'Utilisation	NIC Recommandée	Mode	Capacité Mpps	Débit (N3)	Scénario de Déploiement
Hub Régional	Mellanox BlueField, Netronome Agilio	Offload	40+ Mpps	40+ Gbps	Agrégation régionale
VM Proxmox (Bridge)	VirtIO	Générique	1-2 Mpps	1-2 Gbps	Test uniquement
VM Proxmox (SR-IOV)	Intel X710/E810 VF, Mellanox CX-5 VF	Natif	5-10 Mpps	5-10 Gbps	VM de production

Estimations de Débit :

- Basé sur une taille de paquet moyenne de 1200 octets avec encapsulation GTP-U (1236 octets sur N3)
- Le débit N6 légèrement inférieur (~9,6 Gbps par 10 Mpps) en raison de l'absence de surcharge GTP-U
- La performance réelle varie avec le mélange de trafic - les réseaux lourds en VoIP verront un débit inférieur

---

Ressources Supplémentaires

Documentation Officielle XDP :

- Projet XDP
- Documentation XDP du Noyau

Listes de Compatibilité NIC :

- Liste de Support XDP de Cilium
- Pilotes XDP de 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
interface_name: [eth0]
xdp_attach_mode: generic
api_address: :8080
pfcp_address: :8805
pfcp_node_id: 127.0.0.1
```

```yaml
n3_address: 127.0.0.1
metrics_address: :9090
logging_level: debug
max_sessions: 1000
```

Exemple 2 : Production Bare Metal (Mode Natif)

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

```yaml
# Configuration bare metal de production
interface_name: [ens1f0, ens1f1]  # N3 sur ens1f0, N6 sur ens1f1
xdp_attach_mode: native
api_address: :8080
pfcp_address: 10.100.50.241:8805
pfcp_node_id: 10.100.50.241
n3_address: 10.100.50.233
n9_address: 10.100.50.234
metrics_address: :9090
logging_level: info
max_sessions: 500000
gtp_peer:
  - 10.100.50.10:2152  # gNB 1
  - 10.100.50.11:2152  # gNB 2
gtp_echo_interval: 30
pfcp_remote_node:
  - 10.100.50.50  # OmniSMF
heartbeat_interval: 10
feature_ueip: true
ueip_pool: 10.45.0.0/16
buffer_max_packets: 50000
buffer_packet_ttl: 60
```

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

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

```yaml
# Configuration Proxmox SR-IOV
interface_name: [ens1f0]  # VF SR-IOV
xdp_attach_mode: native
api_address: :8080
pfcp_address: 192.168.100.10:8805
pfcp_node_id: 192.168.100.10
n3_address: 192.168.100.10
metrics_address: :9090
logging_level: info
max_sessions: 100000
```

```yaml
gtp_peer:
  - 192.168.100.50:2152
gtp_echo_interval: 15
pfcp_remote_node:
  - 192.168.100.20  # SMF
```

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

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

```yaml
# Configuration PGW-U
interface_name: [eth0]
xdp_attach_mode: native
api_address: :8080
pfcp_address: 10.200.1.10:8805
pfcp_node_id: 10.200.1.10
n3_address: 10.200.1.10  # Interface S5/S8 (GTP-U)
metrics_address: :9090
logging_level: info
max_sessions: 200000
gtp_peer:
  - 10.200.1.50:2152  # SGW-U
gtp_echo_interval: 20
pfcp_remote_node:
  - 10.200.2.10  # OmniPGW-C (interface Sxb)
heartbeat_interval: 5
```

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

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

```yaml
# Configuration multi-mode
interface_name: [eth0, eth1]
xdp_attach_mode: native
api_address: :8080
pfcp_address: :8805
pfcp_node_id: 10.50.1.100
n3_address: 10.50.1.100
n9_address: 10.50.1.101
metrics_address: :9090
logging_level: info
max_sessions: 300000
gtp_peer:
  - 10.50.2.10:2152  # gNB 5G
  - 10.50.2.20:2152  # eNodeB 4G (via SGW-U)
gtp_echo_interval: 15
```

```
pfcp_remote_node:
  - 10.50.3.10  # OmniSMF (5G)
  - 10.50.3.20  # OmniPGW-C (4G)
heartbeat_interval: 10
feature_ueip: true
ueip_pool: 10.60.0.0/16
```

Exemple 6 : Mode Offload SmartNIC

Scénario : Déploiement à très haut débit avec SmartNIC Netronome Agilio CX

```
# Configuration de déchargement SmartNIC
interface_name: [enp1s0np0]  # Interface SmartNIC
xdp_attach_mode: offload
api_address: :8080
pfcp_address: 10.10.1.50:8805
pfcp_node_id: 10.10.1.50
n3_address: 10.10.1.50
metrics_address: :9090
logging_level: warn  # Réduire la surcharge
max_sessions: 1000000
pdr_map_size: 2000000
far_map_size: 2000000
qer_map_size: 1000000
gtp_peer:
  - 10.10.2.10:2152
  - 10.10.2.20:2152
  - 10.10.2.30:2152
gtp_echo_interval: 30
pfcp_remote_node:
  - 10.10.3.10
heartbeat_interval: 15
buffer_max_packets: 100000
buffer_max_total: 1000000
```

Dimensionnement de Carte et Planification de Capacité

Dimensionnement Automatique (Recommandé)

Définissez `max_sessions` et laissez OmniUPF calculer les tailles de carte automatiquement :

```
max_sessions: 100000
# Tailles auto-calculées :
```

```
# PDR : 200 000 entrées (2 × max_sessions)
# FAR : 200 000 entrées (2 × max_sessions)
# QER : 100 000 entrées (1 × max_sessions)
# URR : 200 000 entrées (2 × max_sessions)
```

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

Dimensionnement Manuel

Remplacez le calcul automatique par des exigences personnalisées :

```yaml
max_sessions: 100000
pdr_map_size: 300000  # Supporter plus de PDR par session
far_map_size: 200000
qer_map_size: 150000  # Plus de QER que par défaut
urr_map_size: 200000
```

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 Mémoire

Utilisation de mémoire par session :

- PDR : 2 × 212 B = 424 B
- FAR : 2 × 20 B = 40 B
- QER : 1 × 36 B = 36 B
- URR : 2 × 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 2× l'utilisation estimée :

```
# Vérifiez la limite actuelle
ulimit -l

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

Documentation Connexe

- **Guide d'Architecture** - Détails techniques eBPF/XDP et optimisation de performance
- **Guide de Gestion des R��gles** - Configuration PDR, FAR, QER, URR
- **Guide de Surveillance** - Statistiques, surveillance de capacité et alertes
- **Guide de l'Interface Web** - Opérations du panneau de contrôle
- **Guide d'Opérations** - Aperçu de l'architecture et du déploiement UPF

Guide de Surveillance

Table des Matières

Aperçu

La surveillance efficace d'OmniUPF est essentielle pour maintenir la qualité du service, prévenir l'épuisement des ressources 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 natif descendant)
- **TX N6**: Paquets transmis au Réseau de Données (IP natif montant)
- **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:

- **Avorté**: Le programme XDP a rencontré une erreur (devrait toujours être 0)
- **Jeter**: Paquets intentionnellement rejetés par le programme XDP
- **Passer**: Paquets passés à la pile réseau pour un traitement ultérieur
- **Rediriger**: Paquets directement redirigés vers l'interface de sortie
- **TX**: Paquets transmis via XDP

Interprétation:

- **Avorté > 0**: Problème critique avec le programme eBPF ou la 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

Détail de la répartition des protocoles de paquets et compteurs de traitement.

Compteurs de Protocoles:

- **RX ARP**: Paquets du Protocole de Résolution d'Adresse
- **RX GTP ECHO**: Demande/Réponse GTP-U Echo (keepalive)
- **RX GTP AUTRE**: Autres messages de contrôle GTP
- **RX GTP PDU**: Données utilisateur encapsulées GTP-U (trafic principal)
- **RX GTP INEXP**: Types de paquets GTP inattendus
- **RX ICMP**: Protocole de Message de Contrôle Internet (ping, erreurs)
- **RX ICMP6**: Paquets ICMPv6
- **RX IP4**: Paquets IPv4
- **RX IP6**: Paquets IPv6
- **RX AUTRE**: 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 configuration

---

Statistiques de Routage

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

Recherche FIB IPv4:

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

Recherche FIB IPv6:

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

Indicateurs de Performance:

- **Taux de Hits de Cache Élevé**: Indique une bonne performance du cache

de routage
- **Compteur OK Élevé**: Confirme que les tables de routage sont correctement configurées
- **Recherches Basses ou Nulles**: 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 Mo
- **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 Mo
- **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 Mo
- **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 Mo

- **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 QoS)
- **Utilisation de Mémoire**: ~2.3 Mo
- **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 Mo
- **Criticité**: Faible - N'affecte que la facturation

Seuils de Capacité

Seuil	Action Requise
0-50% (Vert)	Fonctionnement normal - Aucune action requise
50-70% (Jaune)	Prudence - 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, les 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

Taux de Traitement des Paquets

Calcul:

```
Taux de Paquets (pps) = (Delta de 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 d'avortements XDP en augmentation
- Utilisation élevée du CPU
- Augmentation des pertes de paquets
- Augmentation de la latence

Calcul du Débit

Calcul:

```
Débit (Mbps) = (Delta de Compte de Bytes × 8) / (Delta de Temps en
secondes × 1,000,000)
```

Exemple:

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

Planification de 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 sécurité

Taux de Pertes

Calcul:

```
Taux de Pertes (%) = (Paquets Perdus / Total 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 les 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 d'avortements XDP > 0
- Taux de pertes > 5%
- Échec de vérification de la santé de l'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 plus vieux que 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
2. **Surveillance des Logs**: Analyser les logs d'OmniUPF pour des motifs d'erreurs
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:

```
Sessions Maximales = 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

Sessions Maximales = 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 Mo
- Carte FAR: 131,070 × 20 B = 2.6 Mo

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

Considérations de 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 sécurité**:

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

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 connectivité médiocre

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. avortements

Causes Courantes:

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

Résolution:

- Augmenter le MBR du 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 avorté > 0

Diagnostic:

1. Naviguer vers Statistiques → Statistiques XDP
2. Vérifier le compteur d'avortements
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 respecte les 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 les 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 de session

Dégradation de la Performance

Symptômes: Latence élevée, faible débit, saturation du 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 du CPU sur l'hôte UPF

4. Vérifier l'utilisation des interfaces N3/N6

Causes Courantes:

- Trafic dépassant la capacité de l'UPF
- Cœurs de 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 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

- **Guide des Opérations UPF** - Architecture générale et opérations de l'UPF
- **Guide de Gestion des Règles** - Configuration 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

N9 Loopback : Exécution de SGWU et PGWU sur la même instance

Vue d'ensemble

OmniUPF prend en charge l'exécution des fonctions **SGWU (Serving Gateway User Plane)** et **PGWU (PDN Gateway User Plane)** sur la **même instance** avec un **loopback N9 à latence nulle**. Ce mode de déploiement est idéal pour :

- **Déploiements EPC 4G simplifiés** - Une seule instance UPF au lieu de deux
- **Optimisation des coûts** - Réduction de l'infrastructure et de la complexité opérationnelle
- **Edge computing** - Minimiser la latence pour les scénarios de rupture locale
- **Environnements de laboratoire/test** - Plan utilisateur EPC complet sur un seul serveur

Lorsqu'il est configuré avec la même adresse IP pour les interfaces N3 et N9, OmniUPF **détecte automatiquement** le trafic circulant entre les rôles SGWU et PGWU et le traite **entièrement en eBPF** sans jamais envoyer de paquets à l'interface réseau.

Comment ça fonctionne

Déploiement traditionnel (Deux instances)

Flux de paquets :

1. eNodeB → SGWU : Le paquet GTP (TEID=100) arrive sur S1-U
2. SGWU : Correspond au PDR de montée, encapsule dans un nouveau tunnel GTP (TEID=200)
3. **Paquet envoyé sur le réseau N9 physique** à l'instance PGWU
4. PGWU : Reçoit GTP (TEID=200), décapsule, transmet à Internet
5. **Total : 2 passages XDP + 1 saut réseau**

Déploiement N9 Loopback (Instance unique)

Flux de paquets avec N9 Loopback :

1. eNodeB → rôle SGWU : Le paquet GTP (TEID=100) arrive sur S1-U
2. Rôle SGWU : Correspond au PDR de montée
3. **Détection de loopback :** IP de destination = IP locale (10.0.1.10)
4. **Traitement sur place :** Met à jour le TEID GTP à 200 (session PGWU)
5. Rôle PGWU : Décapsule, transmet à Internet
6. **Total : 1 passage XDP, zéro saut réseau**

Avantage de performance : Transfert interne sub-microseconde contre millisecondes pour un aller-retour réseau

---

Détails du traitement des paquets

Flux de montée : eNodeB → SGWU → PGWU → Internet

Chemin du code : `cmd/ebpf/xdp/n3n6_entrypoint.c` lignes 349-403

Étapes clés :

1. **Recevoir :** Paquet GTP de eNodeB avec TEID=100
2. **Correspondance PDR :** Recherche du PDR de montée pour la session SGWU (TEID=100)
3. **Action FAR :** Encapsuler en GTP avec TEID=200, transmettre à 10.0.1.10
4. **Vérification de Loopback :** `is_local_ip(10.0.1.10)` retourne TRUE
5. **Mettre à jour TEID :** Changer `ctx->gtp->teid` de 100 à 200 (en mémoire du noyau)
6. **Re-traiter :** Recherche du PDR pour TEID=200 (session PGWU)
7. **Action FAR :** Supprimer l'en-tête GTP, transmettre à Internet
8. **Route :** Envoyer le paquet IP simple à l'interface N6

Flux de descente : Internet → PGWU → SGWU → eNodeB

Chemin du code : `cmd/ebpf/xdp/n3n6_entrypoint.c` lignes 137-194 (IPv4), 265-322 (IPv6)

Étapes clés :

1. **Recevoir :** Paquet IP simple de l'Internet destiné à l'UE (10.60.0.1)
2. **Correspondance PDR :** Recherche du PDR de descente par l'IP de l'UE (session PGWU)
3. **Action FAR :** Encapsuler en GTP avec TEID=200, transmettre à 10.0.1.10
4. **Vérification de Loopback :** `is_local_ip(10.0.1.10)` retourne TRUE
5. **Ajouter GTP :** Encapsuler le paquet avec TEID=200
6. **Re-traiter :** Recherche du PDR pour TEID=200 (session SGWU)
7. **Action FAR :** Mettre à jour le tunnel GTP vers l'eNodeB TEID=100
8. **Route :** Envoyer le paquet GTP à l'interface S1-U (eNodeB)

Configuration

Exigences

Plan de contrôle :

- **SGWU-C** : Doit se connecter à l'interface PFCP d'OmniUPF (par exemple, `192.168.1.10:8805`)
- **PGWU-C** : Doit se connecter à la **même** interface PFCP d'OmniUPF

Réseau :

- **Une seule adresse IP** pour les interfaces N3 et N9
- **Adresses IP différentes** pour SGWU-C et PGWU-C (si exécuté sur le même hôte, utilisez des ports différents)

Configuration d'OmniUPF

config.yml :

```yaml
# Interfaces réseau
interface_name: [eth0]              # Interface unique pour S1-U et N9
xdp_attach_mode: native             # Utiliser native pour les meilleures performances

# Interface PFCP
pfcp_address: ":8805"               # Écouter sur toutes les interfaces, port 8805
pfcp_node_id: "192.168.1.10"        # ID de nœud PFCP d'OmniUPF

# Interfaces de plan utilisateur
n3_address: "10.0.1.10"             # IP de l'interface S1-U/N3
n9_address: "10.0.1.10"             # IP de l'interface N9 (IDENTIQUE à N3)

# APIs
api_address: ":8080"                # API REST
metrics_address: ":9090"            # Métriques Prometheus

# Pools de ressources
ueip_pool: "10.60.0.0/16"           # Pool d'adresses IP UE
teid_pool: 65535                    # Pool d'allocation TEID

# Capacité
```

```
max_sessions: 100000                    # Nombre maximum de sessions UE
simultanées
```

Configuration clé :

- ⚠ **`n3_address` et `n9_address` DOIVENT être identiques** pour activer le loopback
- ⚠ Adresse d'écoute PFCP unique pour les deux plans de contrôle
- ⚠ Nombre suffisant de `max_sessions` pour la charge combinée SGWU + PGWU

Configuration du plan de contrôle

Configuration SGWU-C

```
# Pointer vers l'interface PFCP d'OmniUPF
upf_pfcp_address: "192.168.1.10:8805"

# Interface S1-U (identique à l'adresse n3 d'OmniUPF)
sgwu_s1u_address: "10.0.1.10"

# Interface N9 pour le transfert vers PGWU (identique à OmniUPF)
sgwu_n9_address: "10.0.1.10"
```

Configuration PGWU-C

```
# Pointer vers la MÊME interface PFCP d'OmniUPF
upf_pfcp_address: "192.168.1.10:8805"

# Interface N9 (reçoit de SGWU)
pgwu_n9_address: "10.0.1.10"

# Interface SGi pour la connectivité Internet
pgwu_sgi_address: "192.168.100.1"
```

Important :

- Les deux plans de contrôle se connectent au **même point de terminaison PFCP** (`:8805`)
- OmniUPF crée **des associations PFCP séparées** pour SGWU-C et PGWU-C
- Les sessions sont isolées par plan de contrôle (suivies par ID de nœud)

Exemple de flux de session

Attachement UE et établissement de session PDU

Scénario : L'UE se connecte au réseau, établit une session de données

Sessions PFCP créées :

Session SGWU (depuis OmniSGW-C) :

- **PDR de montée :** Correspondre TEID=100 (depuis eNodeB) → FAR : Encapsuler TEID=200, dst=10.0.1.10
- **PDR de descente :** Correspondre TEID=200 (depuis PGWU) → FAR : Mettre à jour le tunnel TEID=100, transmettre à eNodeB

Session PGWU (depuis OmniPGW-C) :

- **PDR de montée :** Correspondre TEID=200 (depuis SGWU) → FAR : Décapsuler, transmettre à Internet
- **PDR de descente :** Correspondre IP UE=10.60.0.1 → FAR : Encapsuler TEID=200, dst=10.0.1.10

Surveillance et vérification

Vérifier que le N9 Loopback est actif

Vérifiez les journaux XDP :

```
# Voir la sortie de débogage eBPF en temps réel
sudo cat /sys/kernel/debug/tracing/trace_pipe | grep loopback
```

Sortie attendue :

```
upf: [n3] session for teid:100 -> 200 remote:10.0.1.10
upf: [n9-loopback] self-forwarding detected, processing inline
TEID:200
upf: [n9-loopback] decapsulated, routing to N6

upf: [n6] use mapping 10.60.0.1 -> teid:200
upf: [n6-loopback] downlink self-forwarding detected, processing
inline TEID:200
upf: [n6-loopback] SGWU updating GTP tunnel to eNodeB TEID:100
upf: [n6-loopback] forwarding to eNodeB
```

Surveiller les sessions via l'API REST

Lister les associations PFCP :

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

Sortie attendue :


```
{
  "associations": [
    {
      "node_id": "sgwc.example.com",
      "address": "192.168.1.20:8805",
      "sessions": 1000
    },
    {
      "node_id": "pgwc.example.com",
      "address": "192.168.1.21:8805",
      "sessions": 1000
    }
  ],
  "total_sessions": 2000
}
```


Vérifiez deux associations séparées (une pour SGWU-C, une pour PGWU-C)

Lister les sessions actives :

```
curl http://localhost:8080/api/v1/sessions | jq '.sessions[] |
{local_seid, ue_ip, uplink_teid}'
```

Sortie attendue :


```
{
  "local_seid": 12345,
  "ue_ip": "10.60.0.1",
  "uplink_teid": 100
}
{
  "local_seid": 67890,
  "ue_ip": "10.60.0.1",
  "uplink_teid": 200
}
```


Chaque UE a DEUX sessions :

- Session de SGWU-C (TEID=100, interface S1-U)
- Session de PGWU-C (TEID=200, interface N9)

Métriques de performance

Vérifiez les statistiques de paquets :

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

Métriques clés :

- `xdp_processed` : Total des paquets traités en eBPF
- `xdp_pass` : Paquets passés à la pile réseau (devrait être zéro pour le trafic de loopback)
- `xdp_redirect` : Paquets transférés via redirection XDP
- `xdp_tx` : Paquets transmis (le trafic de loopback utilise cela)

Pour le trafic de loopback N9 :

- `xdp_pass` devrait être **minimal** (seulement le trafic non-loopback)
- `xdp_tx` ou `xdp_redirect` comptent le transfert de loopback

Dépannage

Trafic N9 allant au réseau au lieu de Loopback

Symptôme : Paquets envoyés à l'interface réseau, latence élevée

Cause racine : `n3_address` ≠ `n9_address`

Solution :

```
# FAUX :
n3_address: "10.0.1.10"
n9_address: "10.0.1.20"   # IP différente, pas de loopback !

# CORRECT :
n3_address: "10.0.1.10"
n9_address: "10.0.1.10"   # Même IP, active le loopback
```

Vérification :

```
curl http://localhost:8080/api/v1/dataplane_config | jq
```

Devrait montrer :

```
{
  "n3_ipv4_address": "10.0.1.10",
  "n9_ipv4_address": "10.0.1.10"
```

```
}
```

PDR non trouvé après Loopback

Symptôme : Les journaux montrent `[n9-loopback] no PDR for destination TEID`

Cause racine : Session PGWU non créée ou désaccord de TEID

Diagnostic :

1. **Vérifiez les sessions PFCP :**

   ```
   curl http://localhost:8080/api/v1/sessions | jq '.sessions[] |
   select(.uplink_teid == 200)'
   ```

2. **Vérifiez la configuration FAR :**

   ```
   curl http://localhost:8080/api/v1/far_map | jq '.[] |
   select(.teid == 200)'
   ```

Solution : Assurez-vous que PGWU-C crée une session avec un TEID correspondant à celui utilisé par SGWU-C pour le transfert N9

Utilisation CPU élevée

Symptôme : Utilisation CPU supérieure à celle attendue

Cause racine : Programme eBPF traitant les paquets plusieurs fois ou recherches de carte excessives

Diagnostic :

```
# Vérifiez les modèles d'accès à la carte eBPF
sudo bpftool map dump name pdr_map_teid_ip4 | wc -l
sudo bpftool map dump name far_map | wc -l
```

Solution :

- Augmentez `max_sessions` si la carte est pleine (provoque des échecs de recherche)
- Vérifiez que la limitation de débit QER ne provoque pas de pertes et de retransmissions
- Vérifiez les tampons de paquets excessifs

Perte de paquets lors du transfert

Symptôme : Paquets perdus lors du transfert de l'eNodeB

Cause racine : Tamponnage non configuré ou limites de tampon insuffisantes

Configuration :

```
buffer_port: 22152
buffer_max_packets: 20000      # Augmenter pour les réseaux à haute mobilité
buffer_max_total: 100000
buffer_packet_ttl: 30          # Ajuster en fonction du temps de transfert
```

Vérification :

```
curl http://localhost:8080/api/v1/upf_buffer_info | jq
```

Avantages du N9 Loopback

Performance

Métrique	Deux Instances	Instance Unique (N9 Loopback)	Amélioration
Latence	1-5 ms	< 1 µs	**1000x plus rapide**
Débit	Limité par le réseau	Limité par le CPU/ mémoire	**2-3x plus élevé**
Utilisation CPU	2× passages XDP + pile réseau	1× passage XDP	**Réduction de 40-50%**
Perte de paquets	Risque pendant la congestion réseau	Zéro (en mémoire)	**Éliminée**

Opérationnel

- **Déploiement simplifié :** Instance OmniUPF unique au lieu de deux
- **Infrastructure réduite :** Moitié des serveurs, ports réseau, adresses IP
- **Moins de complexité :** Configuration unique, point de surveillance unique
- **Économies de coûts :** Réduction du matériel, de l'énergie, du refroidissement, de la maintenance
- **Dépannage plus facile :** Trace de paquets unique, sortie de débogage eBPF unique

Cas d'utilisation

Idéal pour :

- 🔷 **Edge Computing :** Minimiser la latence pour la rupture locale
- 🔷 **Déploiements petits/moyens :** < 100K abonnés
- 🔷 **Laboratoire/Test :** Plan utilisateur EPC complet sur une seule VM
- 🔷 **Contraintes budgétaires :** Budget matériel limité

Non recommandé pour :

- 🔷 **Redondance géographique :** SGWU et PGWU dans différents centres de données
- 🔷 **Échelle massive :** > 1M abonnés (envisager l'évolutivité horizontale)
- 🔷 **Exigences réglementaires :** Séparation obligatoire de SGW et PGW

Comparaison avec d'autres modes de déploiement

Instance unique (N9 Loopback) vs. Instances séparées

Fonctionnalité	N9 Loopback	Séparé	Conteneurs
Latence	⚡ < 1 µs	🔷 1-5 ms	🔷 5-20 ms
Débit	⚡ 40+ Gbps	🔷 20+ Gbps	🔷 10+ Gbps
Infrastructure	🔷 1 serveur	🔷 2 serveurs	⚠ 1 serveur, 2 VMs
Complexité	🔷 Simple	🔷 Complexe	⚠ Modérée
Coût	🔷 Le plus bas	🔷 Le plus élevé	⚠ Moyen
Évolutivité	⚠ Verticale uniquement	🔷 Horizontale	🔷 Horizontale
Redondance	🔷 Point de défaillance unique	🔷 Redondance géographique	⚠ Redondance locale

---

Résumé

N9 Loopback permet un **plan utilisateur EPC 4G de qualité opérateur sur une seule instance OmniUPF** en traitant le trafic SGWU→PGWU entièrement en eBPF sans sauts réseau. Cela fournit :

- 🔷 **Latence sub-microseconde** pour le transfert inter-passarelle
- 🔷 **Réduction de 40-50% de l'utilisation CPU** par rapport aux instances séparées
- 🔷 **Opérations simplifiées** - instance unique, configuration, surveillance
- 🔷 **Coût inférieur** - moitié de l'infrastructure
- 🔷 **Conformité totale 3GPP** - protocoles PFCP, GTP-U standard

La configuration est automatique lorsque `n3_address == n9_address` - aucun drapeau ou paramètre spécial requis. Le chemin de données eBPF d'OmniUPF détecte les conditions de loopback et traite les paquets en ligne.

Pour plus d'informations :

- **Configuration :** CONFIGURATION.md
- **Architecture :** ARCHITECTURE.md
- **Opérations :** OPERATIONS.md
- **Dépannage :** TROUBLESHOOTING.md

Guide de Gestion des Règles

Table des Matières

Aperçu

OmniUPF utilise un ensemble de règles interconnectées pour classifier, transférer, façonner 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 des paquets 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 Paquet)	Classifier 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 QoS)	Appliquer des limites de bande passante et marquage	ID QER	SMF via Établissement/ Modification de Session PFCP
URR (Règle de Rapport 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 paquet dans le UPF.

Structure des PDR

PDR Montants

Les PDR montants 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 QoS (optionnel)
- **IDs URR** : Liste des règles de rapport d'utilisation (optionnel)
- **Suppression d'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 par 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, rejeter le paquet

Exemple :

```
TEID : 5678
ID FAR : 2
ID QER : 1
Suppression d'En-tête Externe : Faux
Mode SDF : Pas de SDF
```

PDR Descendants

Les PDR descendants 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 QoS (optionnel)
- **IDs URR** : Liste des règles de rapport 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 Seulement` : 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 par 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, rejeter le paquet

Exemple :

```
IP UE : 10.45.0.1
ID FAR : 1
ID QER : 1
Suppression d'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 Filtrage :

- **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 Seulement
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 des FAR

Indicateurs d'Action

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

Indicateur	Bit	Valeur	Description
TRANSFERT	1	2	Transférer le paquet à la destination
BUFFER	2	4	Stocker le paquet dans le tampon
REJET	0	1	Rejeter 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 (TRANSFERT)` - Transfert normal (le plus courant)
- `Action : 6 (TRANSFERT + BUFFER)` - Transférer et tamponner pendant la

transition
- `Action : 4 (BUFFER)` - Tamponner uniquement (pendant le changement de chemin)
- `Action : 1 (REJET)` - Rejeter le paquet (rare, généralement pour l'application de politiques)

Contrôle de Tamponnage

L'indicateur BUFFER (bit 2) contrôle le tamponnage des paquets pendant les événements de mobilité.

Opérations de Tamponnage :

- **Activer le Tampon** : Définir le bit 2 de l'action FAR (Action |= 4)
- **Désactiver le Tampon** : Effacer le bit 2 de l'action FAR (Action &= ~4)
- **Vider le Tampon** : Rejouer tous les paquets tamponnés en utilisant les règles FAR actuelles
- **Effacer le Tampon** : Rejeter tous les paquets tamponnés sans transfert

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 du gNB (par exemple, 200.198.5.10)
- TEID : ID de Tunnel pour le trafic UE
- Action : Ajouter l'en-tête GTP-U, transférer au gNB

Recherche de FAR dans l'Interface Web

La page de Gestion des Règles fournit une recherche de FAR par ID :

Étapes :

1. Naviguer vers Règles → Onglet FARs
2. Entrer l'ID FAR dans le champ de recherche
3. Cliquer sur "Rechercher" pour voir les détails du FAR

Informations Affichées :

- ID FAR

- Action (numérique + indicateurs décodés)
- Statut de Tamponnage (ACTIF/DÉSACTIF)
- Création d'En-tête Externe
- Adresse IP distante (avec représentation entière)
- TEID
- Marquage de Niveau de Transport

Règles d'Application de QoS (QER)

Objectif

Les QER appliquent des paramètres de Qualité de Service aux flux de trafic, y compris des limites de bande passante et le marquage des paquets.

Structure des QER

Paramètres de 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 = porteur par défaut)
- Utilisé pour le marquage des paquets dans 5GC

Statut de la Porte :

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

Débit Max (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 rejeté

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 QoS

Règles de Rapport d'Utilisation (URR)

Objectif

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

Structure des URR

Suivi des Volumes

Volume Montant :

- Octets transmis de l'UE au Réseau de Données
- Mesuré après la décapsulation GTP-U
- Inclut l'en-tête IP et la charge utile

Volume Descendant :

- Octets transmis du Réseau de Données à 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 rapport d'utilisation total

Déclencheurs de Rapport 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 Événements :

- Rapport à la terminaison de session
- Rapport sur changement de QoS
- Rapport lors du transfert

Formatage d'Affichage des Volumes

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

Octets	Affichage
0 - 1023	B (Octets)
1024 - 1048575	Ko (Kilooctets)
1048576 - 1073741823	Mo (Mégaoctets)
1073741824 - 1099511627775	Go (Gigaoctets)
1099511627776+	To (Téraoctets)

Exemple :

```
ID URR : 0
Volume Montant : 12.3 Ko
Volume Descendant : 9.0 Ko
Volume Total : 21.3 Ko
```

Flux de Rapport URR

Relations entre Règles

Chaîne PDR → FAR → QER → URR

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

Exemple de Configuration de Session

PDR Montant :

```
TEID : 5678
ID FAR : 2
ID QER : 1
IDs URR : [0]
Suppression d'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
```

ID FAR 1 (Descendant) :

```
Action : 2 (TRANSFERT)
Création d'En-tête Externe : Vrai
IP Distante : 200.198.5.10
TEID : 5678
```

ID FAR 2 (Montant) :

```
Action : 2 (TRANSFERT)
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 Ko
Volume Descendant : 9.0 Ko
Volume Total : 21.3 Ko
```

Opérations Courantes

Voir les Règles pour une Session

Via la Page des Sessions :

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

Via la Page des Règles :

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

Activer/Désactiver le Tamponnage

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

Étapes :

1. Naviguer vers Règles → FARs
2. Entrer l'ID FAR dans le champ de recherche
3. Cliquer sur "Rechercher"
4. Si le tamponnage est DÉSACTIVÉ, cliquer sur "Activer le Tamponnage"
5. Vérifier que le bit 2 de l'action FAR est défini (la valeur de l'action augmente de 4)

Alternative via la Page des Tampons :

1. Naviguer vers Tampons
2. Voir les FARs avec tamponnage activé
3. Cliquer sur "Désactiver le Tampon" lorsque le transfert est terminé

Surveiller la Conformité QoS

Vérifier si le trafic est limité en débit :

1. Naviguer vers Règles → QERs
2. Trouver l'ID QER associé à la session UE
3. Noter les valeurs MBR Montant et MBR Descendant
4. Comparer avec le taux de croissance du volume URR

Calculer le Débit Moyen :

```
Débit (kbps) = (Delta de Volume en octets × 8) / (Delta de Temps en secondes × 1000)
```

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

Suivre l'Utilisation des Données

Surveiller les volumes URR :

1. Naviguer 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 Circulant

Vérifier le PDR :

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

Vérifier le FAR :

1. Vérifier que l'action FAR est TRANSFERT (pas REJET ou BUFFER uniquement)
2. Confirmer que la création d'en-tête externe correspond à la direction
3. Vérifier que l'IP distante et le TEID sont corrects pour le descendant

Vérifier le QER :

1. Vérifier que le Statut de la Porte est Ouvert (0)
2. Vérifier que le MBR n'est pas trop restrictif

Paquets Rejetés

Vérifier la Limitation de Débit QER :

1. Naviguer vers Règles → QERs
2. Vérifier que le MBR est adéquat pour la charge de trafic

3. Vérifier que la croissance du volume URR correspond au débit attendu

Vérifier l'Action FAR :

1. Naviguer vers Règles → FARs
2. Vérifier que l'action est TRANSFERT, pas REJET
3. Vérifier que le tamponnage n'est pas bloqué en mode BUFFER uniquement

Problèmes de Tamponnage

Paquets bloqués dans le tampon :

1. Naviguer vers la page des Tampons
2. Vérifier l'horodatage du plus ancien paquet
3. Si > 30 secondes, le transfert peut avoir échoué
4. Vider ou effacer manuellement le tampon
5. Désactiver le tamponnage sur le FAR

Dépassement de Tampon :

1. Vérifier le nombre total de paquets par rapport au Max Total (par défaut 100 000)
2. Vérifier le nombre de paquets par FAR par rapport au Max Par FAR (par défaut 10 000)
3. Effacer les tampons si pleins
4. Enquêter sur pourquoi le tamponnage n'a pas été désactivé

URR Ne Suit Pas

Compteurs de volume à zéro :

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

Volume ne rapportant pas au SMF :

1. Vérifier la configuration du Rapport de Session PFCP
2. Vérifier les déclencheurs de rapport URR (seuils de volume/temps)
3. Examiner les journaux pour les messages de Rapport de Session PFCP

Documentation Connexe

- **Guide des Opérations UPF** - Aperçu de l'architecture et des composants d'OmniUPF
- **Guide des Opérations PFCP** - Gestion des sessions PFCP et installation des règles

- **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

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 un dépannage approfondi, vérifiez :

```
# 1. Vérifiez qu'OmniUPF fonctionne
systemctl status omniupf  # ou ps aux | grep eupf

# 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 de 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

Liste de 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
```

Liste de 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 GTP-U sur N3 :

```
tcpdump -i eth0 -n udp port 2152 -w /tmp/n3_traffic.pcap
```

Capturez le trafic PFCP sur N4 :

```
tcpdump -i eth0 -n udp port 8805 -w /tmp/pfcp_traffic.pcap
```

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] failed to load eBPF objects: mount bpf filesystem at /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

# Vérifier le montage
mount | grep bpf
```

Problème : "Opération non permise" lors du chargement de eBPF

Symptômes :

```
ERRO[0000] failed to load eBPF program: operation not permitted
```

Cause : Capacités insuffisantes ou limites de mémoire verrouillée

Résolution :

```
# Vérifiez la limite de mémoire verrouillée actuelle
ulimit -l

# Définir la mémoire verrouillée illimitée (nécessaire pour eBPF)
ulimit -l unlimited

# Rendre persistant (ajouter à /etc/security/limits.conf)
echo "* soft memlock unlimited" | sudo tee -a /etc/security/
limits.conf
echo "* hard memlock unlimited" | sudo tee -a /etc/security/
limits.conf

# Exécuter OmniUPF avec les capacités requises
sudo setcap cap_sys_admin,cap_net_admin,cap_bpf+eip /usr/bin/eupf

# Ou exécuter avec sudo
sudo ./eupf
```

Problème : Version du noyau trop ancienne

Symptômes :

```
ERRO[0000] kernel version 5.4.0 is too old, minimum required is
5.15.0
```

Cause : Version du noyau Linux inférieure à l'exigence minimale

Résolution :

```
# Vérifiez la version du noyau
uname -r

# Mettre 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 :

```
error while loading shared libraries: libbpf.so.0: cannot open shared
object file
```

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] unable to read config file: unmarshal errors
```

Cause : Erreur de syntaxe YAML dans le fichier de configuration

Résolution :

```
# Valider la syntaxe YAML
cat config.yml | python3 -c "import yaml, sys;
yaml.safe_load(sys.stdin)"

# Problèmes courants :
# - Indentation incorrecte (utilisez des espaces, pas des
tabulations)
# - Deux-points manquants après les clés
# - Chaînes non citées avec des caractères spéciaux
# - Éléments de liste sans tirets

# Exemple de YAML correct :
cat > config.yml <<EOF
interface_name: [eth0]
xdp_attach_mode: generic
api_address: :8080
pfcp_address: :8805
EOF
```

Problème : Nom d'interface non trouvé

Symptômes :

```
ERRO[0000] interface eth0 not found
```

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 config.yml :
interface_name: [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] failed to start API server: address already in use
```

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 la configuration
api_address: :8081
pfcp_address: :8806
metrics_address: :9091
```

Problème : ID de nœud PFCP invalide

Symptômes :

```
ERRO[0000] invalid pfcp_node_id: must be valid IPv4 address
```

Cause : L'ID de nœud PFCP n'est pas une adresse IPv4 valide

Résolution :

```
# Correct : Utilisez une adresse IP (pas un nom d'hôte)
pfcp_node_id: 10.100.50.241

# Incorrect :
# pfcp_node_id: localhost
# pfcp_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 :

```
# Définir l'ID de nœud PFCP sur l'IP correcte de l'interface N4
pfcp_node_id: 10.100.50.241  # Doit correspondre à l'IP sur le réseau N4
```

Réseau inaccessible au SMF

Résolution :

```
# Tester la connectivité au SMF
ping <SMF_IP>

# Vérifiez le routage vers le SMF
ip route get <SMF_IP>

# Ajoutez une route si manquante
```

```
sudo ip route add <SMF_NETWORK>/24 via <GATEWAY>
```

SMF configuré avec une mauvaise IP de UPF

Résolution :

- Vérifiez la configuration 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 heartbeat PFCP

Symptômes :

```
WARN[0030] PFCP heartbeat timeout for 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 heartbeat
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 heartbeat trop agressif

Résolution :

```
# Augmenter l'intervalle de heartbeat
heartbeat_interval: 30  # Augmenter de 5 à 30 secondes
heartbeat_retries: 5    # Augmenter les tentatives
```

```
heartbeat_timeout: 10   # Augmenter le délai d'attente
```

Probl��mes de Traitement de Paquets

Problème : Aucun paquet ne circule (compteurs RX/TX à 0)

Symptômes :

- La page des statistiques affiche 0 paquets RX/TX
- L'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. Capturez le trafic sur l'interface
tcpdump -i eth0 -n -c 10

# 4. Vérifiez les statistiques de paquets
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 ligne ou pas de lien

Résolution :

```
# Mettez l'interface en ligne
sudo ip link set eth0 up

# Vérifiez l'état du lien
```

```
ethtool eth0 | grep "Link detected"

# Si le lien est hors ligne, vérifiez la connexion physique ou la
configuration réseau de la VM
```

Mauvaise interface configurée

Résolution :

```
# Mettez à jour config.yml avec la bonne interface
interface_name: [ens1f0]  # Utilisez le nom réel de l'interface de
'ip link show'
```

Problème : Paquets reçus mais non transférés (taux de perte élevé)

Symptômes :

- Compteurs RX augmentent mais les compteurs TX ne le sont pas
- 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

Résolution :

```
# Vérifiez les statistiques QER
curl http://localhost:8080/api/v1/sessions | jq '.[].qers'

# Si MBR (Débit Maximum) trop bas, demandez au SMF de mettre à jour
QER
# Ou vérifiez si le trafic dépasse les taux configurés
```

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 de l'interface
curl http://localhost:8080/api/v1/packet_stats | jq
'.interface_stats'

# Capturez le trafic sur les deux interfaces
tcpdump -i eth0 -n udp port 2152 &  # N3
tcpdump -i eth1 -n not udp port 2152 &  # N6
```

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 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 que le 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 instructions 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] failed to load XDP program: invalid argument
```

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 d'aborts XDP en augmentation

Symptômes :

- Les statistiques XDP montrent des aborts > 0
- Augmentation des pertes de paquets

Diagnostic :

```
# Vérifiez le compte d'aborts 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 :

```
# Augmentez la capacité de la carte dans config.yml
max_sessions: 200000  # Augmentez de 100000

# Ou définissez les tailles de carte individuelles
pdr_map_size: 400000
far_map_size: 400000
qer_map_size: 200000
```

Important : Changer les tailles de carte nécessite un redémarrage d'OmniUPF et **efface toutes les sessions existantes**.

Problèmes de Performance

Problème : Faible débit (en dessous des attentes)

Symptômes :

- Débit < 1 Gbps malgré un NIC capable
- Utilisation CPU élevée

Diagnostic :

```
# Vérifiez le taux de paquets
curl http://localhost:8080/api/v1/packet_stats | jq '.total_rx,
.total_tx'

# Surveillez l'utilisation CPU
top -bn1 | grep eupf

# 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 :

```
# Passez au mode natif pour de meilleures performances
xdp_attach_mode: native  # Nécessite un NIC/pilote compatible XDP
```

Goulot 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 :

```
# Réduisez les limites de tampon pour diminuer la latence
buffer_max_packets: 5000
buffer_packet_ttl: 15
```

Problème : Latence élevée

Symptômes :

- Latence de 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 tampon
curl http://localhost:8080/api/v1/upf_buffer_info | jq
'.total_packets_buffered'

# Vérifiez la performance du cache de routage
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'
```

Résolutions :

Paquets étant mis en tampon de manière excessive

Résolution :

```
# Vérifiez pourquoi les paquets sont mis en 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 ring
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, consultez 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 ponté sans SR-IOV

Résolution :

Option 1 : Utiliser le mode générique (le plus simple)

```
xdp_attach_mode: generic
```

Option 2 : Configurer le passage SR-IOV

```
# Sur l'hôte Proxmox :
# 1. Activez IOMMU
nano /etc/default/grub
# Ajoutez : intel_iommu=on iommu=pt
update-grub
reboot

# 2. Créez des VFs
echo 4 > /sys/class/net/eth0/device/sriov_numvfs

# 3. Assignez le VF à la VM dans l'interface Proxmox
# Matériel → Ajouter → Périphérique PCI → Sélectionnez le VF

# Dans la VM :
interface_name: [ens1f0]  # VF SR-IOV
xdp_attach_mode: native
```

VMware : Mode Promiscuous requis

Symptômes :

- Paquets non reçus par OmniUPF

Cause : vSwitch bloquant les adresses MAC non correspondantes

Résolution :

```
# Activez le mode promiscuous sur vSwitch (dans vSphere Client) :
# 1. Sélectionnez 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 :

```
# Utilisez le mode générique (seule option)
xdp_attach_mode: generic

# Optimisez les paramètres de VirtualBox :
# - Utilisez un adaptateur VirtIO-Net (si disponible)
# - Activez le mode promiscuous "Allow All"
# - Allouez plus de cœurs CPU à la VM
# - Utilisez un réseau ponté au lieu de NAT

# Envisagez 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] failed to attach XDP program: operation not supported
```

Diagnostic :

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

# Vérifiez si le pilote prend en charge XDP
```

```
modinfo <driver_name> | grep -i xdp

# Listez les interfaces compatibles XDP
ip link show | grep -B 1 "xdpgeneric\|xdpdrv\|xdpoffload"
```

Résolution :

Option 1 : Utiliser le mode générique

```
xdp_attach_mode: generic
```

Option 2 : Mettre à jour le pilote NIC

```
# Vérifiez les mises à jour de pilote (Ubuntu)
sudo apt update
sudo apt install linux-modules-extra-$(uname -r)

# Ou installez le pilote spécifique au fournisseur
# Exemple pour Intel :
# Téléchargez depuis https://downloadcenter.intel.com/
```

Option 3 : Remplacer le NIC

```
# Utilisez un NIC compatible 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 :

```
# Vérifiez les journaux du noyau
dmesg | tail -100

# Vérifiez les bugs de pilote
journalctl -k | grep -E "BUG:|panic:"
```

Résolution :

```
# 1. Mettez à jour le noyau et les pilotes
sudo apt update
sudo apt upgrade
```

```
sudo reboot

# 2. Désactivez le déchargement XDP (utilisez uniquement natif)
xdp_attach_mode: native

# 3. Utilisez le mode générique comme solution de contournement
xdp_attach_mode: generic

# 4. Signalez le bug au fournisseur de NIC ou à l'équipe du noyau
Linux
```

Échecs d'Établissement de Session

Problème : Échec de l'établissement de session

Symptômes :

- Le SMF signale un échec de l'établissement de session
- L'UE ne peut pas établir de session PDU

Diagnostic :

```
# Vérifiez les journaux OmniUPF pour les erreurs de session
journalctl -u omniupf | grep -i "session establishment"

# Vérifiez le compte de sessions PFCP
curl http://localhost:8080/api/v1/sessions | jq 'length'

# Capturez 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 :

```
# Vérifiez l'utilisation de la carte
curl http://localhost:8080/api/v1/map_info | jq '.[] |
select(.usage_percent > 90)'

# Augmentez la capacité (voir la section carte eBPF pleine ci-dessus)
```

Paramètres PDR/FAR invalides

Résolution :

```
# Vérifiez 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

# Vérifiez la configuration SMF et les paramètres de session
```

Fonction non supportée (UEIP/FTUP)

Résolution :

```
# Activez les fonctionnalités requises si nécessaire
feature_ueip: true  # Allocation IP UE par UPF
ueip_pool: 10.60.0.0/16

feature_ftup: true  # Allocation F-TEID par UPF
teid_pool: 100000
```

Problèmes de Mise en Tampon

Problème : Paquets bloqués dans le tampon

Symptômes :

- Compte de paquets mis en tampon en augmentation
- Paquets non livrés après un transfert

Diagnostic :

```
# Vérifiez les statistiques de tampon
curl http://localhost:8080/api/v1/upf_buffer_info

# Vérifiez les tampons individuels FAR
curl http://localhost:8080/api/v1/upf_buffer_info | jq '.buffers[] |
{far_id, packet_count, oldest_packet_ms}'

# Surveillez la taille du tampon
watch -n 5 'curl -s http://localhost:8080/api/v1/upf_buffer_info | jq
".total_packets_buffered"'
```

Causes et Résolutions :

FAR jamais mis à jour vers FORWARD

Cause : SMF n'a jamais envoyé de modification de session PFCP pour appliquer le FAR

Résolution :

```
# Vérifiez 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 (transfert)

# Si bloqué dans l'état BUFF, demandez au SMF de :
# - Envoyer une demande de modification de session PFCP
# - Mettre à jour le FAR avec l'action FORW
```

TTL de tampon expiré

Cause : Paquets expirés avant la mise à jour du FAR

Résolution :

```
# Augmentez le TTL de tampon
buffer_packet_ttl: 60  # Augmentez de 30 à 60 secondes
```

Débordement de tampon

Cause : Trop de paquets mis en tampon par FAR

Résolution :

```
# Augmentez les limites de tampon
buffer_max_packets: 20000  # Par FAR
buffer_max_total: 200000   # Limite globale
```

Débogage Avancé

Activer la Journalisation de Débogage

```
logging_level: debug  # trace | debug | info | warn | error
```

```
# Redémarrez OmniUPF avec la journalisation de débogage
sudo systemctl restart omniupf
```

```
# Surveillez les journaux en temps réel
journalctl -u omniupf -f --output cat
```

Traçage du Programme eBPF

```
# Tracez l'exécution du programme eBPF (nécessite bpftrace)
sudo bpftrace -e 'tracepoint:xdp:* { @[probe] = count(); }'

# Tracez les opérations de carte
sudo bpftrace -e 'tracepoint:bpf:bpf_map_lookup_elem { printf("%s\n",
str(args->map_name)); }'
```

Capture de Paquets au Niveau XDP

```
# Capturez les paquets avant XDP (tcpdump)
tcpdump -i eth0 -w /tmp/before_xdp.pcap

# Capturez les paquets après XDP (nécessite XDP_PASS)
tcpdump -i any -w /tmp/after_xdp.pcap

# Comparez les comptes de paquets pour identifier les pertes
```

Obtenir de l'Aide

Si les étapes de dépannage ne résolvent pas votre problème :

1. **Collectez 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. **Signalez le problème** avec :
    - Version d'OmniUPF
    - Version du noyau Linux
    - Diagramme topologique du réseau
    - Fichier de configuration (cacher les informations sensibles)
    - Extraits de journaux pertinents
    - Étapes pour reproduire

3. **Support communautaire** :
    - Problèmes GitHub : https://github.com/edgecomllc/eupf/issues
    - Documentation : Voir les guides connexes ci-dessous

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
- **Guide de Gestion des Règles** - Concepts PDR, FAR, QER, URR
- **Guide d'Opérations** - Architecture et aperçu de l'UPF

Guide des opérations de l'interface Web

Table des matières

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
- **Visionnage 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

Accéder 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'hôtes 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 et surveillance de la capacité des cartes eBPF
- **Configuration** - `/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 de 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	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 des 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 complets des PDR/FAR/QER/URR en JSON
- **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
- **Règles d'action de transfert (FARs)** : Drapeaux d'action, création d'en-tête externe, 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)

Cas d'utilisation

Vérifier la connectivité UE :

1. Accédez à la vue des sessions
2. Entrez l'adresse IP UE dans le filtre
3. Confirmez que la session existe avec le TEID correct
4. Étendez pour vérifier la configuration PDR/FAR

Surveiller le nombre de sessions :

- Vérifiez le nombre total de sessions dans l'en-tête
- Comparez à travers plusieurs instances UPF
- Suivez la croissance des sessions au fil du temps

Dépanner les problèmes de session :

- Recherchez une IP UE ou un TEID spécifique
- Étendez la session pour inspecter la configuration des règles
- Vérifiez les paramètres de transfert FAR
- Vérifiez 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 :

- **SANTÉ** (vert) : UPF est accessible et répond
- **NON SANTÉ** (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, 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 :

PDRs de montée (N3 → N6) :

- **TEID** : ID de point de terminaison de tunnel GTP-U provenant de gNB
- **ID FAR** : Règle d'action de transfert associée
- **ID QER** : Règle d'application de QoS associée (le cas échéant)
- **Suppression d'en-tête externe** : Drapeau de décapsulation GTP-U

PDRs de descente (N6 → N3) :

- **IP UE** : Adresse IPv4 ou IPv6 de l'équipement utilisateur
- **ID FAR** : Règle d'action de transfert associée
- **ID QER** : Règle d'application de QoS associée (le cas échéant)
- **Mode SDF** : Mode de filtre de flux de données de service (aucun, sdf seulement, sdf + par défaut)

PDRs IPv6 :

- Tables séparées pour les PDRs de montée et de descente IPv6
- Même structure que pour IPv4 mais indexée par adresses IPv6

Onglet FAR - Règles d'action de transfert

Voir tous les FARs avec leurs actions de transfert et paramètres :

Colonne	Description
ID FAR	Identifiant unique de règle de transfert
Action	Drapeaux d'action de transfert (FORWARD, DROP, BUFFER, DUPLICATE, NOTIFY)
Tamponnage	État actuel du tamponnage (Activé/Désactivé)
Destination	Paramètres de création d'en-tête externe (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 :

Colonne	Description
ID QER	Identifiant unique de règle QoS
MBR (Montée)	Débit maximal pour le trafic montant (kbps)
MBR (Descente)	Débit maximal pour le trafic descendant (kbps)
GBR (Montée)	Débit garanti pour le trafic montant (kbps)
GBR (Descente)	Débit garanti pour le trafic descendant (kbps)
QFI	Identifiant de flux QoS (marquage 5G)

Interprétation 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 :

Colonne	Description
ID URR	Identifiant unique de règle de rapport d'utilisation
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
Méthode	Méthode de rapport (volume, temps, événement)

Affichage du volume :

- Formaté automatiquement (B, KB, MB, GB, TB)
- 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
- Limité aux 1000 URRs les plus actifs pour des performances optimales

Cas d'utilisation

Inspecter la classification du trafic :

1. Accédez à Règles → onglet PDR
2. Recherchez un TEID ou une IP UE spécifique
3. Vérifiez que le PDR est associé au bon FAR et QER

Dépanner les problèmes de transfert :

1. Accédez à Règles → onglet FAR
2. Localisez l'ID FAR à partir du PDR de la session
3. Vérifiez que l'action est FORWARD (pas DROP ou BUFFER)
4. Vérifiez les paramètres de création d'en-tête externe

Surveiller l'application de QoS :

1. Accédez à Règles → onglet QER
2. Vérifiez que les valeurs MBR et GBR correspondent à la politique
3. Vérifiez le marquage QFI pour les flux 5G

Suivre l'utilisation des données :

1. Accédez à Règles → onglet URR
2. Triez par volume total pour trouver les utilisateurs les plus élevés
3. Surveillez la croissance du volume au fil du temps
4. Vérifiez 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 sur les tampons :

- **Total des paquets** : Nombre de paquets tamponnés à travers tous les

FARs
- **Total des octets** : Taille totale des données tamponnées
- **Total des FARs** : Nombre de FARs avec des paquets tamponnés
- **Max par FAR** : Maximum de paquets autorisés par FAR
- **Max total** : Maximum total de paquets tamponnés
- **TTL des paquets** : Durée de vie des paquets tamponnés (secondes)

Tampons par FAR

Tableau de tous les FARs avec des paquets tamponnés :

Colonne	Description
ID FAR	Identifiant de règle d'action de transfert
Nombre de paquets	Nombre de paquets tamponnés pour ce FAR
Nombre d'octets	Total d'octets tamponnés pour ce FAR
Paquet le plus ancien	Horodatage du plus ancien paquet tamponné
Paquet le plus récent	Horodatage du plus récent paquet tamponné
Actions	Boutons de contrôle des tampons (style pilule)

Actions de contrôle des tampons

Pour chaque FAR avec des paquets tamponnés, 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 tamponnés en utilisant les règles FAR actuelles
- **Effacer** (gris) : Supprimer tous les paquets tamponnés 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, vérifiez que les paquets sont tamponnés

2. Vérifiez l'état du tamponnage du FAR (doit être activé)
3. Surveillez le nombre de paquets et leur ancienneté

Compléter le transfert :

1. Après le changement de chemin, cliquez sur "Vider" pour rejouer les paquets tamponnés
2. Vérifiez que les paquets sont transférés vers le nouveau chemin
3. Cliquez sur "Désactiver le tampon" pour arrêter le tamponnage

Effacer les tampons bloqués :

1. Identifiez les FARs avec des paquets tamponnés anciens (vérifiez l'horodatage le plus ancien)
2. Cliquez sur "Effacer" pour jeter les paquets obsolètes
3. Ou cliquez sur "Désactiver le tampon" pour empêcher un tamponnage supplémentaire

Dépanner un débordement de tampon :

1. Vérifiez le nombre total de paquets par rapport au maximum total
2. Identifiez les FARs avec un tamponnage excessif
3. Vérifiez que le SMF a envoyé une modification de session pour désactiver le tamponnage
4. Désactivez 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 chemin de données OmniUPF.

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 jetés 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 disponible) :

- **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 traité** : Total des paquets traités au niveau XDP
- **XDP passé** : Paquets envoyés à la pile réseau
- **XDP perdu** : Paquets perdus au niveau XDP
- **XDP avorté** : 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 réseau de données) :

- **RX N6** : Paquets reçus du réseau de données (Internet/IMS)
- **TX N6** : Paquets transmis au réseau de données

Total : Compte agrégé des paquets à travers les interfaces

Utilisation : Surveiller l'équilibre du trafic et les problèmes spécifiques aux interfaces

Cas d'utilisation

Surveiller la charge de trafic :

1. Vérifiez les taux de paquets RX/TX
2. Vérifiez que le trafic circule dans les deux directions
3. Comparez le trafic N3 par rapport au trafic N6 (devrait être à peu près égal)

Détecter les pertes de paquets :

1. Vérifiez le compteur de paquets perdus
2. Consultez le compteur de paquets perdus XDP
3. Enquêtez sur la cause dans les journaux si les pertes sont élevées

Analyse de performance :

1. Surveillez le ratio des paquets traités par rapport aux paquets passés
2. Vérifiez les avortements XDP (indique des erreurs)
3. Vérifiez la distribution du trafic des interfaces N3/N6

Planification de capacité :

1. Suivez le taux de paquets au fil du temps
2. Comparez aux limites de capacité du UPF
3. Planifiez une montée en charge si vous approchez des limites

Mises à jour en temps réel

Les statistiques se rafraîchissent automatiquement toutes les 10 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 clé	Taille des clés de la carte en octets
Taille de 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, surveillez la croissance
- **Amber (70-90%)** : Avertissement, planifiez une augmentation de capacité
- **Rouge (>90%)** : Critique, action immédiate requise

Cartes critiques à surveiller

uplink_pdr_map :

- Stocke les PDRs de montée 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 de descente 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
- **Basse priorité** : Affecte la facturation mais pas la connectivité

Cas d'utilisation

Planification de capacité :

1. Surveillez les tendances d'utilisation des cartes au fil du temps
2. Identifiez quelles cartes croissent le plus rapidement
3. Planifiez des augmentations de capacité avant d'atteindre les limites

Prévenir les échecs d'établissement de session :

1. Vérifiez l'utilisation de la carte PDR avant une augmentation de trafic prévue
2. Augmentez la capacité de la carte si vous approchez des limites
3. Surveillez après l'augmentation de capacité pour vérifier

Dépanner les échecs de session :

1. Lorsque l'établissement de session échoue, vérifiez la vue de capacité
2. Si les cartes PDR sont rouges (>90%), la capacité est épuisée
3. Augmentez la capacité de la carte ou effacez les sessions obsolètes

Optimiser la configuration de la carte :

1. Examinez les tailles de clé et de valeur
2. Calculez l'utilisation de la mémoire par carte
3. Optimisez 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 réseau de données
- **Interface N9** : Adresse IP pour la communication inter-UPF (facultatif)
- **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 à l'exécution :

- **Adresse N3 active** : Liaison de l'interface N3 à l'exécution
- **Adresse N9 active** : Liaison de l'interface N9 à l'exécution (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. Vérifiez que l'adresse IP de l'interface N3 correspond à la configuration gNB
2. Vérifiez que l'interface N6 peut acheminer vers le réseau de données
3. Confirmez que l'interface PFCP est accessible depuis le SMF

Dépanner les problèmes d'interface :

1. Comparez la configuration statique avec les adresses actives du plan de données

2. Vérifiez que les interfaces sont correctement liées
3. Vérifiez les modifications de configuration des interfaces

Documentation et audit :

1. Enregistrez la configuration UPF pour la documentation
2. Vérifiez que le déploiement correspond aux spécifications de conception
3. Auditez 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 sur les 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 vers les 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 (Complet, Init, etc.)
Priorité	Valeur de priorité OSPF
Temps de fonctionnement	Durée pendant laquelle le voisin est actif
Temps mort	Temps jusqu'à ce que le voisin soit considéré comme mort

États OSPF :

- **Complet** (vert) : Complètement adjacent et échangeant des informations de routage
- **Autres états** (jaune) : Formation d'adjacence 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 la session BGP (Établi, Inactif, etc.)
Actif/Inactif	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 (typiquement l'adresse réseau)
Masque	Masque réseau pour la route

Colonne	Description
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 l'origine de la LSA (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 changements de configuration ou pour récupérer des échecs de synchronisation

Bouton Rafraîchir :

- Rafraîchit 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. Accédez à la vue des routes
2. Vérifiez les états des voisins OSPF (doivent être "Complet")
3. Vérifiez que les pairs BGP sont "Établis"
4. Confirmez le nombre attendu de voisins/pairs

Vérifier la distribution des routes UE :

1. Vérifiez le tableau des routes IP UE actives pour une UE spécifique
2. Faites défiler jusqu'à la section des routes redistribuées OSPF
3. Vérifiez que la route UE apparaît dans les LSA externes
4. Confirmez que le routeur annonçant correspond à l'UPF attendu

Dépanner les probl��mes de synchronisation des routes :

1. Vérifiez les compteurs Synchronisés vs. Échoués dans l'aperçu de l'état
2. Si les routes échouent, cliquez sur le bouton "Synchroniser les routes"

3. Surveillez les messages d'erreur dans la bannière rouge si la synchronisation échoue
4. Vérifiez les messages d'erreur OSPF/BGP dans les sections respectives

Vérifier le déploiement multi-UPF :

1. Sélectionnez différentes instances UPF dans le menu déroulant
2. Comparez les comptes de routes entre les instances
3. Vérifiez que les voisins OSPF se voient
4. Vérifiez les relations de pairage BGP

Surveiller l'évolutivité des routes :

1. Suivez le nombre total de routes à mesure que les sessions UE augmentent
2. Vérifiez que les routes sont distribuées aux protocoles de routage
3. Surveillez la croissance du compte LSA OSPF
4. Vérifiez 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 des routes UE.

Intégration de routage

La vue des routes s'intègre à FRR (Free Range Routing) fonctionnant 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 du mode eXpress Data Path (XDP), les capacités de performance et les calculs de débit pour le chemin de données UPF.

Configuration de l'interface

Affiche les informations sur l'interface réseau et le pilote :

Champ	Description
Nom de l'interface	Interface réseau utilisée pour XDP (par exemple,

Champ	Description
	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 un 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 natif XDP dans le pilote, performance maximale
- **Nécessite** : Pilote NIC avec support natif XDP (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** : N'importe quelle 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 mode non supporté :

- 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 Mpps interactif

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"
- Fonctionne en mode générique alors que le mode pilote est disponible
- "⚠ Avertissement : XDP_DRV non supporté par ce pilote"
- Les 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 rapides :

- **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 du fil 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 la 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 inclut une section d'explication couvrant :

Qu'est-ce que Mpps :

- Millions de paquets par seconde
- Mesure clé pour la performance de traitement des paquets
- Indépendant de la taille des paquets

Relation avec le débit :

- Même Mpps avec des paquets plus grands = plus de Gbps
- Même Mpps avec des paquets plus petits = moins de Gbps
- 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. Accédez à la vue des capacités XDP
2. Vérifiez le mode XDP actuel (doit être DRV pour la meilleure performance)
3. Notez la plage de performance Mpps
4. Consultez la bannière de recommandation

Calculer le débit attendu :

1. Entrez le taux de paquets attendu en Mpps
2. Entrez la taille de paquet moyenne pour votre profil de trafic
3. Consultez le débit calculé en Gbps
4. Comparez à la capacité du lien ou aux exigences de performance

Optimiser la configuration XDP :

1. Vérifiez si le mode XDP_DRV est supporté mais non actif
2. Vérifiez la version du pilote et la compatibilité
3. Suivez la recommandation pour passer au mode pilote si disponible
4. Vérifiez que le nombre de files d'attente multiples correspond aux cœurs CPU

Planification de capacité :

1. Utilisez le calculateur pour déterminer le Mpps requis pour le débit cible
2. Comparez aux capacités du mode XDP actuel
3. Déterminez si une mise à niveau matérielle est nécessaire
4. Planifiez la sélection d'interface et de pilote pour les nouveaux déploiements

Dépanner les problèmes de performance :

1. Vérifiez que le mode XDP est DRV, pas SKB
2. Vérifiez la version du pilote pour des problèmes de performance connus
3. Vérifiez que le nombre de files d'attente multiples est suffisant
4. Calculez si le mode actuel prend en charge le débit requis

Conseils d'optimisation de performance

Mode pilote (XDP_DRV) :

- Utilisez des NIC avec support natif XDP (Intel i40e/ixgbe, Mellanox mlx5)
- Mettez à jour les pilotes NIC à la dernière version
- Activez les files d'attente multiples (RSS) pour un traitement parallèle
- Ajustez les tailles de tampon de la NIC

Mode générique (XDP_SKB) :

- Acceptable pour le développement et les tests
- Pas recommandé pour la production à haut débit
- Envisagez une mise à niveau matérielle pour les déploiements de production

Configuration des 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 pendant les sessions actives

Niveaux de journalisation

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. Ouvrez la vue des journaux
2. Initiez l'établissement de session depuis le SMF
3. Surveillez les journaux de messages PFCP et toute erreur

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 cœur

Déboguer les problèmes de transfert :

1. Rechercher des erreurs de traitement de paquets
2. Vérifiez les journaux d'opération de la carte eBPF
3. Identifier les problèmes de configuration FAR/PDR

Meilleures pratiques

Directives opérationnelles

Surveillance :

- Vérifiez régulièrement la vue de capacité pour prévenir l'épuisement des cartes
- Surveillez les statistiques pour des modèles de trafic ou des pertes inhabituels
- Suivez la croissance du nombre de sessions au fil du temps
- Surveillez les erreurs de traitement XDP

Gestion des tampons :

- Surveillez les tampons lors des scénarios de transfert
- Effacez les tampons bloqués si les paquets dépassent le TTL
- Vérifiez que le tamponnage est désactivé après la fin du transfert
- Utilisez "Vider" au lieu de "Effacer" pour éviter la perte de paquets

Gestion des sessions :

- Utilisez des filtres pour localiser rapidement des sessions UE spécifiques
- Étendez les sessions pour vérifier la configuration des règles
- Comparez les sessions à travers plusieurs instances UPF
- Vérifiez l'indicateur de santé avant de dépanner

Dépannage :

- Utilisez les journaux pour le débogage en temps réel
- Vérifiez la vue des sessions pour vérifier la connectivité UE
- Vérifiez la configuration des règles pour les flux de trafic

- Surveillez les statistiques pour des pertes de paquets ou des erreurs de transfert

Performance

- L'auto-rafraîchissement 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 des opérations PFCP** - Gestion des sessions PFCP et détails du protocole
- **Guide de gestion des règles** - Configuration PDR, FAR, QER, URR
- **Guide de surveillance** - Statistiques, métriques et planification de capacité
- **Guide des routes** - Détails sur le routage UE et l'intégration FRR
- **Guide des modes XDP** - Documentation détaillée sur le mode 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

Aperçu

OmniUPF utilise **XDP (eXpress Data Path)** pour le traitement de 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 qualité production.

Comparaison des modes XDP

Aspect	Mode générique	Mode natif	Mode de déchargement
Point de connexion	Pile réseau Linux	Driver 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

Aspect	Mode générique	Mode natif	Mode de déchargement
Exigences NIC	Toute NIC	Driver compatible XDP	SmartNIC avec support XDP
Support des hyperviseurs	Tous les hyperviseurs	La plupart (nécessite multi-queue)	Rare (PCI passthrough)
Cas d'utilisation	Test, développement	**Production (recommandé)**	Sites périphériques à haut débit
Configuration	`xdp_attach_mode: generic`	`xdp_attach_mode: native`	`xdp_attach_mode: 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 driver 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

```
# config.yaml
interface_name: [eth0]
xdp_attach_mode: generic  # Mode par défaut
```

Avertissement : Le mode générique est **non adapté à la production**. Il sera un goulot d'étranglement à des taux de paquets élevés et gaspillera des ressources CPU.

Mode natif (recommandé pour la production)

Description

XDP natif exécute le programme eBPF **à l'intérieur du driver 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 driver)
- **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 driver NIC

XDP natif nécessite un driver réseau avec support XDP. La plupart des NIC modernes supportent 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` (PCI passthrough) ✓

Remarque : VirtualBox ne prend **pas** en charge XDP natif (utilisez uniquement le mode générique).

Configuration

```yaml
# config.yaml
interface_name: [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 périphériques** 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 déploiement **bare metal** ou **PCI passthrough** - non disponible dans les configurations VM standard.

Configuration

```yaml
# config.yaml
interface_name: [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 driver `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 Proxmox

1. **Éteindre complètement la VM** (pas seulement redémarrer)
    - Sélectionnez votre VM dans l'interface web 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"**
    - Définissez sur **8** (ou correspondant 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 les performances élevées (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 afficher 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/eupf/config.yaml
```

Changer le mode XDP :

```
# Avant
xdp_attach_mode: generic

# Après
xdp_attach_mode: native
```

Redémarrer OmniUPF :

```
sudo systemctl restart eupf
```

Étape 5 : Vérifier que XDP natif est actif

Vérifiez les journaux :

```
# Voir les journaux de démarrage
journalctl -u eupf --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 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 driver VirtIO est chargé : `lsmod | grep virtio_net`

Problème : Seulement 1 file d'attente malgré la configuration

Solution :

- La VM doit être **entièrement 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 doit 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 driver `vmxnet3` qui prend en charge XDP natif.

Exigences :

- ESXi 6.7 ou version ultérieure
- Version du driver 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 **Scaling de réception** 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 :

```
interface_name: [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 driver `hv_netvsc` qui prend en charge XDP natif.

Exigences :

- Windows Server 2016 ou version ultérieure
- Services d'intégration Linux 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 d'Hyper-V
Set-VMNetworkAdapter -VMName "YourVM" -VrssEnabled $true -VmmqEnabled
$true
```

Configurer OmniUPF :

```
interface_name: [eth0]
xdp_attach_mode: native
```

VirtualBox

Avertissement : VirtualBox ne prend **pas** en charge XDP natif.

Raison : Les drivers 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. Vérifiez les journaux d'OmniUPF

```
# Voir les journaux récents
journalctl -u eupf --since "5 minutes ago" | grep -i xdp

# Recherchez :
# ✓ "xdp_attach_mode:native"
# ✓ "Programme XDP attaché à l'interface"
# ✗ "Échec de l'attachement" ou "retour au générique"
```

2. Vérifiez via l'API

```
# Interroger le point de configuration
curl -s http://localhost:8080/api/v1/config | jq .xdp_attach_mode

# Sortie attendue :
# "native"
```

3. Vérifiez 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 abandonnés
  "xdp_pass": 5678,      # Passé à la pile
  "xdp_redirect": 9012,  # Paquets redirigés
  "xdp_tx": 3456         # Paquets transmis
}
```

4. Vérifiez le support du driver

```
# Vérifiez si le driver prend en charge XDP
ethtool -i eth0 | grep driver

# Pour Proxmox/KVM : Doit afficher "virtio_net"
```

```
# Pour VMware : Doit afficher "vmxnet3"
# Pour Hyper-V : Doit afficher "hv_netvsc"
```

5. Test de performance

Comparez le traitement des paquets avant et après :

```
# Surveillez 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 driver** :

   ```
   ethtool -i eth0 | grep driver

   # Si le driver 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 driver virtio_net est chargé : `modprobe virtio_net`
- Retournez au mode générique si le driver 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 driver :

```
dmesg | grep -i xdp | tail -20
```

Causes courantes :

1. **Le driver ne prend pas en charge XDP natif** :
    - Drivers de VirtualBox (pas de support XDP natif)
    - Drivers de NIC plus anciens
2. **Multi-queue non activé** :
    - Vérifiez : `ethtool -l eth0`
    - Doit afficher > 1 file d'attente combinée
3. **Support XDP du noyau désactivé** :

```
# Vérifiez si XDP est activé dans le noyau
grep XDP /boot/config-$(uname -r)

# Doit afficher :
# CONFIG_XDP_SOCKETS=y
# CONFIG_BPF=y
```

Solution :

- Activez le multi-queue (voir section Proxmox)
- Mettez à jour vers un driver 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 abandonné (xdp_aborted > 0)

Symptômes :

```
curl http://localhost:8080/api/v1/xdp_stats
{
  "xdp_aborted": 1234,  # Non zéro indique des erreurs
  ...
}
```

Diagnostic :

XDP abandonné 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` affiche seulement 1 file d'attente malgré la configuration

Diagnostic :

1. **Vérifiez la config de la VM Proxmox** :

   ```
   # Sur l'hôte Proxmox
   grep net0 /etc/pve/qemu-server/YOUR_VM_ID.conf

   # Doit afficher : queues=8
   ```

2. **Vérifiez que la VM était complètement arrêtée** :

   ```
   # Sur l'hôte Proxmox
   qm status YOUR_VM_ID

   # Doit afficher "status: stopped" avant de démarrer
   ```

Solution :

```
# Sur l'hôte Proxmox
# Forcer l'arrêt et redémarrer
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 au nombre de files d'attente nécessitent un **arrêt complet de la VM**, pas seulement 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 eupf
   ```

2. **Si vous utilisez systemd**, vérifiez que le fichier de service a des capacités :

   ```
   # /lib/systemd/system/eupf.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 les performances

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 **multi-queue activé** pour les déploiements en production.

Recommandations matérielles pour XDP

⚠ IMPORTANT : Avant d'acheter du matériel, consultez le support d'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	Driver	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 les besoins de bande passante inférieure

NICs Mellanox/NVIDIA (hautes performances)

Modèle	Vitesse	Driver	Support XDP	Remarques
ConnectX-4	25/50/100GbE	mlx5	Natif ✓	Haut débit, bon pour le calcul en périphérie
ConnectX-5	25/50/100GbE	mlx5	Natif ✓	Excellentes performances, accélération matérielle
ConnectX-6	50/100/200GbE	mlx5	Natif ✓	Dernière génération, meilleure pour l'ultra-haut débit
BlueField-2	100/200GbE	mlx5	Natif ✓	SmartNIC avec capacités DPU

NICs Broadcom

Modèle	Vitesse	Driver	Support XDP	Remarques
Série BCM57xxx	10/25/50GbE	bnxt_en	Natif ✓	Commun dans les serveurs Dell/HP

NICs virtuels (déploiements VM)

Plateforme	Type de NIC	Driver	Support XDP	Multi-Queue	Remarques
Proxmox/ KVM	VirtIO	virtio_net	Natif ✓	Oui (configurable)	**Meilleur pour les VM**
VMware ESXi	vmxnet3	vmxnet3	Natif ✓	Oui	Nécessite ESXi 6.7+
Hyper-V	NIC synthétique	hv_netvsc	Natif ✓	Oui	Windows Server 2016+
AWS	ENA	ena	Natif ✓	Oui	Instances metal EC2
VirtualBox	N'importe quel	divers	Seulement générique ⚠	Non	Pas recommandé pour la production

NICs avec support de déchargement matériel

Déchargement XDP véritable (eBPF s'exécute sur NIC) :

Fournisseur	Modèle	Vitesse	Remarques
Netronome	Agilio CX 10G	10GbE	Seul support confirmé de déchargement XDP
Netronome	Agilio CX 25G	25GbE	Nécessite un firmware spécial
Netronome	Agilio CX 40G	40GbE	Très coûteux (~2 500-5 000 $)
Netronome	Agilio CX 100G	100GbE	Niveau entreprise uniquement

Remarque : Les NICs de déchargement matériel sont rares, coûteux et nécessitent un déploiement bare metal. La plupart des déploiements devraient plutôt utiliser XDP natif.

Configurations testées

Ces configurations ont été vérifiées avec OmniUPF en production :

Option économique (1-10 Gbps)

- **NIC** : Intel X520 (10GbE double-port)
- **Mode** : XDP natif
- **Débit** : ~8-10 Gbps par instance UPF
- **Coût** : ~$100-200 (d'occasion/rénové)

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** : Aucun coût matériel supplémentaire

Ce qu'il ne faut PAS acheter

Évitez ceci pour les déploiements OmniUPF en production :

NIC/Plateforme	Raison	Alternative
NICs Realtek	Pas de support XDP, mauvais drivers Linux	Intel i350 ou mieux
VirtualBox	Pas de support XDP natif	Migrer vers Proxmox/KVM
NICs de consommation	Prise en charge limitée des files d'attente, peu fiables	NICs Intel/Mellanox de niveau serveur
NICs très anciennes (<2014)	Pas de support de driver XDP	Intel X520 ou plus récent

Liste de vérification avant achat

Avant d'acheter du matériel, vérifiez :

1. 🔍 **Support du driver** : Vérifiez si le driver Linux prend en charge XDP

   ```
   # Sur un système similaire
   modinfo <driver_name> | 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 la 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 requise
    - VM : Support VirtIO ou SR-IOV nécessaire
    - Conteneur : Configuration de l'hôte NIC héritée

⚠ N'achetez pas de matériel uniquement sur la base de ce guide - confirmez toujours d'abord avec le support d'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 de l'architecture eBPF et XDP
- **Guide de surveillance** : MONITORING.md - Surveillance des performances et statistiques

---

Référence rapide

Configuration XDP natif sur 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/eupf/config.yaml  # Définir : xdp_attach_mode: native
sudo systemctl restart eupf
journalctl -u eupf --since "1 min ago" | grep xdp  # Vérifiez le mode natif
```

Vérifiez 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

Aperçu

L'API OmniUPF fournit une interface RESTful pour gérer et surveiller la Fonction de Plan Utilisateur basée sur eBPF. L'API permet un contrôle et une observabilité en temps réel de :

- **Sessions PFCP** : Gestion du cycle de vie des sessions et des associations
- **Règles de Détection de Paquets (PDR)** : Classification du trafic pour l'ascendant et le descendant (IPv4 et IPv6)
- **Règles d'Action de Transfert (FAR)** : Actions de transfert, de mise en tampon et de suppression des paquets
- **Règles d'Application de QoS (QER)** : Politiques de qualité de service et limitation de débit
- **Règles de Rapport d'Utilisation (URR)** : Suivi et rapport du volume de données
- **Tampons de Paquets** : Fonctionnalité de mise en tampon et de relecture des paquets
- **Statistiques** : Métriques en temps réel pour les paquets, les routes, XDP et les interfaces N3/N6
- **Gestion des Routes** : Synchronisation des routes UE avec le démon de routage FRR
- **Configuration** : Gestion de la configuration du UPF et du plan de données

Documentation de l'API Swagger

L'API est entièrement documentée en utilisant la spécification **OpenAPI 3.0 (Swagger)**. L'interface interactive Swagger UI 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 interactive Swagger UI montrant les points de terminaison de l'API OmniUPF avec une documentation détaillée.

Accéder à Swagger UI

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
```

```
## Voir Aussi

- [Documentation de la Gestion des Routes UE](./routes.md) - Guide
détaillé sur l'intégration FRR et la synchronisation des routes
- [Guide des Opérations](../OPERATIONS.md) - Opérations de
l'interface Web et surveillance
- [Swagger UI](http://10.98.0.20:8080/swagger/index.html) -
Documentation interactive de l'API
```

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 actuelle du réseau.

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. Il 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. Lorsqu'elle est activée, 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 en utilisant des **modèles Ansible** pour établir les paramètres de routage de base. Vous définissez la configuration FRR une fois comme un **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 kernel
 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** : Créez le modèle Jinja2 FRR dans votre rôle Ansible (par exemple, `roles/frr/templates/frr.conf.j2`)
2. **Configurer les Paramètres** : Définissez des variables dans votre inventaire Ansible pour chaque hôte UPF

3. **Déployer Partout** : Exécutez 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 du noyau redistribuer (par exemple, `redistribute kernel`)
- **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** à la table de routage du noyau en fonction des sessions PFCP actives. Ces routes sont ensuite :

1. **Installées dans la table de routage du noyau** par le moteur de synchronisation des routes UPF
2. **Récupérées par FRR** via la directive `redistribute kernel`
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

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 config 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 /32 des UE en fonction de ses sessions PFCP actives
- **Annonce automatique des routes** : FRR sur chaque UPF redistribue automatiquement les routes locales UE 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 Route

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** : Compte des routes synchronisées avec succès et des 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 des sessions 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 les statistiques de synchronisation mises à jour

Flux de Route

Avantages

- **Provisionnement Sans Intervention** : Les routes sont gérées automatiquement sans intervention manuelle
- **Adaptation Dynamique** : Le routage du réseau s'adapte en temps réel à la mobilité et aux changements de session des UE
- **Scalabilité** : Prend en charge des milliers de routes UE concurrentes

- **Résilience** : Les opérations de synchronisation échouées sont suivies et peuvent être réessayées
- **Visibilité** : Visibilité complète sur 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 protocole de routage (voisins OSPF, pairs BGP)
- `GET /api/v1/ospf/database/external` - Obtenir la base de données LSA AS-External OSPF (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/mask de destination
- Passerelle/prochain saut
- Liaison d'interface
- Métrique et distance administrative

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 de Réseau (10.98.0.20)** : L'annonce de réseau propre du UPF
- **LSA de 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