Guide des Opérations OmniPGW

OmniPGW - Passerelle de Contrôle de Paquet (PGW-C)

par Omnitouch Network Services

Table des Matières

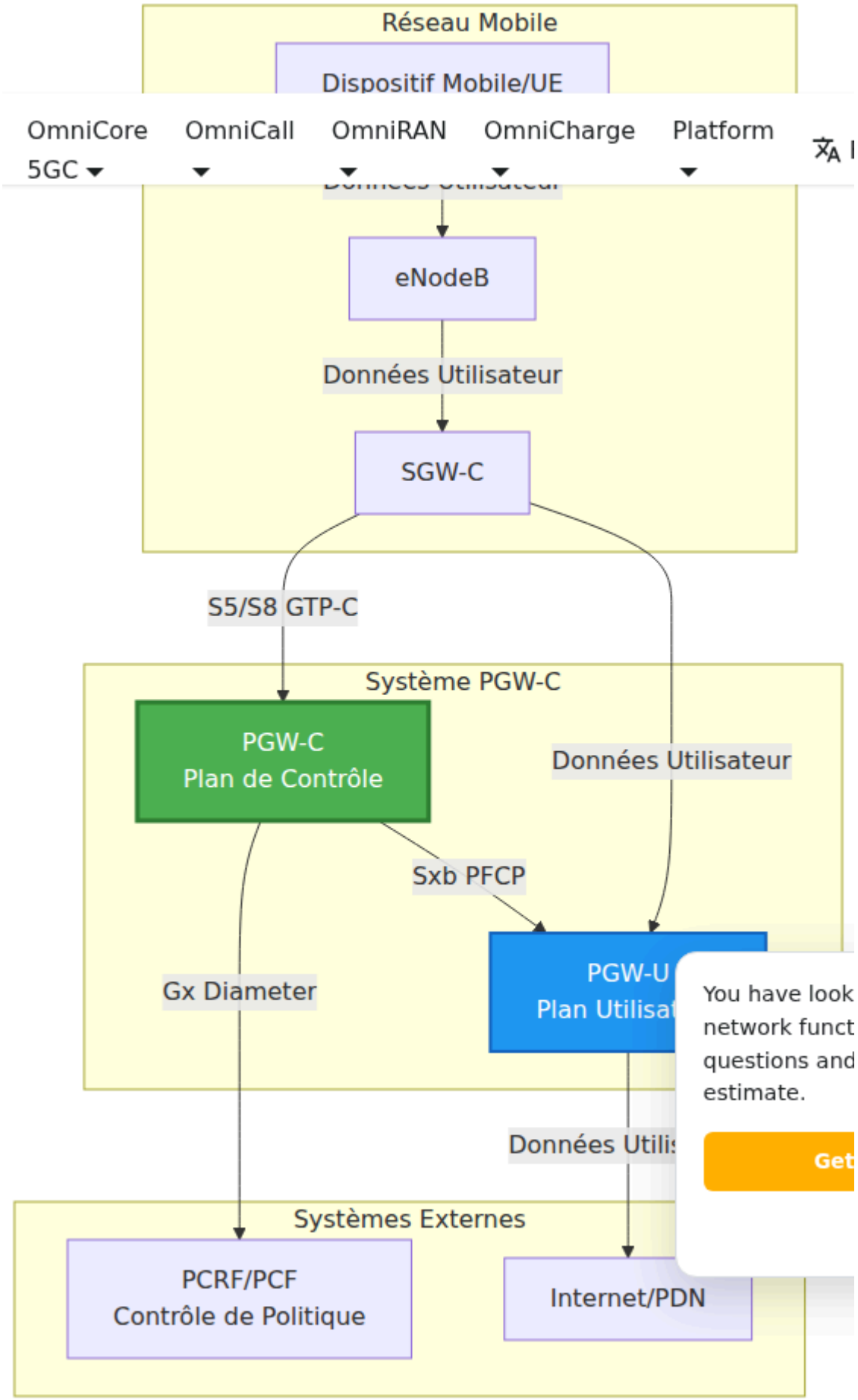
1. Aperçu
2. Architecture
3. Interfaces Réseau
4. Concepts Clés
5. Prise en Main
6. Configuration
7. Surveillance des Opérations en Temps Réel (API OAM)
8. Surveillance & Métriques
9. Documentation Détailée
10. Ressources Supplémentaires

Aperçu

OmniPGW est une implémentation de haute performance de la Passerelle de Contrôle de Paquet (PGW-C) pour les réseaux 3GPP LTE Evolved Packet Core (EPC), développée par Omnitouch Network Services. Elle gère les fonctions de plan de contrôle pour les sessions de données, y compris :

- **Gestion de Session** - Création, modification et terminaison des sessions de données UE (Équipement Utilisateur)

- **Attribution d'Adresse IP** - Attribution d'adresses IP aux appareils mobiles à partir de pools configurés
- **Contrôle de Politique et de Facturation** - Interface avec PCRF pour l'application des politiques et la facturation
- **Coordination du Plan Utilisateur** - Contrôle du PGW-U (Plan Utilisateur) pour le transfert de paquets


Réseau Mobile
Dispositif Mobile/UE
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
eNodeB
Données Utilisateur
SGW-C
S5/S8 GTP-C
Système PGW-C
PGW-C
Plan de Contrôle
Données Utilisateur
Sxb PFCP
Gx Diameter
PGW-U
Plan Utilisat
You have look
network funct
questions and
estimate.
Get
Systèmes Externes
PCRF/PCF
Contrôle de Politique
Internet/PDN

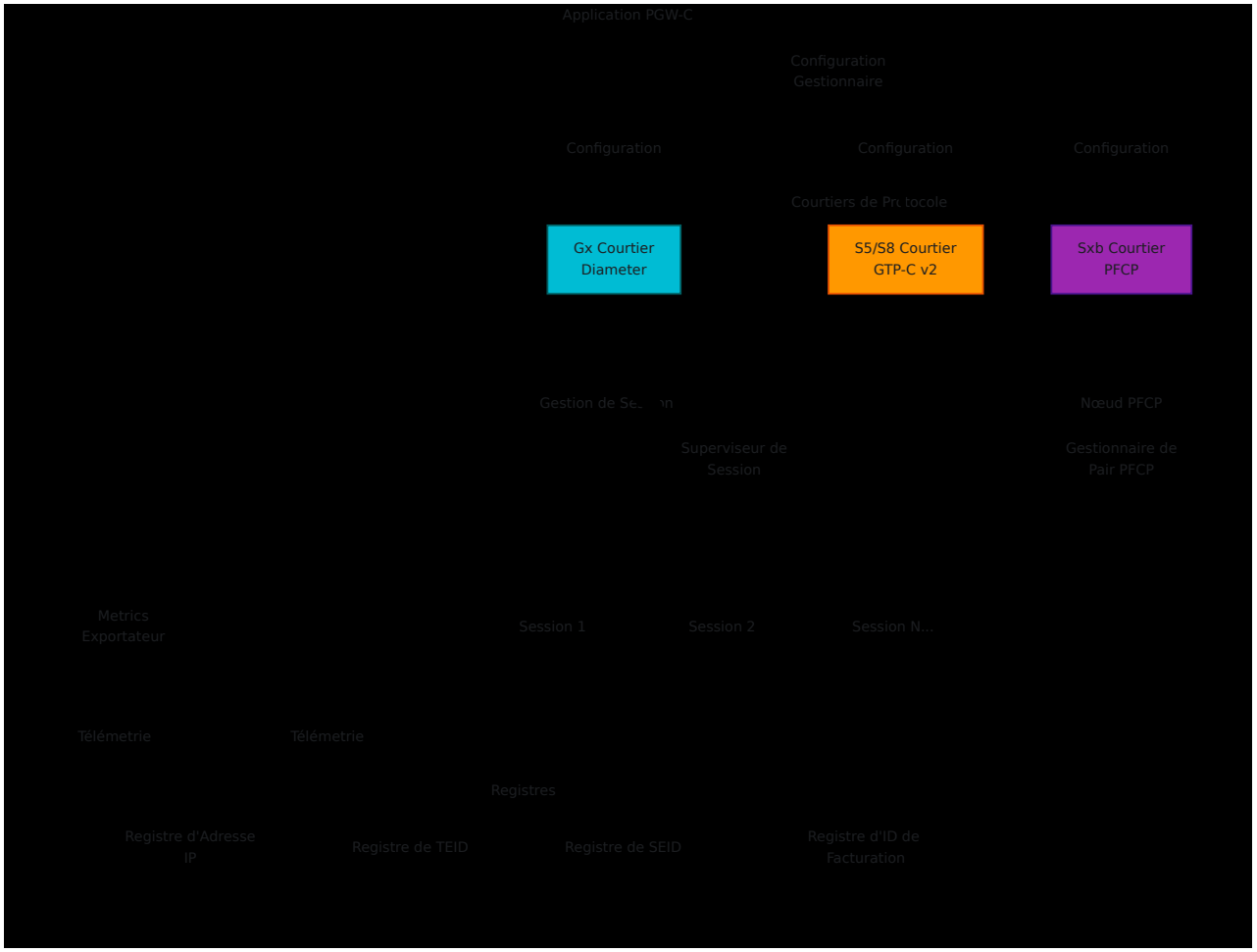
Ce que fait PGW-C

- **Accepte les demandes de session** de SGW-C via l'interface S5/S8 (GTP-C)
- **Attribue des adresses IP UE** à partir de pools de sous-réseaux configurés
- **Demande des décisions de politique** à PCRF via l'interface Gx (Diameter)
- **Programme les règles de transfert** dans PGW-U via l'interface Sxb (PFCP)
- **Gère l'application QoS** à travers des contextes de porteur et des règles QoS
- **Suit les informations de facturation** pour les systèmes de facturation

---

Architecture

Vue d'Ensemble des Composants



Architecture des Processus

PGW-C est construit sur Elixir/OTP et utilise une architecture de processus supervisée :

- **Superviseur d'Application** - Superviseur de niveau supérieur gérant tous les composants
- **Courtiers de Protocole** - Gèrent les messages de protocole entrants/sortants
- **Processus de Session** - Un GenServer par connexion PDN active
- **Registres** - Suivent les ressources allouées (IPs, TEIDs, SEIDs, etc.)

- **Gestionnaire de Nœud PFCP** - Maintient les associations PFCP avec les pairs PGW-U

Chaque composant est supervisé et redémarrera automatiquement en cas d'échec, garantissant la fiabilité du système.

Interfaces Réseau

PGW-C implémente trois interfaces 3GPP principales :

Interface S5/S8 (GTP-C v2)

But : Signalisation de plan de contrôle entre SGW-C et PGW-C

Protocole : GTP-C Version 2 sur UDP

Messages Clés :

- Demande/Réponse de Création de Session
- Demande/Réponse de Suppression de Session
- Demande/Réponse de Création de Porteur
- Demande/Réponse de Suppression de Porteur

Configuration : Voir Configuration S5/S8

Interface Sxb (PFCP)

But : Signalisation de plan de contrôle entre PGW-C et PGW-U

Protocole : PFCP (Protocole de Contrôle de Transfert de Paquet) sur UDP

Messages Clés :

- Demande/Réponse de Configuration d'Association
- Demande/Réponse d'Établissement de Session
- Demande/Réponse de Modification de Session

- Demande/Réponse de Suppression de Session
- Demande/Réponse de Heartbeat

Configuration : Voir Documentation de l'Interface PFCP/Sxb

Interface Gx (Diameter)

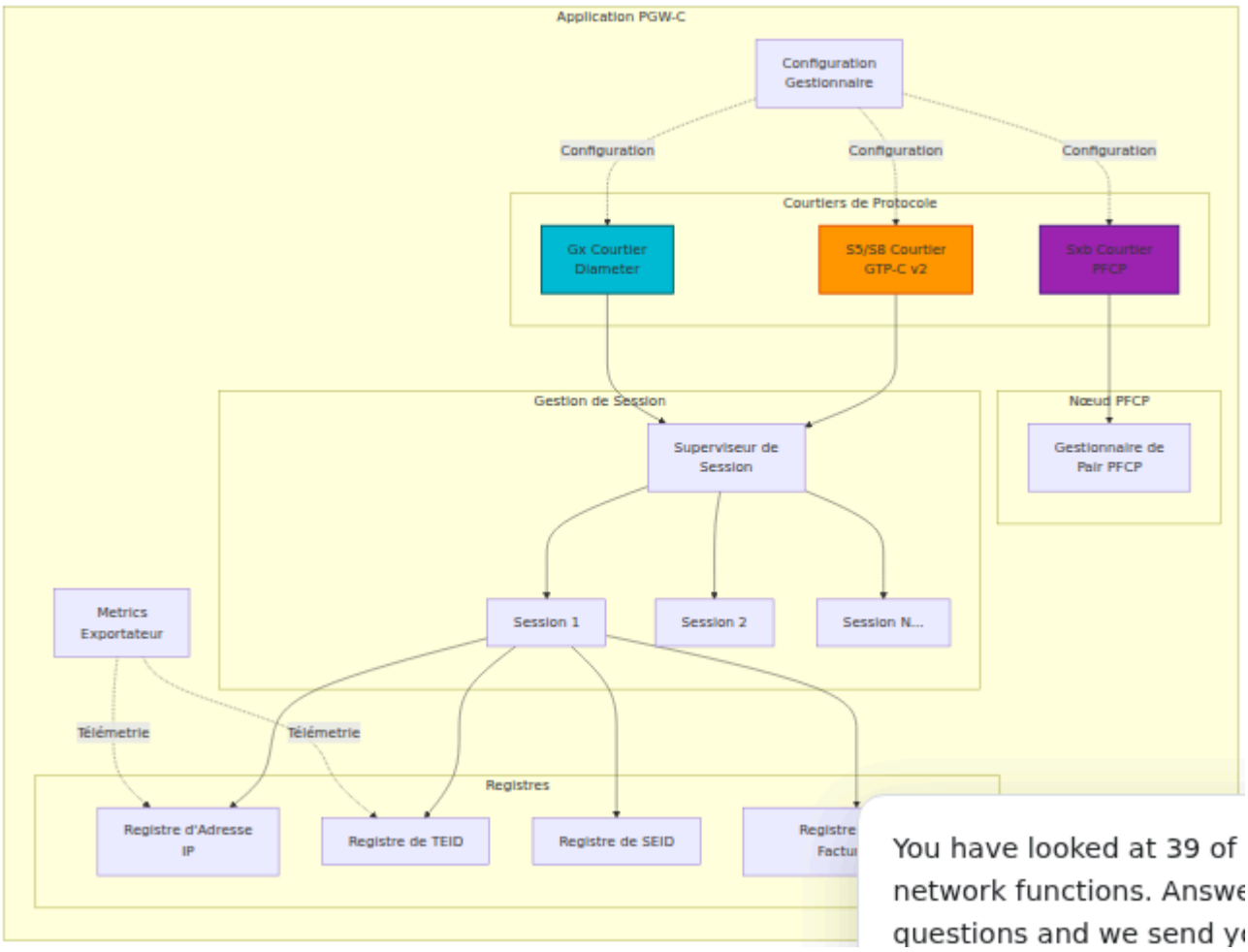
But : Interface de Fonction de Règles de Politique et de Facturation (PCRF)

Protocole : Diameter (IETF RFC 6733)

Messages Clés :

- Demande/Réponse de Contrôle de Crédit Initial (CCR-I/CCA-I)
- Demande/Réponse de Contrôle de Crédit de Terminaison (CCR-T/CCA-T)

Configuration : Voir Documentation de l'Interface Diameter Gx



Concepts Clés

Session PDN

Une Session PDN (Packet Data Network) représente la connexion de données d'un UE à un réseau externe (comme Internet). Chaque session a :

- **Adresse IP UE** - Attribuée à partir d'un pool de sous-réseaux configuré
- **APN** (Nom de Point d'Accès) - Identifie le réseau externe
- **Contexte de Porteur** - Contient des paramètres QoS et des informations de tunnel
- **ID de Facturation** - Identifiant unique pour la facturation
- **TEID** (ID de Point de Terminaison de Tunnel) - Identifiant de tunnel de l'interface S5/S8
- **SEID** (ID de Point de Terminaison de Session) - Identifiant de session de l'interface Sxb

Contexte de Porteur

Un porteur représente un flux de trafic avec des caractéristiques QoS spécifiques :

- **Porteur par Défaut** - Créé avec chaque session PDN
- **Porteurs Dédiés** - Porteurs supplémentaires pour des besoins QoS spécifiques
- **EBI** (ID de Porteur EPS) - Identifiant unique pour chaque porteur
- **Paramètres QoS** - QCI, ARP, débits (MBR, GBR)

Règles PFCP

Le PGW-C programme le PGW-U avec des règles de traitement de paquets :

- **PDR** (Règle de Détection de Paquet) - Correspond aux paquets (montant/descendant)
- **FAR** (Règle d'Action de Transfert) - Spécifie le comportement de transfert
- **QER** (Règle d'Application de QoS) - Applique des limites de débit

- **BAR** (Règle d'Action de Mise en Tampon) - Contrôle la mise en tampon des paquets

Voir Documentation de l'Interface PFCP pour plus de détails.

Attribution d'Adresse IP

Les adresses IP UE sont attribuées à partir de pools de sous-réseaux configurés :

- **Sélection basée sur APN** - Différents APNs peuvent utiliser différents sous-réseaux
- **Attribution dynamique** - Sélection aléatoire d'IP à partir de la plage disponible
- **Attribution statique** - Support pour les adresses IP demandées par l'UE
- **Détection de collision** - Assure une attribution d'IP unique

Voir Attribution de Pool IP UE pour la configuration, et IPv6 / Double-Stack pour le fonctionnement PDN IPv6 et IPv4v6.

Prise en Main

Prérequis

- Elixir ~1.16
- Erlang/OTP 26+
- Connectivité réseau avec SGW-C, PGW-U, et PCRF
- Compréhension de l'architecture EPC LTE

Démarrer OmniPGW

1. **Configurer les paramètres d'exécution** dans `config/runtime.exs`
2. **Compiler l'application :**

```
mix deps.get
mix compile
```

3. **Démarrer l'application :**

```
mix run --no-halt
```

Vérification de l'Opération

Vérifiez les journaux pour un démarrage réussi :

```
[info] Démarrage d'OmniPGW...
[info] Démarrage de l'Exportateur de Métriques sur
127.0.0.42:42069
[info] Démarrage du Courtier S5/S8 sur 127.0.0.10
[info] Démarrage du Courtier Sxb sur 127.0.0.20
[info] Démarrage du Courtier Gx
[info] Démarrage du Gestionnaire de Nœud PFCP
[info] OmniPGW démarré avec succès
```

Accédez aux métriques à `http://127.0.0.42:42069/metrics` (adresse configurée).

Configuration

Toute la configuration d'exécution est définie dans `config/runtime.exs`. Ce fichier est évalué au démarrage de l'application. Un exemple complet suit ; chaque section est décrite dans le tableau de référence ci-dessous et dans la Référence de Configuration.

```
# config/runtime.exs
import Config

# Configuration du Logger
config :logger, level: :info

config :pgw_c,
  # Métriques (Prometheus)
  metrics: %{
    enabled: true,
    ip_address: "10.0.0.20",  # Réseau de gestion
    port: 9090,
    registry_poll_period_ms: 5_000
  },

  # Diameter/Gx (interface PCRF)
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf-primary.epc.mnc001.mcc001.3gppnetwork.org",
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        ip: "10.0.1.30",
        initiate_connection: true
      },
      %{
        host: "pcrf-backup.epc.mnc001.mcc001.3gppnetwork.org",
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        ip: "10.0.2.30",
        initiate_connection: true
      }
    ]
  },

  # S5/S8 (interface SGW-C)
  s5s8: %{
    local_ipv4_address: "10.0.0.20"
  },

  # Sxb/PFCP (interface PGW-U)
  sxb: %{
```

```
    local_ip_address: "10.0.0.20"
  },

  # Sélection UPF
  upf_selection: %{
    rules: [
      %{
        name: "Internet par Défaut",
        priority: 1,
        match_field: :apn,
        match_regex: ~r/^internet/,
        upf_pool: [
          %{remote_ip_address: "10.0.0.21", remote_port: 8805,
weight: 100},
          %{remote_ip_address: "10.0.0.22", remote_port: 8805,
weight: 0}  # Standby
        ]
      }
    ],
    fallback_pool: [
      %{remote_ip_address: "10.0.0.21", remote_port: 8805, weight:
100}
    ]
  },

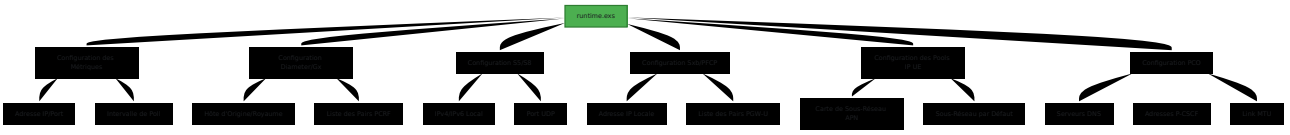
  # Pools IP UE
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.0.0/18"  # 16,382 IPs
      ],
      "ims" => [
        "100.64.64.0/22"  # 1,022 IPs
      ],
      "enterprise.corp" => [
        "10.100.0.0/16"  # 65,534 IPs
      ],
      default: [
        "100.64.127.0/24"  # 254 IPs
      ]
    }
  },

  # Options de Configuration de Protocole
```

```
  pco: %{
    primary_dns_server_address: "8.8.8.8",
    secondary_dns_server_address: "8.8.4.4",
    p_cscf_ipv4_address_list: ["10.0.0.50", "10.0.0.51"],
    ipv4_link_mtu_size: 1400
  }
```

La configuration est structurée en plusieurs sections :

Vue d'Ensemble de la Configuration



Référence de Configuration Rapide

Section	But	Documentation
metrics	Exportateur de métriques Prometheus	Guide de Surveillance
diameter	Interface Gx vers PCRF	Configuration Diameter Gx
s5s8	Interface GTP-C vers SGW-C	Configuration S5/S8
sxb	Interface PFCP vers PGW-U	Configuration PFCP
ue	Pools d'adresses IP UE	Configuration de Pool IP
pco	Options de Configuration de Protocole	Configuration PCO
CDR	Facturation hors ligne & rapports d'utilisation	Format CDR
charging	Mode de facturation en ligne/hors ligne, contournement d'urgence	Contournement de Facturation

Voir le Guide Complet de Configuration pour des informations détaillées.

Surveillance des Opérations en Temps Réel (API OAM)

OmniPGW expose une **API REST OAM** pour la surveillance et les opérations en temps réel, fournissant une visibilité instantanée sur l'état du système. Interrogez ces points de terminaison de lecture à la demande (ou à intervalles

de polling / depuis un script) pour une vue opérationnelle en direct - aucun accès à la ligne de commande de l'hôte ou requêtes de métriques brutes nécessaires.

Accéder à l'API OAM

L'API est servie sur HTTPS/TLS. Tous les itinéraires sont servis sous le préfixe `/api`.

```
https://<omnipgw-ip>:8443/
```

La documentation interactive de l'API (Swagger UI) est disponible à :

```
https://<omnipgw-ip>:8443/api/docs
```

Exemple de vérification de santé (utilisez `-k` pour un certificat de laboratoire auto-signé) :

```
curl -k https://localhost:8443/api/status
{"result":"ok"}
```

Points de Terminaison Disponibles :

But	Point de Terminaison	Remplace (ancienne page)
Plongée dans un abonné spécifique (IMSI/MSISDN/IP)	`POST /api/ue_search`	Recherche UE
Lister / rechercher les sessions PDN actives	`GET /api/sessions[?search=<imsi|msisdn|ip>]`	Sessions PGW
Une session par IMSI	`GET /api/sessions/<imsi>`	Sessions PGW (détail)
Historique des sessions / événements d'audit	`GET /api/session_history[?type=&search=&page=&page_size=]`	Historique des Sessions
Vue d'ensemble de la topologie réseau	`GET /api/topology`	Topologie Réseau
Utilisation du pool d'adresses IP UE	`GET /api/ip_pools`, `GET /api/ip_pools/<name>`	Pools IP
État des pairs UPF / PFCP	`GET /api/upf`, `GET /api/upf/<ip>`	Sessions PFCP / État UPF
Règles de sélection UPF, santé, configuration PCO	`GET /api/upf_selection`	Sélection UPF
Connectivité des pairs Diameter	`GET /api/diameter[?status=connected|disconnected]`,	Pairs Diameter

But	Point de Terminaison	Remplace (ancienne page)
(Gx/Gy)	`GET /api/diameter/<origin-host>`	
État de facturation en ligne (Gy) par session	`GET /api/charging[?search=<imsi>]`, `GET /api/charging/<imsi>`	État Gy
Contournement : désactiver la facturation en ligne à l'échelle du nœud	`PUT /api/charging/override`, `GET /api/charging/override`, `DELETE /api/charging/override`	-
État de découverte DNS P-CSCF	`GET /api/pcscf_monitor`	Moniteur P-CSCF
Vérification de santé	`GET /api/status`	-

Pour une vue de style tableau de bord en direct, interrogez les points de terminaison de collection à intervalles, par exemple :

```
watch -n 2 'curl -sk https://localhost:8443/api/sessions'
watch -n 1 'curl -sk https://localhost:8443/api/diameter?
status=disconnected'
```

> Remarque : l'ancienne page **Logs** (streaming de journaux en direct) et la page **Gy Simulator** n'ont pas d'équivalent API OAM. Stream les journaux depuis l'hôte / votre agrégateur de journaux, et utilisez `GET /api/charging` pour inspecter l'état en direct de Gy/facturation en ligne.

Fonctionnalités Clés

Données à la Demande et Pollées :

- Chaque point de terminaison retourne l'état actuel en direct des processus OmniPGW
- Poll à un intervalle avec `watch`/cron/un script pour une vue auto-rafraîchissante
- Les champs d'état (par exemple, pair `connected`/`associated`) sont retournés explicitement dans le JSON

Recherche & Filtre :

- Rechercher des sessions par IMSI, IP, MSISDN ou APN via `GET /api/sessions?search=...`
- Filtrer les pairs Diameter par `?status=connected|disconnected`
- Filtrer l'état de facturation par `?search=<imsi>`

Enregistrements de Détail Complet :

- Points de terminaison par entité (`/api/sessions/<imsi>`, `/api/upf/<ip>`, `/api/diameter/<origin-host>`, `/api/charging/<imsi>`) retournent l'état complet en JSON
- Inspecter la session complète, la configuration des pairs et les capacités

Contrôle d'Accès :

- Servi sur HTTPS/TLS sur le port 8443
- Lier uniquement à l'IP de gestion et front avec vos contrôles d'accès
- Destiné à l'équipe NOC/opérations et à l'utilisation automatisée

Flux de Travail Opérationnels

D��pannage de Session (Plongée Profonde) :

```
1. L'utilisateur signale un problème de connexion
2. POST /api/ue_search avec un corps JSON contenant l'IMSI, MSISDN
ou IP
3. Examiner les détails complets de la session dans la réponse :
   a) Sessions Actives - Vérifier si la session existe avec les
bons paramètres
   b) Localisation Actuelle - Vérifier TAC, ID de Cellule, réseau
de service
   c) Informations sur le Porteur - Vérifier les porteurs par
défaut et dédiés
      - QCI, MBR/GBR, Noms de Règles de Facturation
      - Limites APN-AMBR
   d) Informations de Facturation - ID de session Gy, état du
quota (également GET /api/charging/<imsi>)
   e) Informations de Politique - Session Gx, règles PCC
installées
   f) Événements Récents - GET /api/session_history?search=<imsi>
4. Si session non trouvée → GET /api/diameter?status=disconnected
pour la connectivité PCRF
```

Recherche Rapide de Session :

```
1. L'utilisateur signale un problème
2. GET /api/sessions?search=<imsi|msisdn> (ou GET
/api/sessions/<imsi>)
3. Vérifier si la session existe avec des détails de base :
   - Adresse IP UE attribuée
   - Paramètres QoS
   - Points de terminaison de tunnel établis
4. Pour une analyse détaillée → POST /api/ue_search
```

Vérification de Santé du Système :

```
1. GET /api/upf              → Vérifier que tous les pairs PGW-U
sont "Associés"
2. GET /api/diameter?status=connected → Vérifier que tous les
pairs PCRF sont "Connectés"
3. GET /api/sessions         → Vérifier le nombre de sessions
actives par rapport à la capacité
4. GET /api/status           → Vérification de santé globale
({"result":"ok"})
```

Surveillance de Capacité :

- Compter les entrées retournées par `GET /api/sessions`
- Comparer à la capacité sous licence/attendue
- Identifier les pics d'utilisation
- Surveiller la distribution à travers les APNs ; utiliser `GET /api/ip_pools` pour l'utilisation par pool

API vs. Métriques

Utilisez l'API OAM pour :

- Dépannage approfondi des abonnés (`POST /api/ue_search`)
- Détails et inspection de l'état de session individuelle (`GET /api/sessions/<imsi>`)
- État des pairs en temps réel (`GET /api/upf`, `GET /api/diameter`)
- Vérifications de santé rapides à travers toutes les interfaces (`GET /api/status`)
- Dépannage d'utilisateurs spécifiques par IMSI/MSISDN/IP
- Analyse de QoS des porteurs (MBR, GBR, QCI)
- Inspection des règles de politique et de facturation (`GET /api/charging`)
- Historique des sessions et pistes d'audit (`GET /api/session_history`)
- Surveillance de la capacité du pool IP (`GET /api/ip_pools`)
- Vérification de la configuration et des règles (`GET /api/upf_selection`)

Utilisez les Métriques Prometheus pour :

- Tendances historiques
- Alertes et notifications
- Graphiques de planification de capacité
- Analyse de performance
- Surveillance à long terme

Meilleure Pratique : Utilisez les deux ensemble - l'API OAM pour les opérations immédiates, Prometheus pour les tendances et les alertes.

Surveillance & Métriques

En plus de l'API OAM, OmniPGW expose des métriques compatibles avec Prometheus pour la surveillance :

Métriques Disponibles

- **Métriques de Session**
  - `teid_registry_count` - Sessions S5/S8 actives
  - `seid_registry_count` - Sessions PFCP actives
  - `session_id_registry_count` - Sessions Gx actives
  - `address_registry_count` - Adresses IP UE attribuées
  - `charging_id_registry_count` - IDs de facturation actifs
- **Métriques de Message**
  - `s5s8_inbound_messages_total` - Messages GTP-C reçus
  - `sxb_inbound_messages_total` - Messages PFCP reçus
  - `gx_inbound_messages_total` - Messages Diameter reçus
  - Durées de traitement des messages
- **Métriques d'Erreur**
  - `s5s8_inbound_errors_total` - Erreurs de protocole S5/S8
  - `sxb_inbound_errors_total` - Erreurs de protocole PFCP

    - `gx_inbound_errors_total` - Erreurs Diameter

- **Métriques de Cycle de Vie de Session**

    - `pgw_session_create_total{result,cause,pdn_type,rat_type}` - Tentatives de création de session PDN, décomposées par résultat (`success`/`failure`), la cause GTP retournée, type de PDN (`ipv4`/`ipv6`/`ipv4v6`) et type RAT
    - `pgw_session_delete_total{pdn_type}` - Suppressions de session PDN par type de PDN
    - `pgw_session_modify_total` - Requêtes de modification de Bearer traitées

- **Métriques de Pool IP UE**

    - `ue_pool_addresses_allocated{pool,ip_version}` - Adresses UE allouées par pool (modèle APN) et version IP
    - `ue_pool_addresses_total{pool,ip_version}` - Capacité totale du pool
    - `ue_pool_addresses_utilization_ratio{pool,ip_version}` - Fraction allouée (0.0-1.0); utile pour l'alerte d'épuisement de pool. Notez que les pools partageant une plage d'adresses comptent chacun les allocations partagées
    - `ue_ip_allocation_failures_total{ip_version,reason}` - Échecs d'allocation (`pool_exhausted`, `already_registered`)

- **Métriques UPF / PFCP par Pair**

    - `upf_peer_associated{peer_ip}` - État d'association PFCP 1/0 par pair
    - `upf_peer_healthy{peer_ip}` - Santé 1/0 par pair
    - `upf_peer_missed_heartbeats{peer_ip}` - Battements de cœur consécutifs manqués par pair
    - `upf_heartbeat_rtt` - Distribution du temps de réponse aller-retour des battements de cœur PFCP, par `peer_ip`

- **Métriques de Bearer & QoS**

    - `pgw_bearer_create_total{type,qci}` - Bearers créés par type (`default`/`dedicated`) et QCI

- `pgw_bearer_delete_total{type}` - Bearers supprimés par type

- **Métriques de Pair Diameter**
  - `diameter_peers_connected{application}` - Pairs Diameter connectés par application (`gx`/`gy`/`all`)

Accéder aux Métriques

Les métriques sont exposées via HTTP à l'endpoint configuré :

```
curl http://127.0.0.42:42069/metrics
```

Voir le Guide de Surveillance & Métriques pour la configuration du tableau de bord et l'alerte.

Documentation Détailée

Cette section fournit un aperçu complet de toute la documentation OmniPGW. Les documents sont organisés par sujet et cas d'utilisation.

Structure de la Documentation

```
Documentation OmniPGW
├── OPERATIONS.md (Ce Guide)
│
└── docs/
    ├── Configuration & Mise en Place
    │   ├── configuration.md          Référence complète de
runtime.exs
    │   ├── ue-ip-allocation.md       Configuration du pool IP
    │   └── pco-configuration.md      Paramètres DNS, P-CSCF, MTU
    │
    ├── Interfaces Réseau
    │   ├── pfcp-interface.md         Sxb/PFCP (communication PGW-
U)
    │   ├── diameter-gx.md            Gx (communication PCRF)
    │   ├── diameter-gy.md            Gy/Ro (communication OCS)
    │   └── s5s8-interface.md         S5/S8 (communication SGW-C)
    │
    └── Opérations
        ├── session-management.md     Cycle de vie de session PDN
        ├── charging-override.md      Surcharge de facturation en
ligne
        └── monitoring.md             Métriques Prometheus &
alerte
```

Documentation par Sujet

□ Introduction

Document	Description	Objectif
OPERATIONS.md	Guide principal des opérations (ce document)	Vue d'ensemble et démarrage rapide

⚙ Configuration

Document	Description	Lignes
configuration.md	Référence complète de configuration runtime.exs	1,600+
ue-ip-allocation.md	Gestion et allocation du pool IP UE	943
ipv6-dual-stack.md	Opération PDN IPv6 / IPv4v6 à travers PGW-C et UPF	-
pco-configuration.md	Options de configuration de protocole (DNS, P-CSCF, MTU)	344

Interfaces Réseau

Document	Description	Lignes
pfcp-interface.md	Interface PFCP/Sxb vers PGW-U	1,355
diameter-gx.md	Interface Diameter Gx vers PCRF (Contrôle de Politique)	941
diameter-gy.md	Interface Diameter Gy/Ro vers OCS (Facturation en Ligne)	1,100+
charging-override.md	Surcharge de bris de verre pour désactiver la facturation en ligne pendant un incident OCS	-
s5s8-interface.md	Interface GTP-C S5/S8 vers SGW-C	456

Opérations & Surveillance

Document	Description	Lignes
session-management.md	Cycle de vie et opérations de session PDN	435
monitoring.md	Métriques Prometheus, tableaux de bord Grafana, alertes	807
data-cdr-format.md	Format de fichier CDR, configuration URR, facturation hors ligne	847
qos-bearers.md	Gestion QoS & bearer, contrôle de politique	448
troubleshooting.md	Procédures de dépannage et problèmes courants	687

□ Fonctionnalités Avancées

Document	Description	Lignes
pcscf-monitoring.md	Découverte P-CSCF et surveillance de santé	894

Caractéristiques de la Documentation

□ Diagrammes Mermaid

Tous les documents incluent des **graphes Mermaid** pour une compréhension visuelle :

- Diagrammes d'architecture
- Diagrammes de séquence (flux de messages)
- Machines d'état
- Topologie réseau

Exemples Pratiques

Chaque document inclut :

- Exemples de configuration du monde réel
- Configurations prêtes à copier-coller
- Cas d'utilisation courants

Dépannage

Chaque document d'interface inclut :

- Problèmes courants et solutions
- Commandes de débogage
- Métriques pour le diagnostic

Renvois Croisés

Les documents sont largement interliés pour une navigation facile.

Chemins de Lecture

Pour les Opérateurs Réseau

1. OPERATIONS.md - Vue d'ensemble (ce document)
2. configuration.md - Mise en place
3. monitoring.md - Surveillance
4. session-management.md - Opérations quotidiennes

Pour les Ingénieurs Réseau

1. OPERATIONS.md - Vue d'ensemble de l'architecture (ce document)
2. pfcp-interface.md - Contrôle du plan utilisateur
3. diameter-gx.md - Contrôle de politique
4. diameter-gy.md - Facturation en ligne
5. s5s8-interface.md - Gestion de session
6. ue-ip-allocation.md - Gestion IP

Pour la Configuration & Déploiement

1. configuration.md - Référence complète
2. ue-ip-allocation.md - Pools IP
3. pco-configuration.md - Paramètres réseau
4. monitoring.md - Configurer la surveillance

Statistiques du Document

- **Documents Totaux :** 14
- **Lignes Totales :** ~10,900+
- **Taille Totale :** ~265 Ko
- **Diagrammes Mermaid :** 75+
- **Exemples de Code :** 150+

Concepts Clés Couverts

Architecture

- □ Séparation du plan de contrôle/utilisateur
- □ Architecture OTP/Elixir
- □ Supervision des processus
- □ Sessions basées sur GenServer

Protocoles

- □ PFCP (Protocole de Contrôle de Transfert de Paquet)
- □ GTP-C v2 (Protocole de Tunnel GPRS)
- □ Diameter (RFC 6733)

Interfaces 3GPP

- □ Sxb (PGW-C ↔ PGW-U)
- □ Gx (PGW-C ↔ PCRF)
- □ Gy/Ro (PGW-C ↔ OCS)
- □ S5/S8 (SGW-C ↔ PGW-C)

Opérations

- □ Gestion de session
- □ Stratégies d'allocation IP
- □ Application de QoS
- □ Intégration de facturation
- □ Surveillance & alerte

Ressources Supplémentaires

Spécifications 3GPP

Spéc	Titre
TS 29.274	GTP-C v2 (interface S5/S8)
TS 29.244	PFCP (interface Sxb)
TS 29.212	Interface Diameter Gx (Contrôle de Politique)
TS 32.299	Applications de Facturation Diameter (Gy/Ro)
TS 32.251	Facturation de domaine à commutation de paquets
TS 23.401	Architecture EPC

Documentation Connexe

- Référence de configuration : configuration.md - Référence complète de `config/runtime.exs`

Surcharge en ligne Override (Break-Glass)

Le surcharge en ligne override est un commutateur au niveau du nœud. Il retire la facturation en ligne (Gy/OCS) du chemin immédiatement, sans redémarrage. Utilisez-le lors d'un incident OCS, lorsque l'OCS est défaillant et que les abonnés sont affectés.

Un redémarrage n'est pas un moyen acceptable de désactiver la facturation en ligne. Un redémarrage coupe chaque session PDN active sur le nœud. L'override change le comportement pour les sessions déjà actives et pour les sessions qui se connectent pendant qu'il est actif.

La facturation hors ligne n'est pas affectée. Le PGW-C continue de mesurer l'utilisation et d'écrire des CDR pendant que l'override est actif. Voir Format CDR.

Table des matières

1. Quand l'utiliser
2. Ce que fait l'override
3. Référence API
4. Procédure opérationnelle
5. Trace d'audit
6. Configuration
7. Métriques
8. Dépannage

Quand l'utiliser

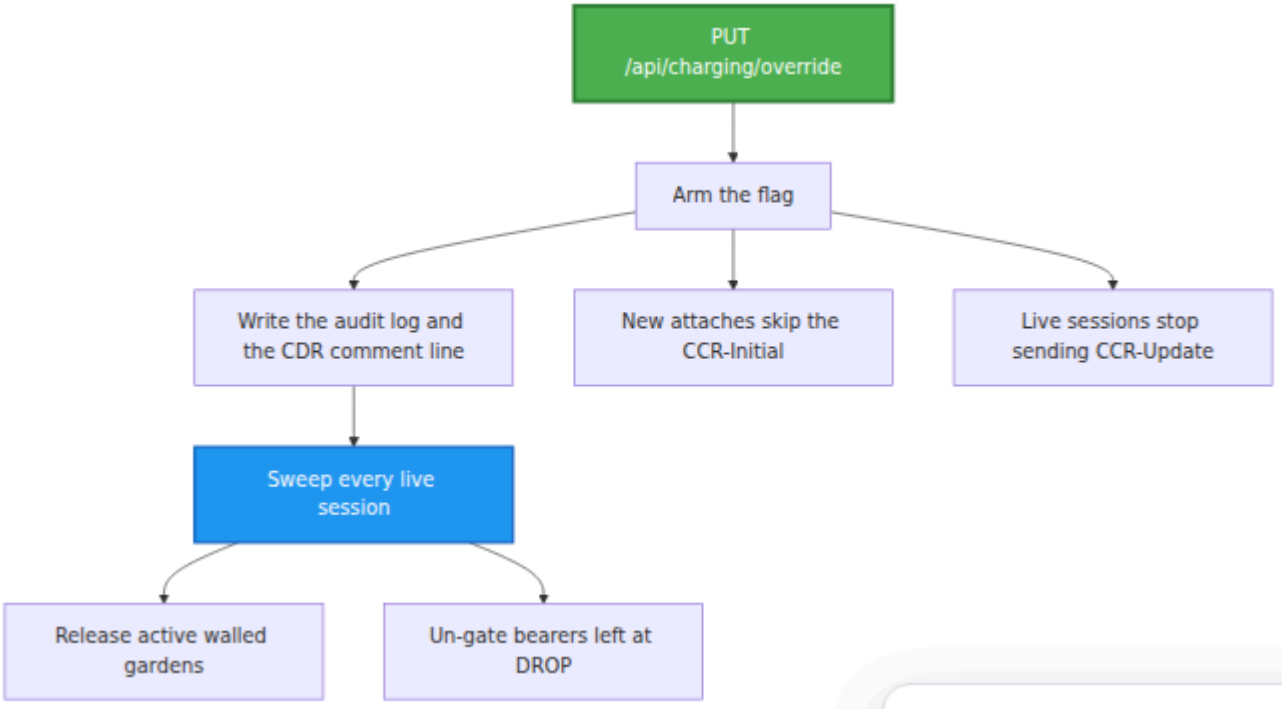
Utilisez l'override lorsque l'OCS cause un impact sur les abonnés que vous ne pouvez pas résoudre au niveau de l'OCS. Trois modes de défaillance sont courants.

Mode de défaillance	Ce que voient les abonnés	Ce que fait l'override
L'OCS rejette ou ne répond pas au CCR-Initial	Les nouvelles connexions sont bloquées ou se retrouvent dans le jardin clos	Les nouvelles connexions n'envoient pas de CCR-Initial et se connectent avec le trafic transféré
L'OCS répond, mais sa comptabilité est incorrecte	Les abonnés sont coupés avec un solde restant	Les jardins clos sont libérés et les porteurs bloqués sont débloqués
L'OCS abandonne les réservations et épuise les paquets	Le solde disparaît sans utilisation	Aucun CCR-Update supplémentaire n'est envoyé, donc la fenêtre n'est pas signalée et n'est pas facturée

N'utilisez pas l'override pour un seul abonné. Pour déplacer une session dans ou hors du jardin clos, utilisez `PUT /api/charging/<imsi>`. Voir Interface Diameter Gy.

La portée de l'override est un nœud. Si vous exécutez plus d'un PGW-C, appliquez l'override à chaque nœud. Une application partielle est survivable. Certains abonnés se rétablissent, et le reste reste tel qu'il était.

Ce que fait l'override



Nouvelles sessions

Pendant que l'override est actif, le PGW-C traite la facturation en ligne comme désactivée. C'est le même comportement que la valeur de configuration `charging.online: :off`. Le PGW-C n'envoie pas de CCR-Initial. La connexion se termine et le porteur par défaut transfère le trafic.

L'override a priorité sur la configuration locale et le PCRF. Un Gx CCA qui définit `Online` sur activé ne réactive pas la facturation tant que l'override est actif.

Sessions existantes

Un drapeau de configuration seul aide uniquement les sessions qui se connectent après. Les abonnés qui sont déjà bloqués sont ceux que l'incident affecte. Lorsque vous activez l'override, le PGW-C balaie chaque session active et fait deux choses.

1. **Libérer le jardin clos.** Si la session est dans un jardin clos, le PGW-C envoie une modification de session PFCP qui supprime la redirection. C'est le même chemin qui s'exécute lorsque l'OCS restaure le crédit.

2. **Débloquer les porteurs bloqués.** Si un porteur a des règles d'action de transfert définies pour appliquer l'action `DROP`, le PGW-C envoie une modification de session PFCP qui les remet à `FORW`. Cela couvre les sessions bloquées par la branche CCR-Initial `DIAMETER_USER_UNKNOWN`.

Le balayage est fait de manière optimale. Une session qui ne répond pas est comptée comme échouée et nommée dans le journal. Le balayage continue avec les sessions restantes. Le PGW-C exécute le balayage en parallèle sur les sessions, donc le temps qu'il faut est déterminé par la réponse UPF la plus lente, pas par la somme de celles-ci.

Un cas ne peut pas être réparé. Si l'OCS a envoyé une action de dernière unité de TERMINER lors de la connexion, le PGW-C a effacé les paramètres de transfert sur ces règles. Le PGW-C ne peut pas les reconstruire à partir de l'état de session. Le journal nomme chaque règle qu'il n'a pas pu restaurer. L'abonné doit se reconnecter pour se rétablir.

Sessions qui ont déjà une session Gy ouverte

Pendant que l'override est actif, le PGW-C n'envoie pas de CCR-Update pour ces sessions. Le rapport d'utilisation est rejeté plutôt que mis en mémoire tampon. C'est délibéré. La désactivation ne doit pas facturer rétroactivement la fenêtre pendant laquelle l'override était actif.

Lors de la déconnexion, le PGW-C n'envoie pas de CCR-Terminate par défaut. Un CCR-Terminate libère proprement la réservation sur l'OCS, mais un par déconnexion surcharge un OCS déjà malade. L'OCS récolte la réservation avec son propre minuteur de session. Définissez `send_ccr_t` sur `true` si vous souhaitez que le CCR-Terminate soit envoyé quand même.

Désactivation

La désactivation ramène le nœud à une facturation normale immédiatement. Aucun redémarrage n'est nécessaire. Le PGW-C ne signale pas la fenêtre pendant laquelle l'override était actif. L'utilisation dans cette fenêtre est mesurée hors ligne et écrite dans des CDR, mais elle n'atteint jamais l'OCS.

Référence API

L'override est une ressource singleton dans la collection de facturation. L'identifiant `override` est réservé. Aucun IMSI ne peut être cette chaîne.

Toutes les routes sont servies sous le préfixe `/api` sur l'API OAM. Voir Surveillance des opérations en temps réel.

Activer

```
PUT /api/charging/override
```

Corps de la requête :

```
{
  "ttl_seconds": 3600,
  "reason": "Incident OCS INC-1234",
  "actor": "noc@example.com",
  "send_ccr_t": false
}
```

Paramètre	Type	Requis	Par défaut	Description
`ttl_seconds`	Entier	Oui	-	Secondes jusqu'à ce que l'override expire automatiquement. Minimum `1`, maximum `86400` (24 heures).
`reason`	Chaîne	Oui	-	Texte libre qui indique pourquoi l'override est actif. Il apparaît dans le journal, le fichier CDR et `GET`.
`actor`	Chaîne	Non	IP de l'appelant	Qui a tiré le levier. Le PGW-C enregistre l'adresse IP de l'appelant lorsque vous omettez cela.
`send_ccr_t`	Booléen	Non	`false`	Définir sur `true` pour toujours envoyer un Gy CCR-Terminate lors de la déconnexion. Laissez `false` pour rester silencieux.

Réponse :

```
{
  "online_charging": "off",
  "override": {
    "active": true,
    "reason": "Incident OCS INC-1234",
    "actor": "noc@example.com",
    "ttl_seconds": 3600,
    "remaining_seconds": 3599,
    "activated_at": "2026-09-15T04:10:22.481Z",
    "expires_at": "2026-09-15T05:10:22.481Z",
    "send_ccr_t": false,
    "max_ttl_seconds": 86400
  },
  "relief": {
    "sessions_swept": 1842,
    "walled_gardens_released": 37,
    "bearers_ungated": 5,
    "failed": 0,
    "changed": [
      { "imsi": "313380000003530", "apn": "internet", "actions":
["walled_garden_released"] }
    ]
  }
}
```

Champ	Description
`online_charging`	`off` tant que l'override est actif, `on` sinon.
`override.active`	`true` tant que l'override est armé et que le TTL n'a pas expiré.
`override.remaining_seconds`	Secondes jusqu'à l'expiration automatique. `null` lorsqu'aucun override n'est actif.
`relief.sessions_swept`	Combien de sessions actives le PGW-C a demandé de supprimer leurs blocs de facturation.
`relief.walled_gardens_released`	Combien de sessions ont quitté le jardin clos.
`relief.bearers_ungated`	Combien de sessions avaient des porteurs bloqués restaurés au transfert.
`relief.failed`	Combien de sessions n'ont pas répondu. Leur état reste inchangé.
`relief.changed`	Les sessions qui ont réellement changé, avec l'action effectuée sur chacune.

Le PGW-C rejette la requête avec `400 Bad Request` lorsque `ttl_seconds` est absent, hors de portée, ou n'est pas un entier, et lorsque `reason` est absent ou vide.

Lire l'état

```
GET /api/charging/override
```

La réponse contient les mêmes champs `online_charging` et `override` que la réponse d'activation, avec `remaining_seconds` comptés à partir du moment de la requête. Il n'y a pas de champ `relief`.

Le point de terminaison de la liste `GET /api/charging` contient également l'état de l'override dans un champ `charging_override`, à côté de l'état de facturation par session.

Désactiver

```
DELETE /api/charging/override
```

Corps de la requête (optionnel) :

```
{ "actor": "oncall@example.com" }
```

La réponse contient les mêmes champs que `GET`, avec `active` défini sur `false`.

Procédure opérationnelle

Suivez ces étapes lors d'un incident OCS.

1. Confirmez que l'OCS est la cause. Vérifiez `GET /api/diameter?status=disconnected` pour l'état du pair Gy, et `GET /api/charging` pour les sessions dans le jardin clos.
2. Armez l'override sur le premier PGW-C. Donnez un `reason` qui nomme le ticket d'incident. Définissez `ttl_seconds` sur une valeur qui couvre votre temps de réparation prévu.

3. Lisez le bloc `relief` dans la réponse. Confirmez que `failed` est `0`. Si ce n'est pas le cas, consultez Dépannage.
4. Répétez les étapes 2 et 3 sur chaque nœud PGW-C restant.
5. Confirmez que la métrique `charging_override_active` lit `1` sur chaque nœud.
6. Réparez l'OCS.
7. Désactivez l'override sur chaque nœud avec `DELETE /api/charging/override`.
8. Confirmez que la métrique `charging_override_active` lit `0` sur chaque nœud, et que les nouvelles connexions envoient à nouveau un CCR-Initial.

Si vous ne désactivez pas l'override, il expire automatiquement lorsque le TTL expire. Le PGW-C écrit un avertissement dans le journal et une ligne de commentaire dans le fichier CDR à l'expiration.

Trace d'audit

Un override qui peut être laissé silencieusement est une fuite de revenus. Le PGW-C enregistre chaque changement d'état à quatre endroits.

Journal

Le PGW-C écrit un `warning` lors de l'activation, de la désactivation, de l'expiration du TTL, et lors de la réactivation après un redémarrage. Chaque ligne est préfixée par `[CHARGING OVERRIDE]` et contient la raison, l'acteur et la date limite. Le balayage écrit un second `warning` avec le nombre de sessions changées et échouées.

Fichier CDR

Le PGW-C écrit une ligne de commentaire dans le fichier CDR qui est ouvert à ce moment-là. L'écriture est asynchrone, donc un répertoire CDR sur un montage lent ne peut pas bloquer ou échouer l'override lui-même. Les lignes de commentaire commencent par `#`, le même marqueur que l'en-tête du fichier CDR utilise, donc la médiation les ignore. La ligne reste dans

l'enregistrement du fichier, ce qui est ce dont vous avez besoin lorsque une fenêtre de trafic n'a pas été délibérément facturée.

```
# CHARGING OVERRIDE ACTIVATED at 2026-09-15T04:10:22.481Z: online
charging (Gy/OCS) disabled node-wide; usage in this window is
metered offline but NOT reported to the OCS. reason=OCS incident
INC-1234 actor=noc@example.com ttl_seconds=3600 expires_at=2026-
09-15T05:10:22.481Z send_ccr_t=false
# CHARGING OVERRIDE DEACTIVATED at 2026-09-15T04:52:09.113Z:
online charging (Gy/OCS) re-enabled node-wide. usage between 2026-
09-15T04:10:22.481Z and 2026-09-15T04:52:09.113Z (2507s) was not
reported to the OCS. original_reason=OCS incident INC-1234
activated_by=noc@example.com deactivated_by=oncall@example.com
```

Les nouvelles lignes intégrées dans la `reason` sont aplaties en espaces. Un commentaire ne peut jamais diviser un enregistrement CDR. Voir Format CDR.

Métriques

Voir Métriques. Alertez sur `charging_override_active` afin qu'un opérateur soit informé pendant que l'override est actif.

Panneau Web

La page des sessions PGW affiche une bannière pendant que l'override est actif. La bannière contient la raison, l'acteur et le temps restant.

Configuration

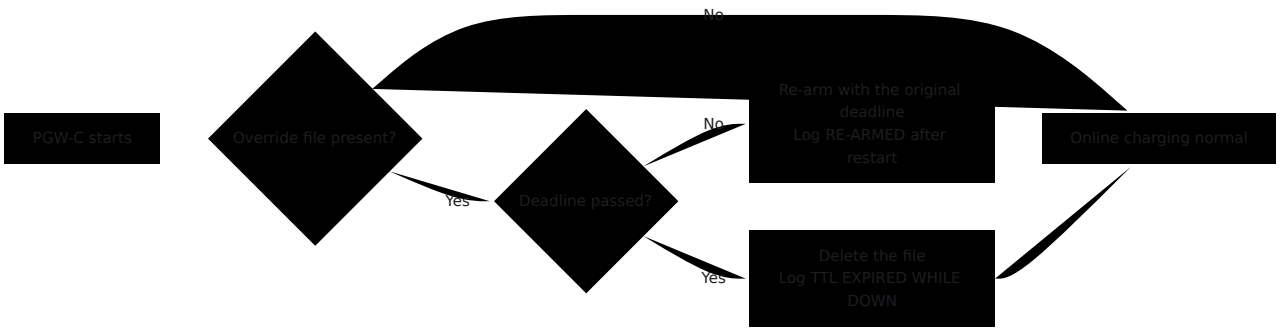
L'override n'a pas besoin de configuration pour fonctionner. Une clé optionnelle contrôle où l'état armé est écrit afin qu'il survive à un redémarrage.

```
config :pgw_c,
  # Où l'override armé est persistant à travers un redémarrage
  charging_override_file: "/var/lib/pgw_c/charging_override.json"
```

Paramètre	Type	Requis	Par défaut
`charging_override_file`	Chaîne	Non	`/var/lib/pgw_c/charging_o`

Comportement de redémarrage

Un crash pendant un incident ne doit pas réarmer silencieusement la coupure pour chaque abonné. Le PGW-C écrit l'override armé dans `charging_override_file` et le lit au démarrage.



Un redémarrage n'��tend pas le TTL. Le PGW-C conserve le `expires_at` original.

Si le PGW-C ne peut pas écrire le fichier, il enregistre un avertissement et continue. L'override est en vigueur sur le nœud, mais il ne survit pas à un redémarrage. Vérifiez les permissions du système de fichiers sur le répertoire.

Métriques

Métrique: `charging_override_active` **Type**: Jauge **Description**: `1` tant que l'override au niveau du nœud désactive la facturation en ligne, `0` sinon. **Étiquettes**: aucune

Métrique: `charging_override_remaining_seconds` **Type**: Jauge **Description**: Secondes restantes avant que l'override expire automatiquement. `0` lorsqu'aucun override n'est actif. **Étiquettes**: aucune

Les deux jauges sont actualisées à chaque intervalle de sondage des métriques. Voir Surveillance & Métriques pour l'intervalle de sondage et la configuration de l'exportateur.

Exemples de requêtes:

```
# Tout nœud avec la facturation en ligne contournée en ce moment
max(charging_override_active)

# Nœuds où l'override expire dans moins de 10 minutes
charging_override_remaining_seconds > 0 and
charging_override_remaining_seconds < 600
```

Exemple d'alerte:

```
- alert: OnlineChargingOverrideActive
  expr: charging_override_active == 1
  for: 5m
  labels:
    severity: critical
  annotations:
    summary: "La facturation en ligne est contournée sur {{
$labels.instance }}"
    description: "L'utilisation des abonnés est mesurée pour les
CDR mais n'est pas facturée en ligne."
```

Dépannage

La requête d'activation retourne 400

Symptômes: Le corps de la réponse mentionne `ttl_seconds` ou `reason`.

Causes possibles:

- `ttl_seconds` est absent, `0`, négatif, ou supérieur à `86400`
- `reason` est absent ou vide

Résolution:

1. Définissez `ttl_seconds` sur un entier entre `1` et `86400`.
2. Définissez `reason` sur un texte qui nomme l'incident.

Le balayage de secours signale des échecs

Symptômes: `relief.failed` est supérieur à `0` dans la réponse d'activation.

Causes possibles:

- Un processus de session n'a pas répondu dans le délai d'appel

- Le UPF n'a pas accepté la modification de session PFCP

Une session qui s'est déconnectée pendant que le balayage était en cours n'est pas comptée comme un échec. Elle n'impose plus rien, donc il n'y a rien à libérer.

Résolution:

1. Recherchez dans le journal `[CHARGING OVERRIDE] failed to release charging enforcement`.
2. Vérifiez la connectivité UPF avec `GET /api/upf`.
3. Exécutez à nouveau `PUT /api/charging/override` une fois que le UPF est sain. Le balayage peut être répété en toute sécurité. Une session qui n'est pas bloquée n'est pas touchée.

Un abonné est toujours bloqué après le balayage

Symptômes: `GET /api/charging/<imsi>` montre que le jardin clos est libéré, mais le trafic ne circule pas.

Causes possibles:

- L'OCS a envoyé une action de dernière unité de TERMINER lors de la connexion. Les paramètres de transfert sur ce porteur ont été effacés et ne peuvent pas être reconstruits.

Résolution:

1. Recherchez dans le journal `Cannot un-gate FAR(s)` avec cet IMSI.
2. Demandez à l'abonné de se reconnecter. Une nouvelle connexion se fait avec le transfert de trafic pendant que l'override est actif.

L'override est revenu après un redémarrage

Symptômes: Le journal contient `[CHARGING OVERRIDE] RE-ARMED after restart`.

C'est le comportement prévu. L'override était encore dans son TTL lorsque le nœud a redémarré.

Résolution:

1. Exécutez `DELETE /api/charging/override` si vous ne le souhaitez plus.
2. Sinon, attendez que le TTL expire.

La facturation en ligne est désactivée et aucun override n'est actif

Symptômes: Aucun CCR-Initial n'est envoyé, et `GET /api/charging/override` signale `active: false`.

Causes possibles:

- `charging.online` est défini sur `:off` dans la configuration d'exécution
- `gy.enabled` est `false`
- Le PCRF a défini l'AVP Gx CCA `Online` sur désactivé

Résolution:

1. Vérifiez les blocs `charging` et `gy`. Voir Référence de configuration.
2. Vérifiez le Gx CCA avec `GET /api/charging/<imsi>`. Voir Interface Diameter Gx.

Documentation Connexe

- Interface Diameter Gy - facturation en ligne, quota et jardin clos
- Format CDR - facturation hors ligne et fichier CDR
- Surveillance & Métriques - exportateur de métriques et tableaux de bord
- Dépannage - procédures générales de dépannage
- Guide des opérations - point d'entrée

Guide de Configuration d'OmniPGW

Référence Complète pour la Configuration de runtime.exs

par Omnitouch Network Services

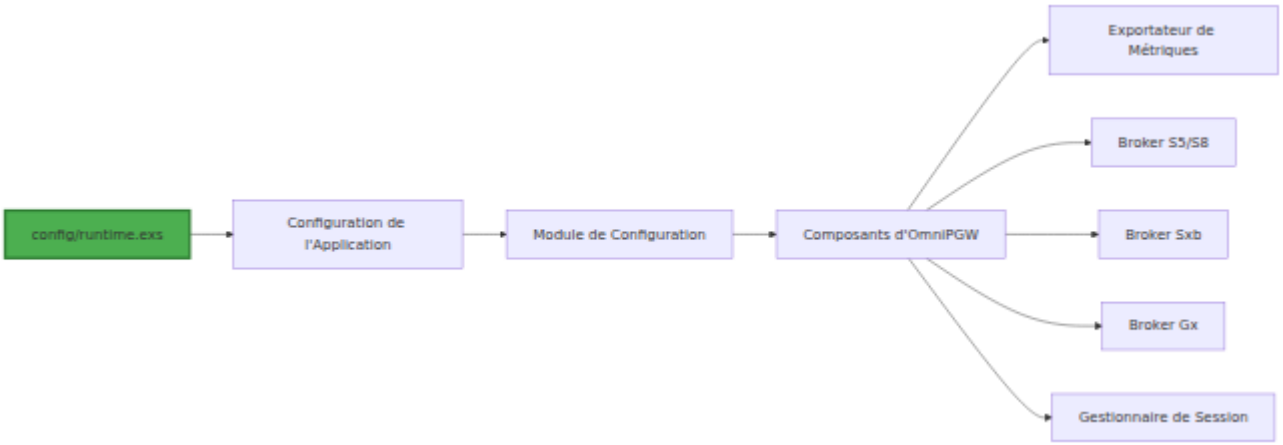
Table des Matières

1. Aperçu
2. Structure du Fichier de Configuration
3. Configuration des Métriques
4. Configuration Diameter/Gx
5. Configuration S5/S8
6. Configuration Gn/Gp (GGSN)
7. Configuration Sxb/PFCP
    - Stratégies de Sélection UPF
    - Équilibrage de Charge avec des Pools UPF
    - Sélection Basée sur DNS
    - Mode Dry-Run
8. Configuration de Pool IP UE
9. Configuration PCO
10. Configuration API OAM
11. Exemple Complet
12. Validation de Configuration

Aperçu

OmniPGW utilise une **configuration runtime** définie dans `config/runtime.exs`. Ce fichier est évalué au **démarrage de l'application** et permet une configuration dynamique basée sur des variables d'environnement ou des sources externes.

Philosophie de Configuration



Principes Clés :

- **Source Unique de Vérité** - Toute la configuration dans un seul fichier
- **Sécurité de Type** - Configuration validée au démarrage
- **Flexibilité Environnementale** - Support pour dev, test, production
- **Defaults Clairs** - Defaults sensés avec des remplacements explicites

---

Structure du Fichier de Configuration

Emplacement du Fichier

```
pgw_c/
├── config/
│   ├── config.exs         # Configuration de base (importe
runtime.exs)
│   ├── dev.exs            # Configuration spécifique au
développement
│   ├── prod.exs           # Configuration spécifique à la
production
│   └── runtime.exs        # ← Fichier de configuration principal
```

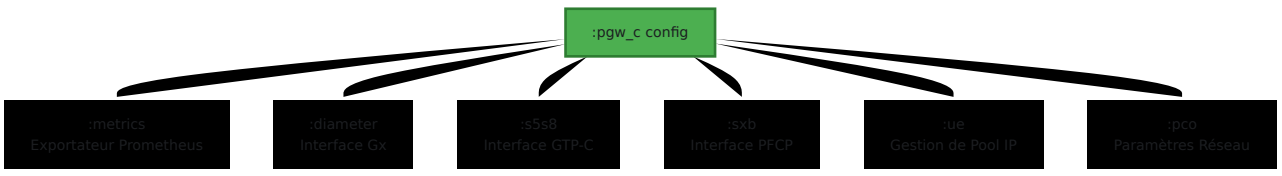
Structure de Niveau Supérieur

```
# config/runtime.exs
import Config

config :logger, level: :info

config :pgw_c,
  metrics: %{...},
  diameter: %{...},
  s5s8: %{...},
  sxb: %{...},
  ue: %{...},
  pco: %{...}
```

Sections de Configuration



Configuration des Métriques

Objectif

Configurer l'exportateur de métriques Prometheus pour surveiller OmniPGW.

Bloc de Configuration

```
config :pgw_c,
  metrics: %{
    # Activer/désactiver l'exportateur de métriques
    enabled: true,

    # Adresse IP pour lier le serveur HTTP
    ip_address: "0.0.0.0",

    # Port pour le point de terminaison des métriques
    port: 9090,

    # Fréquence de sondage des registres (millisecondes)
    registry_poll_period_ms: 10_000
  }
```

Paramètres

Paramètre	Type	Par Défaut	Description
`enabled`	Booléen	`true`	Activer l'exportateur de métriques
`ip_address`	Chaîne (IP)	`"0.0.0.0"`	Adresse de liaison (0.0.0.0 = toutes les interfaces)
`port`	Entier	`9090`	Port HTTP pour le point de terminaison `/metrics`
`registry_poll_period_ms`	Entier	`10_000`	Intervalle de sondage pour les comptes de registre

Exemples

Production - Lier à une IP spécifique :

```
metrics: %{
  enabled: true,
  ip_address: "10.0.0.20",  # Réseau de gestion
  port: 9090,
  registry_poll_period_ms: 5_000  # Sondage toutes les 5 secondes
}
```

Développement - Localhost uniquement :

```
metrics: %{
  enabled: true,
  ip_address: "127.0.0.1",
  port: 42069,  # Port non standard
  registry_poll_period_ms: 10_000
}
```

Désactiver les métriques :

```
metrics: %{
  enabled: false
}
```

Accéder aux Métriques

```
# Point de terminaison par défaut
curl http://<ip_address>:<port>/metrics

# Exemple
curl http://10.0.0.20:9090/metrics
```

Voir : Guide de Surveillance & Métriques pour une documentation détaillée sur les métriques.

Configuration Diameter/Gx

Objectif

Configurer le protocole Diameter pour l'interface Gx (communication PCRF).

Bloc de Configuration

```
config :pgw_c,
  diameter: %{
    # Adresse IP pour écouter les connexions Diameter
    listen_ip: "0.0.0.0",

    # Identité Diameter d'OmniPGW (Origin-Host)
    host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",

    # Domaine Diameter d'OmniPGW (Origin-Realm)
    realm: "epc.mnc001.mcc001.3gppnetwork.org",

    # Liste des pairs PCRF
    peer_list: [
      %{
        # Identité Diameter PCRF
        host: "pcrf.epc.mnc001.mcc001.3gppnetwork.org",

        # Domaine PCRF
        realm: "epc.mnc001.mcc001.3gppnetwork.org",

        # Adresse IP PCRF
        ip: "10.0.0.30",

        # Initier la connexion au PCRF
        initiate_connection: true
      }
    ]
  }
```

Paramètres

Paramètre	Type	Requis	Description
`listen_ip`	Chaîne (IP)	Oui	Adresse d'écoute Diameter
`host`	Chaîne (FQDN)	Oui	Origin-Host d'OmniPGW (doit être FQDN)
`realm`	Chaîne (Domaine)	Oui	Origin-Realm d'OmniPGW
`peer_list`	Liste	Oui	Configurations des pairs PCRF

Configuration du Pair :

Paramètre	Type	Requis	Description
`host`	Chaîne (FQDN)	Oui	Identité Diameter PCRF
`realm`	Chaîne (Domaine)	Oui	Domaine PCRF
`ip`	Chaîne (IP)	Oui	Adresse IP PCRF
`initiate_connection`	Booléen	Oui	Si OmniPGW se connecte au PCRF

Format FQDN

Les identités Diameter DOIVENT être des FQDN :

```
# CORRECT
host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org"

# INCORRECT
host: "omnipgw"              # Pas un FQDN
host: "10.0.0.20"            # IP non autorisée
```

Format 3GPP :

```
<hostname>.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org

Exemples :
- omnipgw.epc.mnc001.mcc001.3gppnetwork.org  (MCC=001, MNC=001)
- pgw-c.epc.mnc260.mcc310.3gppnetwork.org   (MCC=310, MNC=260 - T-
Mobile US)
```

Exemples

Un seul PCRF :

```
diameter: %{
  listen_ip: "0.0.0.0",
  host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
  realm: "epc.mnc001.mcc001.3gppnetwork.org",
  peer_list: [
    %{
      host: "pcrf.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.0.30",
      initiate_connection: true
    }
  ]
}
```

Plusieurs PCRFs (Redondance) :

```
diameter: %{
  listen_ip: "0.0.0.0",
  host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
  realm: "epc.mnc001.mcc001.3gppnetwork.org",
  peer_list: [
    %{
      host: "pcrf-primary.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.1.30",
      initiate_connection: true
    },
    %{
      host: "pcrf-backup.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.2.30",
      initiate_connection: true
    }
  ]
}
```

Connexion Initiée par le PCRF :

```
diameter: %{
  listen_ip: "0.0.0.0",
  host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
  realm: "epc.mnc001.mcc001.3gppnetwork.org",
  peer_list: [
    %{
      host: "pcrf.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.0.0.30",
      initiate_connection: false  # Attendre que le PCRF se
connecte
    }
  ]
}
```

Voir : Documentation de l'Interface Diameter Gx

Configuration S5/S8

Objectif

Configurer l'interface GTP-C pour la communication avec SGW-C.

Bloc de Configuration

```
config :pgw_c,
  s5s8: %{
    # Adresse IPv4 locale pour l'interface S5/S8
    local_ipv4_address: "10.0.0.20",

    # Optionnel : Adresse IPv6 locale
    local_ipv6_address: nil,

    # Optionnel : Remplacer le port GTP-C par défaut (2123)
    local_port: 2123,

    # Délai d'attente des requêtes GTP-C en millisecondes (par
défaut : 500ms)
    # Délai d'attente par tentative lors de l'attente des réponses
GTP-C
    request_timeout_ms: 500,

    # Nombre de tentatives de réessai pour les requêtes GTP-C (par
défaut : 3)
    # Temps d'attente maximum total = request_timeout_ms *
request_attempts
    request_attempts: 3
  }
```

Paramètres

Paramètre	Type	Par Défaut	Description
`local_ipv4_address`	Chaîne (IPv4)	Requis	Adresse IPv4 de l'interface S5/S8
`local_ipv6_address`	Chaîne (IPv6)	`nil`	Adresse IPv6 de l'interface S5/S8 (optionnelle)
`local_port`	Entier	`2123`	Port UDP pour GTP-C (port standard 2123)
`request_timeout_ms`	Entier	`500`	Délai d'attente par tentative de réessai en millisecondes
`request_attempts`	Entier	`3`	Nombre de tentatives de réessai avant d'abandonner

Détails du Protocole

- **Protocole :** GTP-C Version 2
- **Transport :** UDP
- **Port Standard :** 2123
- **Direction :** Reçoit de SGW-C

Exemples

IPv4 Seulement (Commun) :

```
s5s8: %{
  local_ipv4_address: "10.0.0.20"
}
```

IPv4 + IPv6 Dual-Stack :

```
s5s8: %{
  local_ipv4_address: "10.0.0.20",
  local_ipv6_address: "2001:db8::20"
}
```

Port Personnalisé (Non-Standard) :

```
s5s8: %{
  local_ipv4_address: "10.0.0.20",
  local_port: 2124  # Port personnalisé
}
```

Réseau à Latence Élevée :

```
s5s8: %{
  local_ipv4_address: "10.0.0.20",
  request_timeout_ms: 1500,  # 1,5 secondes par tentative
  request_attempts: 3         # Total : 4,5 secondes max
}
```

Configuration des Délais d'Attente

L'interface S5/S8 utilise des délais d'attente configurables pour les transactions de requêtes/réponses GTP-C (Create Bearer Request, Delete Bearer Request).

Calcul du Temps d'Attente Total :

```
Temps d'Attente Maximum Total = request_timeout_ms ×
request_attempts
Par Défaut : 500ms × 3 = 1,5 secondes
```

Directives de Réglage :

Latence Réseau	Délai d'Attente Recommandé	Temps d'Attente Total
Faible latence (<50ms)	200-300ms	600-900ms
Normal (50-150ms)	500ms (par défaut)	1,5s
Haute latence (>150ms)	1000-2000ms	3-6s
Satellite/instable	2000-3000ms	6-9s

Quand Ajuster :

- **Augmenter le délai** si vous voyez fréquemment des erreurs "Create Bearer Request timed out" mais Wireshark montre des réponses arrivant
- **Diminuer le délai** pour une détection d'échec plus rapide dans des environnements à faible latence
- **Augmenter les tentatives de réessai** pour des réseaux peu fiables avec perte de paquets

Comportement des Délais d'Attente :

- En cas de délai d'attente, une erreur est enregistrée : `"Create Bearer Request timed out"`
- Erreur Diameter retournée au PCRF : Result-Code 5012 (UNABLE_TO_COMPLY)
- Le porteur reste en stockage précoce pour nettoyage lorsque Charging-Rule-Remove arrive

Planification Réseau

Sélection d'Adresse IP :

- Utiliser un réseau de gestion/signaling dédié
- Assurer la connectivité depuis tous les nœuds SGW-C

- Considérer la redondance (VRRP/HSRP) pour la HA

Règles de Pare-feu :

```
# Autoriser GTP-C depuis SGW-C
iptables -A INPUT -p udp --dport 2123 -s <sgw_c_network> -j ACCEPT
```

Configuration Gn/Gp (GGSN)

Objectif

Configurer l'interface GTP-C v1 pour la communication avec les SGSNs, permettant la fonctionnalité GGSN pour les réseaux 2G/3G.

Bloc de Configuration

```
config :pgw_c,
  gn: %{
    # Adresse IPv4 locale pour l'interface Gn
    local_ipv4_address: "10.0.0.20",

    # Optionnel : Adresse IPv6 locale (la plupart des réseaux
2G/3G sont uniquement IPv4)
    local_ipv6_address: nil,

    # Port GTP-C (partagé avec S5/S8)
    local_port: 2123
  },

  # Serveurs DNS retournés dans PCO (partagés avec S5/S8)
  dns: %{
    primary_ipv4: {8, 8, 8, 8},
    secondary_ipv4: {8, 8, 4, 4}
  }
```

Paramètres

Paramètre	Type	Par Défaut	Description
`local_ipv4_address`	Chaîne (IPv4)	Requis	Adresse IPv4 de l'interface Gn (adresse GGSN)
`local_ipv6_address`	Chaîne (IPv6)	`nil`	Adresse IPv6 de l'interface Gn (optionnelle, rarement utilisée)
`local_port`	Entier	`2123`	Port UDP pour GTP-C v1 (partagé avec S5/S8)

Paramètres DNS

Paramètre	Type	Par Défaut	Description
`primary_ipv4`	Tuple	`{8, 8, 8, 8}`	Serveur DNS principal pour la réponse PCO
`secondary_ipv4`	Tuple	`{8, 8, 4, 4}`	Serveur DNS secondaire pour la réponse PCO

Détails du Protocole

- **Protocole :** GTP-C Version 1 (3GPP TS 29.060)
- **Transport :** UDP
- **Port Standard :** 2123 (partagé avec GTP-C v2)
- **Direction :** Reçoit des SGSNs

Coexistence avec S5/S8

OmniPGW peut fonctionner à la fois comme PGW (4G) et GGSN (2G/3G) simultanément :

- Les deux interfaces partagent le port UDP 2123
- La version GTP est détectée automatiquement à partir des en-têtes de message
- La même adresse IP peut servir à la fois le trafic 4G et 2G/3G
- Infrastructure partagée : pools IP, UPF/PFCP, facturation en ligne

Voir : Documentation de l'Interface Gn/Gp

Configuration Sxb/PFCP

Objectif

Configurer l'interface PFCP pour la communication avec PGW-U (Plan Utilisateur).

Bloc de Configuration

```
config :pgw_c,
  sxb: %{
    # Adresse IP locale pour la communication PFCP
    local_ip_address: "10.0.0.20",

    # Optionnel : Remplacer le port PFCP par défaut (8805)
    local_port: 8805
  }
```

Paramètres

Paramètre	Type	Par Défaut	Description
`local_ip_address`	Chaîne (IP)	Requis	Adresse d'écoute PFCP
`local_port`	Entier	`8805`	Port UDP PFCP

> **Important :**
>
> - **Tous les pairs UPF sont automatiquement enregistrés** à partir de la configuration `upf_selection` (règles + pool de secours) au démarrage
> - Les UPFs auto-enregistrés utilisent des valeurs par défaut sensées :
>   - Nom généré automatiquement : `"UPF-<ip>:<port>"`
>   - Association PFCP passive (attendre que l'UPF initie)
>   - Intervalle de battement de 5 secondes
> - Les règles de sélection UPF et les pools sont configurés dans la section séparée `upf_selection`. Voir Stratégies de Sélection UPF ci-dessous.
> - L'enregistrement dynamique des UPF est pris en charge pour les UPF découverts par DNS qui ne sont pas dans la configuration

Exemples

Configuration Minimale :

```
sxb: %{
  local_ip_address: "10.0.0.20"
}

# Tous les UPFs dans upf_selection seront automatiquement
enregistrés avec :
# - Nom généré automatiquement : "UPF-10.0.0.21:8805"
# - Association PFCP passive (attendre que l'UPF se connecte)
# - Intervalle de battement de 5 secondes
```

Port PFCP Personnalisé :

```
sxb: %{
  local_ip_address: "10.0.0.20",
  local_port: 8806  # Port PFCP non standard
}
```

Exemple Complet avec Sélection UPF :

```
sxb: %{
  local_ip_address: "10.0.0.20"
},
upf_selection: %{
  rules: [
    %{
      name: "IMS Pool",
      priority: 10,
      match_field: :apn,
      match_regex: ~r/^ims$/,
      upf_pool: [
        %{remote_ip_address: "10.0.1.21", remote_port: 8805,
weight: 100},
        %{remote_ip_address: "10.0.1.22", remote_port: 8805,
weight: 100}
      ]
    }
  ],
  fallback_pool: [
    %{remote_ip_address: "10.0.2.21", remote_port: 8805, weight:
100}
  ]
}
# Tous les 3 UPFs (10.0.1.21, 10.0.1.22, 10.0.2.21) sont
automatiquement enregistrés
```

Sélection Basée sur DNS (Enregistrement Dynamique) :

```
sxb: %{
  local_ip_address: "10.0.0.20"
},
upf_selection: %{
  dns_enabled: true,
  dns_query_priority: [:ecgi, :tai],
  dns_suffix: "epc.3gppnetwork.org",
  fallback_pool: [
    %{remote_ip_address: "10.0.2.21", remote_port: 8805, weight:
100}
  ]
}
# Les UPFs découverts par DNS seront enregistrés dynamiquement
lors de leur première utilisation
```

Stratégies de Sélection UPF

Important : La configuration de sélection UPF a été simplifiée. Tous les pairs UPF sont automatiquement enregistrés à partir de la configuration `upf_selection`.

Structure de Configuration

La sélection UPF est configurée dans la section **`upf_selection`** qui définit :

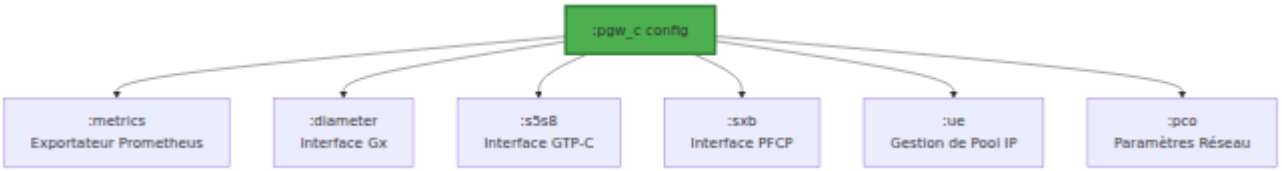
1. **Règles Statique** - Routage basé sur des motifs avec des pools d'équilibrage de charge
2. **Paramètres DNS** - Découverte dynamique UPF basée sur la localisation
3. **Pool de Secours** - Pool par défaut lorsque aucune règle ne correspond et que DNS échoue

Ordre de Priorité de Sélection

1. **Règles Statique** (Priorité la Plus Élevée) - Routage basé sur des motifs avec des pools d'équilibrage de charge

2. **Sélection Basée sur DNS** (Priorité Inférieure) - Découverte dynamique UPF basée sur la localisation
3. **Pool de Secours** (Priorité la Plus Basse) - Pool par défaut lorsque aucune règle ne correspond et que DNS échoue

Flux de Décision de Sélection UPF



Champs de Correspondance Disponibles

Les règles statiques peuvent correspondre à n'importe lequel de ces attributs de session :

Champ de Correspondance	Description	Exemple de Motif
`:imsi`	Identité Internationale de l'Abonné Mobile	`^001001.*` (réseau de test)
`:apn`	Nom de Point d'Accès / DNN	`^internet\.` ou `^ims\.`
`:serving_network_plmn_id`	Identifiant du réseau de service	`^001001$`
`:sgw_ip_address`	Adresse IP SGW	`^10\.100\..*`
`:uli_tai_plmn_id`	Identifiant PLMN de la Zone de Suivi	`^313.*`
`:uli_ecgi_plmn_id`	Identifiant PLMN de la Cellule E-UTRAN	`^313.*`

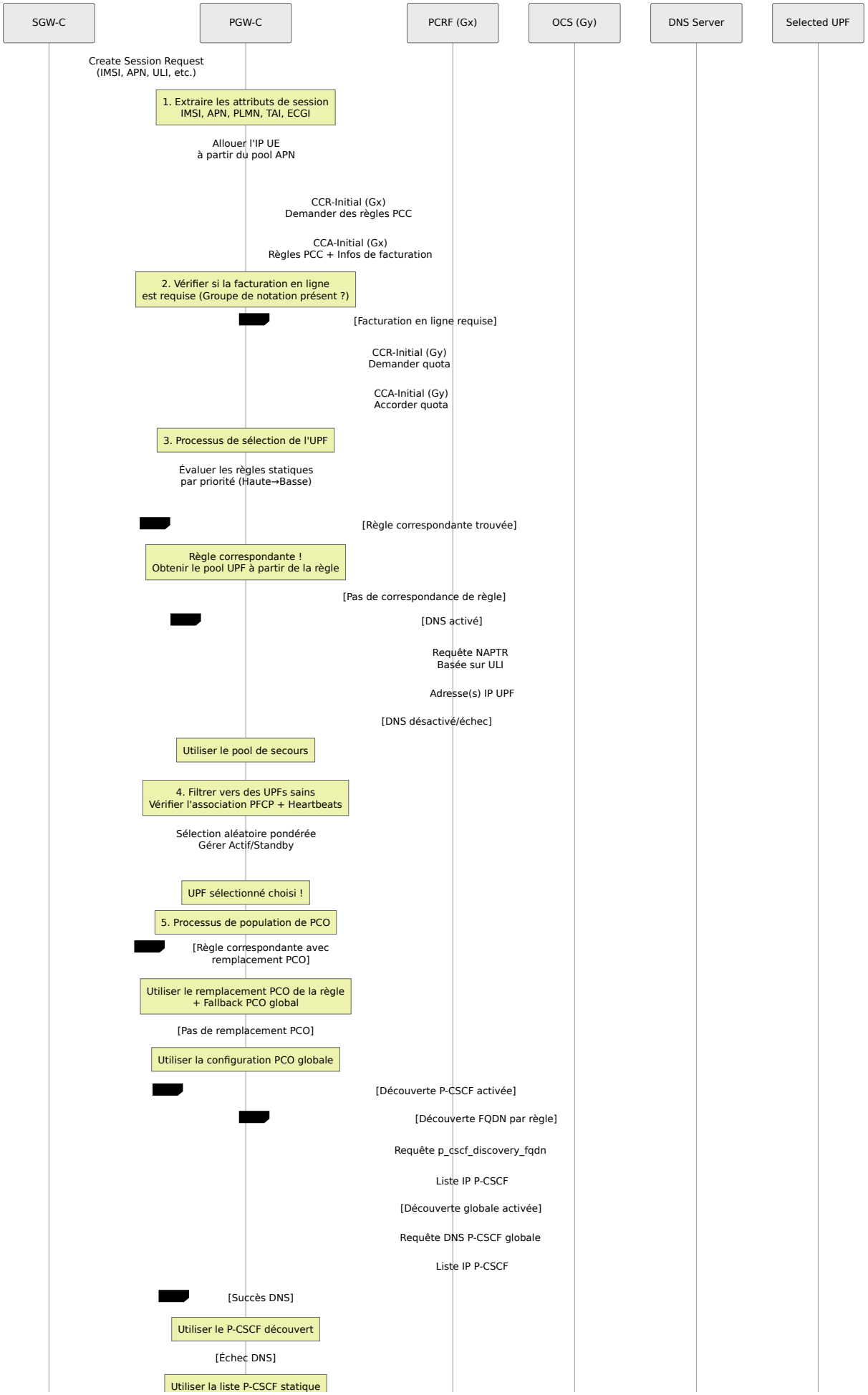
Comparaison des Méthodes de Sélection

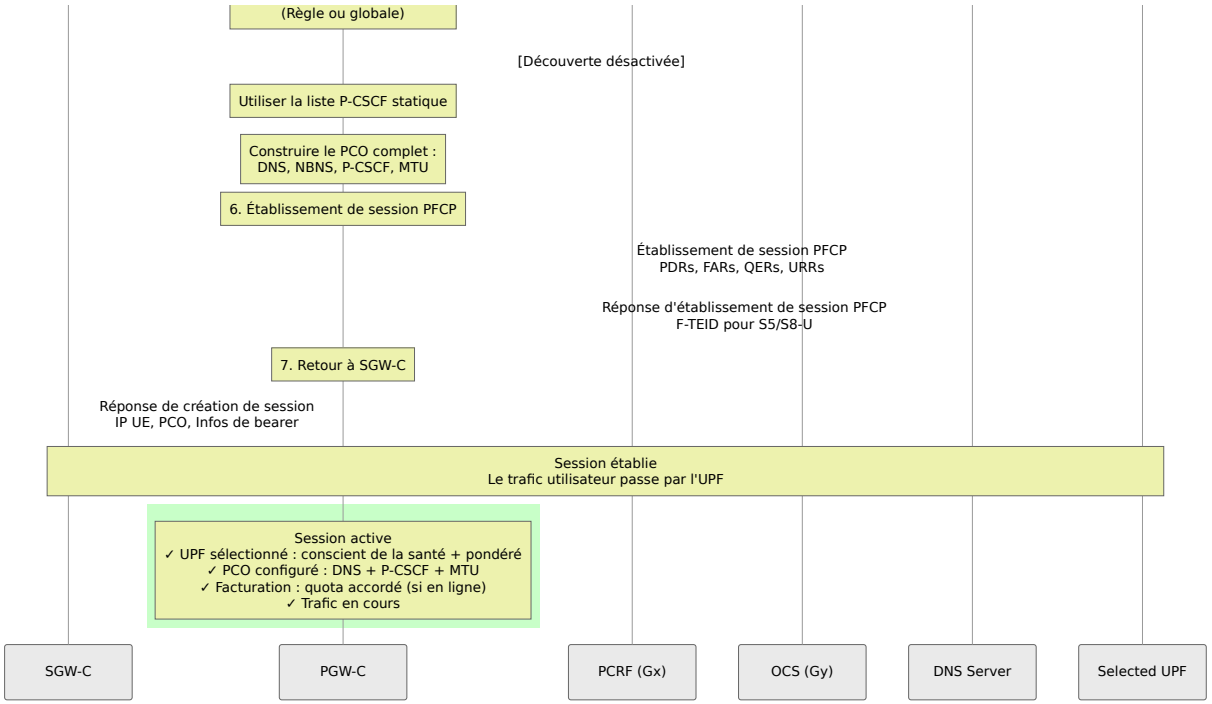
Méthode	Quand Utiliser	Avantages	Inconvénients
Pools UPF	Déploiements en production	Équilibrage de charge, HA, poids flexibles	Nécessite plusieurs UPFs
Basé sur APN	Différenciation de service	Routage IMS/Internet séparément	Configuration statique
Basé sur IMSI	Scénarios de Roaming	Routage géographique	Complexité des regex
Basé sur DNS	MEC/Edge computing	Dynamique, conscient de la localisation	Nécessite une infrastructure DNS
Pool de Secours	Filet de sécurité	Toujours avoir un UPF	Peut ne pas être optimal
Mode Dry-Run	Tester les configs	Test sécurisé	Pas de trafic réel

---

Flux Complet d'Établissement de Session

Ce diagramme montre le flux complet de bout en bout de l'établissement de session, y compris la sélection de l'UPF et la population de PCO :


SGW-C
PGW-C
PCRF (Gx)
OCS (Gy)
DNS Server
Selected UPF
Create Session Request
(IMSI, APN, ULI, etc.)
1. Extraire les attributs de session
IMSI, APN, PLMN, TAI, ECGI
Allouer l'IP UE
à partir du pool APN
CCR-Initial (Gx)
Demander des règles PCC
CCA-Initial (Gx)
Règles PCC + Infos de facturation
2. Vérifier si la facturation en ligne
est requise (Groupe de notation présent ?)
[Facturation en ligne requise]
CCR-Initial (Gy)
Demander quota
CCA-Initial (Gy)
Accorder quota
3. Processus de sélection de l'UPF
Évaluer les règles statiques
par priorité (Haute→Basse)
[Règle correspondante trouvée]
Règle correspondante !
Obtenir le pool UPF à partir de la règle
[Pas de correspondance de règle]
[DNS activé]
Requête NAPTR
Basée sur ULI
Adresse(s) IP UPF
[DNS désactivé/échec]
Utiliser le pool de secours
4. Filtrer vers des UPFs sains
Vérifier l'association PFCP + Heartbeats
Sélection aléatoire pondérée
Gérer Actif/Standby
UPF sélectionné choisi !
5. Processus de population de PCO
[Règle correspondante avec
remplacement PCO]
Utiliser le remplacement PCO de la règle
+ Fallback PCO global
[Pas de remplacement PCO]
Utiliser la configuration PCO globale
[Découverte P-CSCF activée]
[Découverte FQDN par règle]
Requête p_cscf_discovery_fqdn
Liste IP P-CSCF
[Découverte globale activée]
Requête DNS P-CSCF globale
Liste IP P-CSCF
[Succès DNS]
Utiliser le P-CSCF découvert
[Échec DNS]
Utiliser la liste P-CSCF statique



Points de Décision Clés :

1. **Priorité de Sélection UPF :**
   - Règles statiques (correspondance de motif) → Découverte DNS → Pool de secours
   - Filtrage de santé appliqué à toutes les étapes
   - Logique Actif/Standby pour haute disponibilité
   - **Voir :** Interface PFCP pour les détails de communication UPF
2. **Priorité de Population PCO :**
   - Remplacement PCO de la règle → Découverte DNS P-CSCF → Configuration PCO globale
   - Fusion par champ (les règles remplacent des champs spécifiques, le global fournit des valeurs par défaut)
   - **Voir :** Configuration PCO pour les paramètres PCO détaillés
3. **Priorité de Découverte P-CSCF :**
   - FQDN par règle → Découverte DNS globale → PCO statique par règle → PCO statique global
   - **Voir :** Surveillance P-CSCF pour les métriques de découverte et le suivi de la santé

4. **Intégration de la Facturation :**
    - Le PCRF détermine si la facturation en ligne est requise (Groupe de notation + En ligne=1)
    - L'OCS accorde un quota avant l'établissement de session
    - Le PGW-C suit le quota et en demande davantage via CCR-Update
    - **Voir :** Interface Diameter Gx et Interface Diameter Gy pour les détails de facturation

Exemple de Configuration Complète

Voici un exemple complet montrant la sélection UPF multi-pool avec enregistrement automatique des pairs :

```
config :pgw_c,
  # Interface PFCP - Tous les UPFs sont auto-enregistrés à partir
de upf_selection
  sxb: %{
    local_ip_address: "127.0.0.20"
  },

  # Logique de sélection UPF - Tous les UPFs définis ici sont
automatiquement enregistrés
  upf_selection: %{
    # Paramètres de sélection basés sur DNS
    dns_enabled: false,
    dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],
    dns_suffix: "epc.3gppnetwork.org",
    dns_timeout_ms: 5000,

    # Règles de sélection statiques (évaluées par ordre de
priorité)
    rules: [
      # Règle 1 : Trafic IMS - Priorité la plus élevée
      %{
        name: "Trafic IMS",
        priority: 20,
        match_field: :apn,
        match_regex: "^ims",
        upf_pool: [
          %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80},
          %{remote_ip_address: "10.100.2.22", remote_port: 8805,
weight: 20}
        ]
      },

      # Règle 2 : APN d'entreprise
      %{
        name: "Trafic d'entreprise",
        priority: 15,
        match_field: :apn,
        match_regex: "^(enterprise|corporate)\.apn",
        upf_pool: [
          %{remote_ip_address: "10.100.3.21", remote_port: 8805,
weight: 100}
        ]
```

```
      },

      # Règle 3 : Trafic Internet - Équilibré
      %{
        name: "Trafic Internet",
        priority: 5,
        match_field: :apn,
        match_regex: "^internet",
        upf_pool: [
          %{remote_ip_address: "10.100.1.21", remote_port: 8805,
weight: 33},
          %{remote_ip_address: "10.100.1.22", remote_port: 8805,
weight: 33},
          %{remote_ip_address: "10.100.1.23", remote_port: 8805,
weight: 34}
        ]
      }
    ],

    # Pool de secours - Utilisé lorsque aucune règle ne correspond
et que DNS échoue
    fallback_pool: [
      %{remote_ip_address: "127.0.0.21", remote_port: 8805,
weight: 100}
    ]
  }
```

Fonctionnalités Clés

Format Actuel :

- ✅ **Enregistrement Automatique** : Tous les UPFs de `upf_selection` sont automatiquement enregistrés au démarrage
- ✅ **Configuration Centralisée** : Toute la sélection UPF et la configuration des pairs dans une seule section
- ✅ **Pools Requis** : Toutes les règles utilisent le format `upf_pool` (même pour un seul UPF)
- ✅ **Fallback Structuré** : Pool de secours dédié avec distribution pondérée
- ✅ **Intégration DNS** : Paramètres DNS aux côtés des règles de sélection

- **Enregistrement Dynamique** : Les UPFs découverts par DNS sont automatiquement enregistrés lors de leur première utilisation
- **Surveillance de la Santé** : Tous les UPFs configurés sont surveillés avec des heartbeats de 5 secondes

Migration depuis le Format Précédent :

- Supprimé : champ `sxb.peer_list` (plus nécessaire)
- Supprimé : `selection_list` intégré dans les configurations de pairs
- Toutes les définitions UPF vont maintenant dans les règles de `upf_selection` et le pool de secours

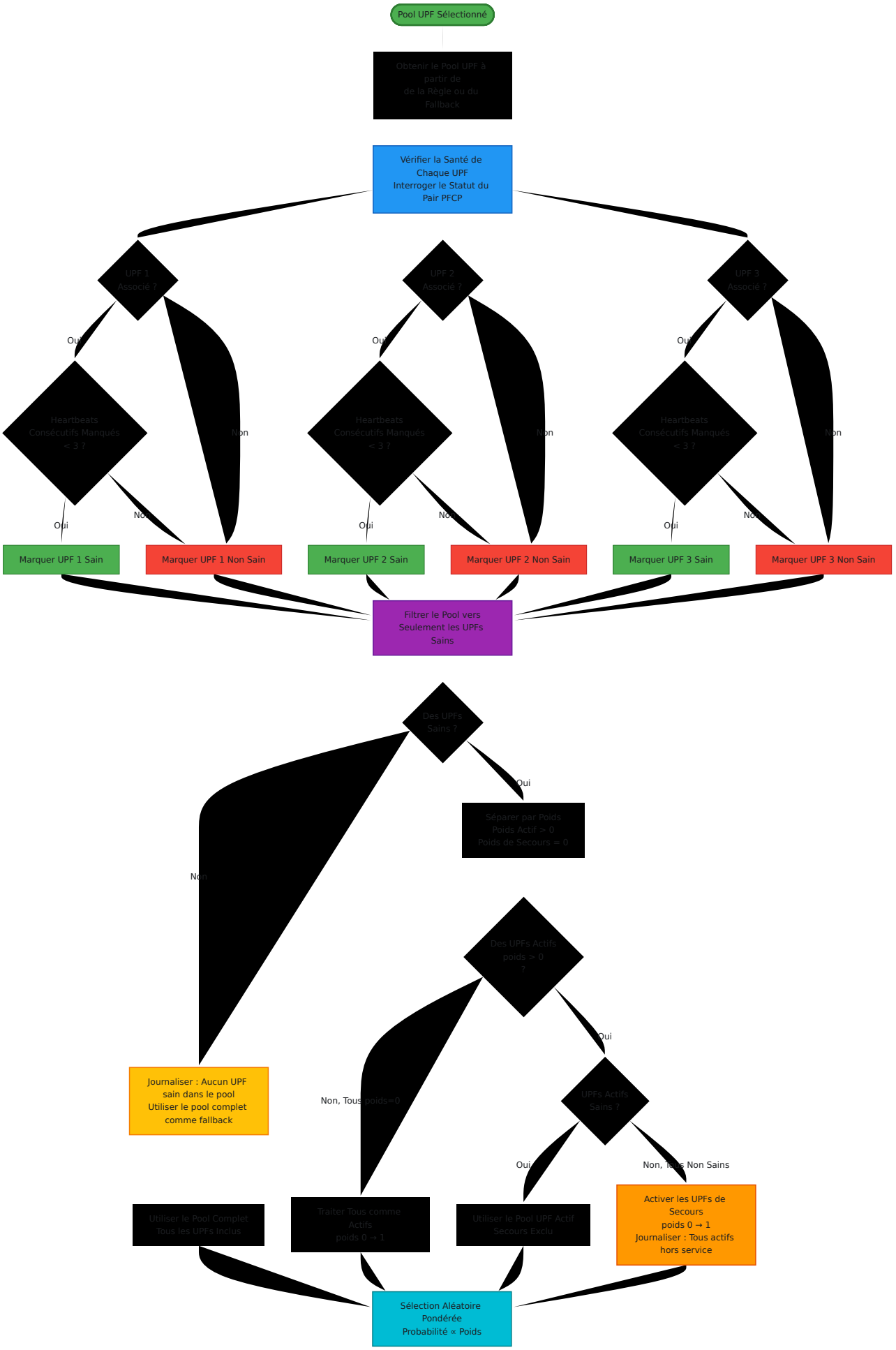
Comment Fonctionnent les Pools UPF :

1. **Sélection Consciente de la Santé** : Seuls les UPFs sains reçoivent du trafic
    - Sain = association PFCP active + moins de 3 heartbeats consécutifs manqués
    - Les UPFs non sains sont automatiquement filtrés
    - Retour à tous les UPFs si aucun n'est sain (échec rapide)
2. **Support Actif/Standby** : Utiliser `weight: 0` pour les UPFs de secours
    - **UPFs Actifs** (poids > 0) : Reçoivent du trafic lorsqu'ils sont sains
    - **UPFs de Secours** (poids == 0) : Reçoivent du trafic uniquement lorsque tous les UPFs actifs sont hors service
    - Les UPFs de secours sont traités comme `weight: 1` lorsqu'ils sont activés
3. **Sélection Aléatoire Pondérée** : Chaque session est assignée aléatoirement à un UPF sain en fonction des poids
    - Dans l'exemple ci-dessus : 70% vont à .21, 20% à .22, 10% à .23
    - Poids plus élevé = plus de sessions assignées à cet UPF
    - Poids égaux = distribution égale
4. **Enregistrement Automatique** : Tous les UPFs dans les pools sont automatiquement enregistrés au démarrage

- Noms auto-générés : `"UPF-<ip>:<port>"`
- Paramètres par défaut : association PFCP passive, heartbeats de 5 secondes
- Suivi immédiat de la santé pour tous les UPFs configurés

Sélection Consciente de la Santé avec

Actif/Standby


Pool UPF Sélectionné
Obtenir le Pool UPF à partir de de la Règle ou du Fallback
Vérifier la Santé de Chaque UPF Interroger le Statut du Pair PFCP
UPF 1 Associé ?
UPF 2 Associé ?
UPF 3 Associé ?
Oui
Non
Heartbeats Consécutifs Manqués < 3 ?
Marquer UPF 1 Sain
Marquer UPF 1 Non Sain
Marquer UPF 2 Sain
Marquer UPF 2 Non Sain
Marquer UPF 3 Sain
Marquer UPF 3 Non Sain
Filtrer le Pool vers Seulement les UPFs Sains
Des UPFs Sains ?
Séparer par Poids Poids Actif > 0 Poids de Secours = 0
Des UPFs Actifs poids > 0 ?
Journaliser : Aucun UPF sain dans le pool Utiliser le pool complet comme fallback
UPFs Actifs Sains ?
Non, Tous poids=0
Non, Tous Non Sains
Utiliser le Pool Complet Tous les UPFs Inclus
Traiter Tous comme Actifs poids 0 → 1
Utiliser le Pool UPF Actif Secours Exclu
Activer les UPFs de Secours poids 0 → 1 Journaliser : Tous actifs hors service
Sélection Aléatoire Pondérée Probabilité ∝ Poids



Exemple de Sélection Aléatoire Pondérée :

```
Pool: [
  UPF-A: poids 50, sain ✓
  UPF-B: poids 30, sain ✓
  UPF-C: poids 20, sain ✓
]

Poids Total: 50 + 30 + 20 = 100

Plages de Poids :
  UPF-A: 0-49   (50%)
  UPF-B: 50-79  (30%)
  UPF-C: 80-99  (20%)

Nombre aléatoire : 63 → Sélectionne UPF-B
Nombre aléatoire : 15 → Sélectionne UPF-A
Nombre aléatoire : 91 → Sélectionne UPF-C
```

Exemple de Basculement Actif/Standby :

```
Pool Initial : [
  UPF-A: poids 100, sain ✓   (Actif)
  UPF-B: poids 0, sain ✓     (Standby)
]

Scénario 1 : UPF-A Sain
→ Utiliser le Pool Actif : [UPF-A: 100]
→ Tout le trafic vers UPF-A

Scénario 2 : UPF-A Échoue
→ Aucun UPF actif sain
→ Activer le Secours : [UPF-B: 1]
→ Tout le trafic bascule vers UPF-B
→ Journaliser : "Tous les UPFs actifs hors service, activation des
UPFs de secours"

Scénario 3 : Les Deux Non Sains
→ Aucun UPF sain
→ Utiliser le pool complet : [UPF-A: 100, UPF-B: 0]
→ Sélectionner avec des poids (essayer la connexion, peut échouer)
→ Journaliser : "Aucun UPF sain dans le pool, utilisation du pool
complet comme fallback"
```

Modèles de Poids Communs :

```
# Distribution égale (25% chacun)
upf_pool: [
  %{remote_ip_address: "10.0.1.1", remote_port: 8805, weight: 1},
  %{remote_ip_address: "10.0.1.2", remote_port: 8805, weight: 1},
  %{remote_ip_address: "10.0.1.3", remote_port: 8805, weight: 1},
  %{remote_ip_address: "10.0.1.4", remote_port: 8805, weight: 1}
]

# Chargement équilibré primaire/de secours (90% / 10%)
upf_pool: [
  %{remote_ip_address: "10.0.1.21", remote_port: 8805, weight:
90},
  %{remote_ip_address: "10.0.1.22", remote_port: 8805, weight: 10}
]

# Actif/Standby (100% primaire, 0% standby jusqu'à ce que le
primaire échoue)
upf_pool: [
  %{remote_ip_address: "10.0.1.21", remote_port: 8805, weight:
100},  # Actif
  %{remote_ip_address: "10.0.1.22", remote_port: 8805, weight: 0}
# Standby (uniquement lorsque actif hors service)
]

# Actif avec plusieurs secours (chargement équilibré
lorsqu'activé)
upf_pool: [
  %{remote_ip_address: "10.0.1.1", remote_port: 8805, weight:
100},   # Actif
  %{remote_ip_address: "10.0.1.2", remote_port: 8805, weight: 0},
# Secours 1
  %{remote_ip_address: "10.0.1.3", remote_port: 8805, weight: 0}
# Secours 2
]
# Résultat : Actif obtient 100%. Si actif échoue, les secours
obtiennent 50/50%.

# Test A/B (50% / 50%)
upf_pool: [
  %{remote_ip_address: "10.0.1.100", remote_port: 8805, weight:
50},  # Ancienne version
  %{remote_ip_address: "10.0.1.200", remote_port: 8805, weight:
```

```
50}   # Nouvelle version
]
```

Cas d'Utilisation :

- **Basculement Actif/Standby** : Utiliser `weight: 0` pour les UPFs de secours chauds qui s'activent uniquement lorsque les primaires échouent
- **HA Consciente de la Santé** : Basculement automatique lorsque les UPFs perdent l'association PFCP ou manquent des heartbeats
- **Mise à l'Échelle Horizontale** : Distribuer la charge entre plusieurs UPFs pour augmenter la capacité
- **Haute Disponibilité** : La distribution automatique empêche la surcharge d'un UPF unique
- **Déploiements Progressifs** : Utiliser des poids pour les déploiements canari (par exemple, 95% ancien, 5% nouveau)
- **Optimisation des Coûts** : Diriger plus de trafic vers des UPFs de plus grande capacité
- **Distribution Géographique** : Équilibrer les sessions entre les UPFs de périphérie

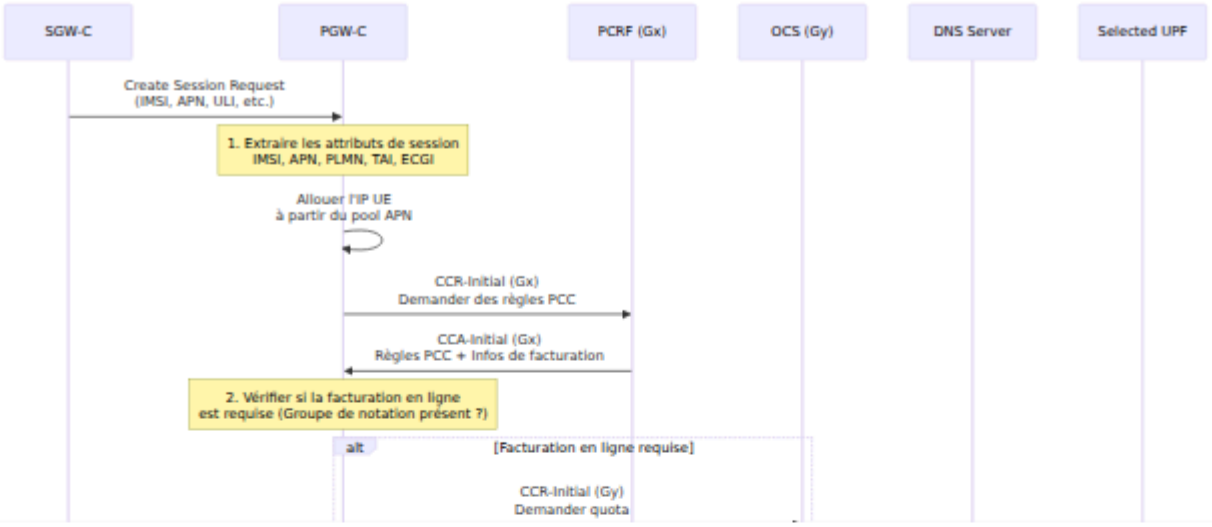
Remplacements PCO (Options de Configuration de Protocole) :

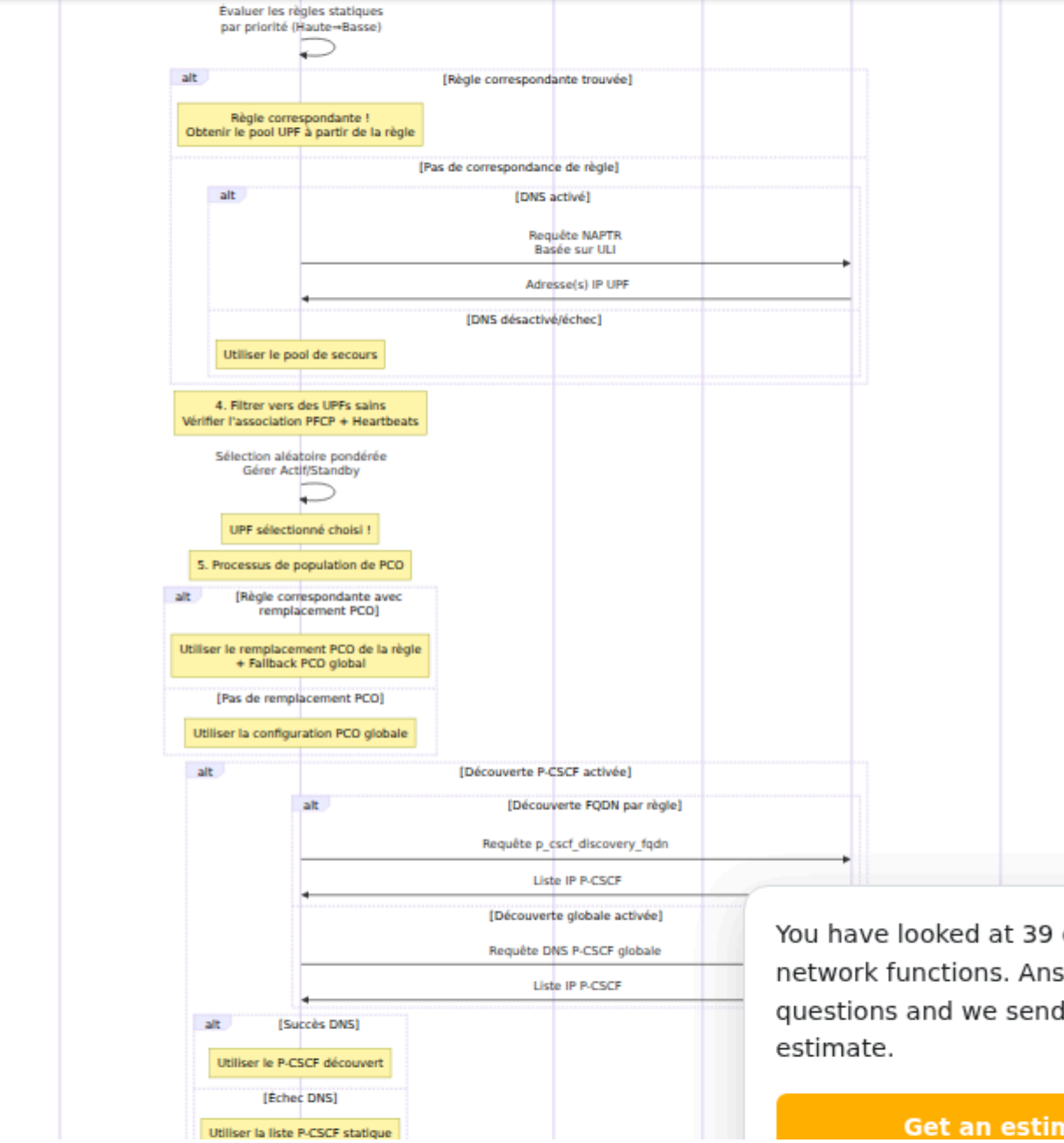
Chaque règle de sélection UPF peut spécifier des valeurs PCO personnalisées qui remplacent la configuration PCO par défaut pour les sessions correspondantes. Cela permet à différents APNs ou types de trafic de recevoir différents paramètres réseau.

Comment Fonctionnent les Remplacements PCO :

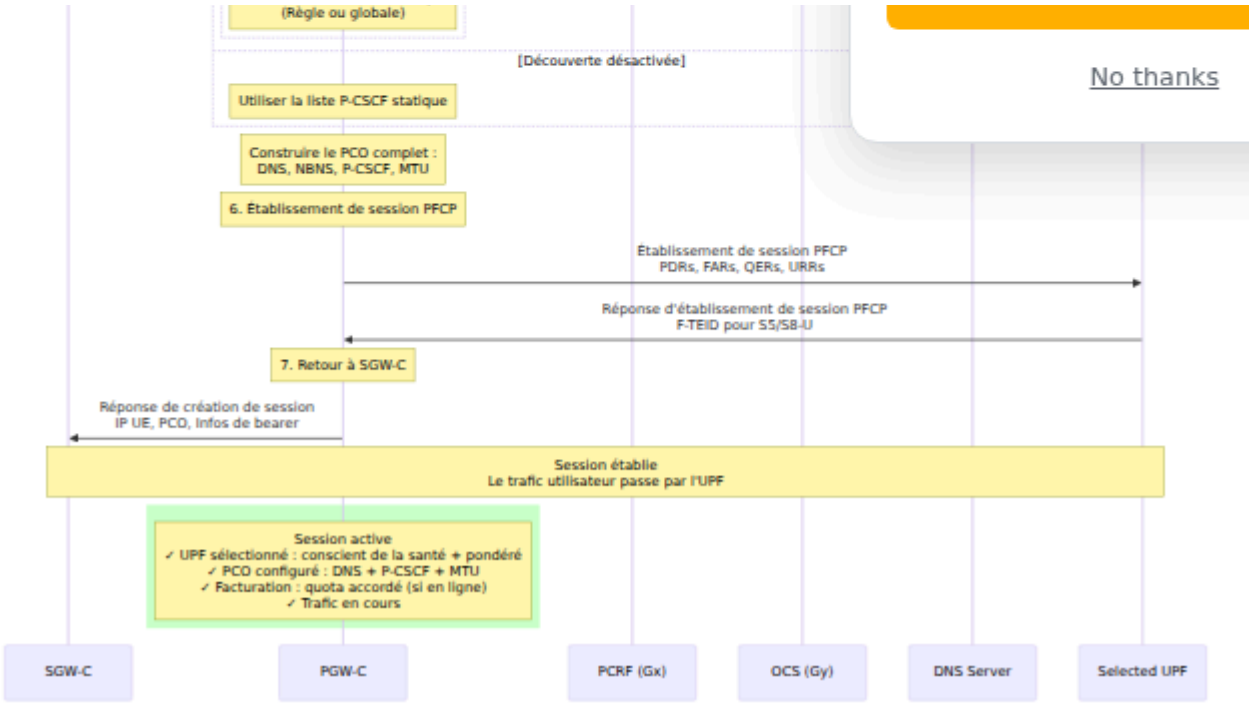
1. **Remplacements Partiels** : Spécifiez uniquement les champs PCO que vous souhaitez remplacer
2. **Fallback par Défaut** : Les champs non spécifiés utilisent les valeurs de la configuration `pco` principale
3. **Spécifique à la Règle** : Chaque règle peut avoir des remplacements PCO différents
4. **Fusion de Priorité** : Le PCO de la règle a la priorité sur le PCO par défaut

Hiérarchie de Population PCO





You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima



Ordre de Priorité pour Chaque Champ PCO :

1. **Remplacement PCO de la Règle** (Priorité la Plus Élevée)
2. **Découverte DNS P-CSCF** (pour les adresses P-CSCF uniquement)
3. **Configuration PCO Globale** (Priorité la Plus Basse / Fallback)

Exemple : La Règle IMS Remplace DNS, La Règle Entreprise Remplace Tout

```
Session IMS (correspond à la règle "Trafic IMS") :
├─ Serveurs DNS : DE LA CONFIGURATION GLOBALE (non remplacé dans
la règle)
���─ P-CSCF : DE LA DÉCOUVERTE DNS (p_cscf_discovery_fqdn défini
dans la règle)
│  └─ Fallback : DE LA RÈGLE si DNS échoue
└─ MTU : DE LA CONFIGURATION GLOBALE (non remplacé dans la règle)

Session Entreprise (correspond à la règle "Trafic d'entreprise") :
├─ Serveurs DNS : DE LA RÈGLE (192.168.1.10, 192.168.1.11)
├─ P-CSCF : DE LA CONFIGURATION GLOBALE (non remplacé dans la
règle)
└─ MTU : DE LA RÈGLE (1500)

Session par Défaut (aucune règle correspondante) :
├─ Serveurs DNS : DE LA CONFIGURATION GLOBALE
├─ P-CSCF : DE LA CONFIGURATION GLOBALE ou DNS si la découverte
globale est activée
└─ MTU : DE LA CONFIGURATION GLOBALE
```

Champs de Remplacement PCO Disponibles :

- `primary_dns_server_address` - Adresse IP du serveur DNS primaire (IPv4)
- `secondary_dns_server_address` - Adresse IP du serveur DNS secondaire (IPv4)
- `primary_dns_server_ipv6_address` - Adresse IP du serveur DNS primaire (IPv6, conteneur 0x0003)
- `secondary_dns_server_ipv6_address` - Adresse IP du serveur DNS secondaire (IPv6, conteneur 0x0003)
- `primary_nbns_server_address` - Adresse IP du serveur WINS primaire
- `secondary_nbns_server_address` - Adresse IP du serveur WINS secondaire
- `p_cscf_ipv4_address_list` - Liste des adresses IP des serveurs P-CSCF IPv4 (pour IMS, conteneur 0x000C) - Voir Configuration PCO et Surveillance P-CSCF pour la découverte dynamique de P-CSCF
- `p_cscf_ipv6_address_list` - Liste des adresses IP des serveurs P-CSCF IPv6 (pour IMS, conteneur 0x0001)
- `ipv4_link_mtu_size` - Taille MTU en octets

Découverte P-CSCF par Règle :

En plus des remplacements PCO, les règles de sélection UPF peuvent spécifier la découverte dynamique de P-CSCF :

- `p_cscf_discovery_fqdn` - (Chaîne) FQDN pour la découverte P-CSCF basée sur DNS (par exemple, `"pcscf.mnc001.mcc001.3gppnetwork.org"`)

Lorsque ce paramètre est défini :

1. PGW-C effectue une recherche DNS pour le FQDN spécifié lors de l'établissement de la session
2. Le serveur DNS renvoie une liste d'adresses IP P-CSCF
3. Les adresses P-CSCF découvertes sont envoyées à l'UE via PCO
4. Si la recherche DNS échoue, il revient à `p_cscf_ipv4_address_list` de l'override PCO (si spécifié) ou de la configuration PCO globale
5. Voir Surveillance P-CSCF pour surveiller les taux de succès/échec de découverte

Ceci est particulièrement utile pour :

- **APN IMS** - Différentes réseaux IMS avec différents serveurs P-CSCF
- **Déploiements multi-locataires** - Différentes entreprises avec une infrastructure P-CSCF dédiée
- **Routage géographique** - Le DNS renvoie le P-CSCF le plus proche en fonction de la localisation de l'UE
- **Haute disponibilité** - Le DNS renvoie automatiquement uniquement les serveurs P-CSCF sains

Exemple : Trafic IMS avec P-CSCF personnalisé :

```
rules: [
  %{
    name: "Trafic IMS",
    priority: 20,
    match_field: :apn,
    match_regex: "^ims",
    upf_pool: [
      %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80},
      %{remote_ip_address: "10.100.2.22", remote_port: 8805,
weight: 20}
    ],
    # Découverte P-CSCF : Interroger dynamiquement le DNS pour les
adresses P-CSCF
    # La recherche DNS renvoie les IPs P-CSCF actuelles basées sur
ce FQDN
    p_cscf_discovery_fqdn: "pcscf.mnc001.mcc001.3gppnetwork.org",
    # Les sessions IMS obtiennent des serveurs P-CSCF
personnalisés (utilisés comme secours si DNS échoue)
    pco: %{
      p_cscf_ipv4_address_list: ["203.0.113.100", "203.0.113.101"]
      # DNS, NBNS, MTU utiliseront les valeurs par défaut de la
configuration PCO principale
    }
  }
]
```

Allocation IP UE par règle :

Chaque règle de sélection UPF choisit **comment l'IP de l'UE est allouée**. Il existe trois scénarios mutuellement cohérents :

Scénario	Configuration de la règle	Qui alloue	Pool utilisé
1. Par défaut	*(aucun champ)*	PGW-C (adresse envoyée au UPF)	`ue.subnet_map` clé APN, revenant à `:default`
2. Pool nommé	`ue_ip_pool: "name"`	PGW-C (adresse envoyée au UPF)	le pool nommé `ue.subnet_map` `"name"`
3. Délégué au UPF	`upf_assigns_ip: true`	UPF (PFCP CHV4)	un pool sur le UPF

- `upf_assigns_ip` - (Booléen, par défaut `false`) Lorsque `true`, le PGW-C envoie l'IE UE-IP-Address avec le **CHV4** ("UP alloue") au lieu d'une adresse explicite, et lit l'adresse allouée depuis le PDR Créé du UPF dans la Réponse d'Établissement de Session. **Quel** pool le UPF utilise est entièrement une préoccupation côté UPF (le UPF peut le lier à l'APN / Instance de Réseau PFCP s'il possède plusieurs pools) — rien concernant le pool n'est signalé depuis le PGW-C.
- `ue_ip_pool` - (Chaîne, optionnel) Nom d'un pool `ue.subnet_map` à partir duquel le **PGW-C** alloue lorsque cette règle est sélectionnée. Cela lie un pool PGW-C à un UPF via la règle de sélection (par exemple, pour que le UPF qui possède/annonce `100.64.20.0/24` ne reçoive jamais d'adresses que de ce pool). Un pool peut être référencé par une ou plusieurs règles. `nil` → le comportement par défaut de `ue.subnet_map` clé APN (Scénario 1).

`upf_assigns_ip` et `ue_ip_pool` sont **mutuellement exclusifs** — l'un délègue l'allocation au UPF, l'autre la garde dans le PGW-C. Si une règle définit les deux, les journaux de chargement de configuration enregistrent un avertissement et `upf_assigns_ip` l'emporte (`ue_ip_pool` est ignoré).

Soit l'allocation PGW-C à partir d'un pool lié au UPF (Scénario 2) *ou* la délégation UPF (Scénario 3) permet à chaque UPF de posséder un sous-réseau

et d'annoncer un **agrégat** OSPF unique (par exemple, `100.64.20.0/24`) au lieu d'un `/32` par UE par attachement — l'agrégat est présent depuis le démarrage du UPF, donc le downlink d'un UE nouvellement attaché est routable immédiatement, supprimant la fenêtre de convergence de route par UE.

> **Important — alignement pool/UPF :** le pool à partir duquel un UE est alloué doit se situer dans l'agrégat annoncé du UPF qui le transporte. Pour le Scénario 2, liez chaque pool par UPF à une règle dont le `upf_pool` est ce UPF unique. Pour le Scénario 3, chaque UPF `upf_pool` doit être configuré avec son propre `ueip_pool` non chevauchant (omniupf `feature_ueip: true`) annonçant cet agrégat (`route_advertise_pools`). Si un UE atterrit sur un UPF mais que son adresse se trouve à l'intérieur de l'agrégat d'un autre UPF, les downlinks sont bloqués.

Exemple — Scénario 2 (PGW-C alloue à partir de pools par UPF liés par la règle) :

```
# ue.subnet_map définit les pools nommés :
ue: %{
  subnet_map: %{
    "ims_upf03" => ["100.64.20.0/24"],
    "ims_upf04" => ["100.64.21.0/24"],
    default:     ["100.64.16.0/24"]
  }
},
# chaque règle fixe un UPF et son pool :
upf_selection: %{
  rules: [
    %{name: "IMS→upf03", priority: 21, match_field: :apn,
match_regex: "^ims",
      upf_pool: [%{remote_ip_address: "10.179.1.42", remote_port:
8805, weight: 1}],
      ue_ip_pool: "ims_upf03"},
    %{name: "IMS→upf04", priority: 20, match_field: :apn,
match_regex: "^ims",
      upf_pool: [%{remote_ip_address: "10.179.1.45", remote_port:
8805, weight: 1}],
      ue_ip_pool: "ims_upf04"}
  ]
}
```

Exemple — Scénario 3 (UPF alloue l'IP UE via CHV4) :

```
rules: [
  %{
    name: "IMS",
    priority: 20,
    match_field: :apn,
    match_regex: "^ims",
    upf_pool: [
      %{remote_ip_address: "10.179.1.42", remote_port: 8805,
weight: 50},
      %{remote_ip_address: "10.179.1.45", remote_port: 8805,
weight: 50}
    ],
    upf_assigns_ip: true   # UPF alloue l'IP UE (CHV4) ; le pool
est côté UPF
  }
]
```

Exemple : Trafic d'entreprise avec DNS personnalisé :

```
rules: [
  %{
    name: "Trafic d'entreprise",
    priority: 15,
    match_field: :apn,
    match_regex: "^(enterprise|corporate)\.apn",
    upf_pool: [
      %{remote_ip_address: "10.100.3.21", remote_port: 8805,
weight: 100}
    ],
    # Les sessions d'entreprise obtiennent DNS d'entreprise et MTU
personnalisé
    pco: %{
      primary_dns_server_address: "192.168.1.10",
      secondary_dns_server_address: "192.168.1.11",
      ipv4_link_mtu_size: 1500
      # P-CSCF, NBNS utiliseront les valeurs par défaut de la
configuration PCO principale
    }
  }
]
```

Exemple : Override complet (Tous les champs PCO) :

```
rules: [
  %{
    name: "IoT APN - Entièrement personnalisé",
    priority: 10,
    match_field: :apn,
    match_regex: "^iot\.m2m",
    upf_pool: [
      %{remote_ip_address: "10.100.5.21", remote_port: 8805,
weight: 100}
    ],
    # Les sessions IoT obtiennent un PCO complètement personnalisé
    pco: %{
      primary_dns_server_address: "8.8.8.8",
      secondary_dns_server_address: "8.8.4.4",
      primary_nbns_server_address: "10.0.0.100",
      secondary_nbns_server_address: "10.0.0.101",
      p_cscf_ipv4_address_list: [],  # Pas de P-CSCF pour IoT
      ipv4_link_mtu_size: 1280       # MTU plus petit pour les
dispositifs contraints
    }
  }
]
```

Cas d'utilisation :

- **IMS/VoLTE** : Fournir des serveurs P-CSCF spécifiques à l'opérateur pour les services vocaux
- **APN d'entreprise** : Router le trafic d'entreprise via les serveurs DNS de l'entreprise
- **IoT/M2M** : Utiliser DNS public et MTU optimisé pour les dispositifs à faible bande passante
- **Roaming** : Fournir des serveurs DNS locaux pour les abonnés en visite
- **Différenciation de service** : Différents paramètres réseau par type de service

Sélection UPF basée sur DNS :

Activer la sélection dynamique UPF basée sur les informations de localisation de l'utilisateur (ULI) en utilisant des requêtes DNS NAPTR. Les paramètres DNS sont maintenant configurés dans la section `upf_selection`.

Remarque : Cela fournit une sélection UPF basée sur la géographie ou la topologie. Voir Interface PFCP pour la configuration d'association PFCP avec des UPF découverts dynamiquement et Gestion de session pour les flux d'établissement de session.

```
upf_selection: %{
  # Activer la sélection basée sur DNS
  dns_enabled: true,

  # Types de localisation à interroger par ordre de priorité
  dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],

  # Suffixe DNS pour les requêtes NAPTR 3GPP
  dns_suffix: "epc.3gppnetwork.org",

  # Délai d'attente de la requête DNS en millisecondes
  dns_timeout_ms: 5000,

  # ... règles et fallback_pool ...
}
```

La sélection basée sur DNS fonctionne comme suit :

1. **Priorité** : La sélection DNS n'est utilisée que lorsque **AUCUNE règle statique ne correspond** (priorité inférieure)
2. **Génération de requêtes** : Construit des requêtes DNS NAPTR basées sur la localisation de l'UE :
   - Requête ECGI : `eci-<hex>.ecgi.epc.mnc<MNC>.mcc<MCC>.epc.3gppnetwork.org`
   - Requête TAI : `tac-lb<hex>.tac-hb<hex>.tac.epc.mnc<MNC>.mcc<MCC>.epc.3gppnetwork.org`
   - Les requêtes RAI, SAI, CGI suivent un format similaire à la norme 3GPP TS 23.003
3. **Hiérarchie de secours** : Essaye chaque type de localisation par ordre de priorité jusqu'à ce qu'une correspondance soit trouvée
4. **Correspondance des pairs** : Les résultats DNS sont filtrés contre la liste de pairs configurée

5. **Sélection** : Choisit le pair correspondant (actuellement première correspondance, sélection basée sur la charge à venir)

Exemple d'enregistrements DNS (configurer sur votre serveur DNS) :

```
; Enregistrement NAPTR pour TAC 100 dans PLMN 001-001
tac-lb64.tac-hb00.tac.epc.mnc001.mcc001.epc.3gppnetwork.org IN
NAPTR 10 50 "a" "x-3gpp-upf:x-sxb" "" upf-edge-1.example.com.

; Enregistrement A pour le UPF
upf-edge-1.example.com IN A 10.100.1.21
```

Cas d'utilisation :

- **Edge Computing multi-accès (MEC)** : Router les sessions vers les UPF de bord les plus proches géographiquement
- **Découverte dynamique UPF** : Ajouter/retirer des UPF sans reconfigurer PGW-C
- **Équilibrage de charge** : Distribuer la charge entre les UPF en fonction de la localisation
- **Tranchage de réseau** : Router différentes tranches vers différents UPF par localisation

Surveillance de la santé UPF

Sélection automatique consciente de la santé : Le PGW-C surveille en continu la santé de tous les UPF et exclut automatiquement les UPF non sains de la sélection.

Critères de vérification de la santé

Un UPF est considéré comme **sain** lorsque TOUTES les conditions suivantes sont remplies :

1. **Association PFCP active** : Le UPF a une association PFCP établie
2. **Réactivité du cœur** : Moins de 3 battements de cœur consécutifs manqués
3. **Processus vivant** : Le processus GenServer pair UPF fonctionne

Un UPF est considéré comme **non sain** si L'UN DES éléments suivants est vrai :

- L'association PFCP n'est pas établie (`associated: false`)
- 3 ou plus de délais d'attente consécutifs de battement de cœur
- Le processus pair UPF a échoué ou est non réactif

Mécanisme de surveillance

Pour les UPF configurés (dans `upf_selection`) :

- Le suivi de la santé commence immédiatement au démarrage
- L'association PFCP est surveillée en continu
- Les battements de cœur sont envoyés toutes les 5 secondes
- Le compteur `missed_heartbeats_consecutive` suit les échecs consécutifs
- Tous les UPF des règles et du pool de secours sont automatiquement enregistrés

Pour les UPF découverts par DNS (enregistrement dynamique) :

- Considéré comme sain jusqu'à la première tentative de session
- Enregistré automatiquement lors de la premi��re utilisation
- Le suivi de la santé commence après l'enregistrement

Comportement de sélection

Mode Actif/Standby (lors de l'utilisation de `weight: 0`) :

1. Filtrer uniquement les UPF sains
2. Séparer en **actif** (poids > 0) et **standby** (poids == 0)
3. Utiliser les UPF actifs si l'un d'eux est sain
4. Activer les UPF de secours (traiter comme poids 1) si tous les actifs sont non sains
5. Revenir au pool complet si aucun UPF sain n'existe

Mode Équilibré par charge (tous poids > 0) :

1. Filtrer uniquement les UPF sains
2. Effectuer une sélection aléatoire pondérée parmi les UPF sains

3. Revenir au pool complet si aucun UPF sain n'existe

Journalisation :

```
[debug] Utilisation du pool UPF actif (2/3 UPF sains, 1 standby)
[info] Tous les UPF actifs sont hors service, activation des UPF
de secours (1 UPF standby, traitement du poids 0 comme 1)
[warning] Aucun UPF sain dans le pool (3 au total), utilisation du
pool complet comme secours
```

Vérification de la santé UPF

Programmatique :

```
# Vérifier si un UPF spécifique est sain
iex> PGW_C.PFCP_Node.is_peer_healthy?({10, 100, 1, 21})
true

# Obtenir des informations détaillées sur la santé
iex> PGW_C.PFCP_Node.get_peer_health({10, 100, 1, 21})
%{
  associated: true,
  missed_heartbeats: 0,
  healthy: true,
  registered: true
}
```

Via l'API OAM :

- Interroger le point de terminaison `/api/upf_selection` : `curl -k https://localhost:8443/api/upf_selection`
- Renvoie l'état de santé en temps réel pour tous les UPF dans chaque pool
- Valeurs d'état : Active-UP, Standby-Ready, Active-DOWN, Non associé
- Valeurs de rôle : ACTIVE (poids > 0), STANDBY (poids == 0), DYNAMIC (découvert par DNS, non dans la config)
- Compteur de manques de battement de cœur signalé pour les UPF associés

Meilleures pratiques de surveillance de la santé

1. **Configurer les UPF dans upf_selection** : Tous les UPF dans les règles et les pools de secours sont automatiquement surveillés

```
upf_selection: %{
  rules: [
    %{
      name: "Trafic Internet",
      priority: 10,
      match_field: :apn,
      match_regex: "^internet",
      upf_pool: [
        %{remote_ip_address: "10.100.1.21", remote_port: 8805,
weight: 100}
      ]
    }
  ],
  fallback_pool: [
    %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 100}
  ]
}
# Tous les UPF obtiennent automatiquement :
# - Battements de cœur de 5 secondes
# - Surveillance de la santé dès le démarrage
# - Noms générés automatiquement
```

2. **Utiliser des UPF de secours** : Configurer des secours chauds avec `weight: 0` pour un basculement automatique

```
upf_pool: [
  %{remote_ip_address: "10.1.1.1", remote_port: 8805, weight:
100},  # Actif
  %{remote_ip_address: "10.1.1.2", remote_port: 8805, weight:
0}     # Standby
]
```

3. **Surveiller via l'API OAM** : Interroger régulièrement le point de terminaison `/api/upf_selection` pour vérifier l'état de santé des UPF

4. **Surveillance des battements de cœur** : Le système utilise un seuil fixe de 3 battements de cœur consécutifs manqués pour déterminer la santé du pair.

Enregistrement dynamique UPF

Fonctionnalité : Le PGW-C enregistre automatiquement et surveille les UPF découverts via DNS, même s'ils ne sont pas dans la configuration `upf_selection`.

Comment cela fonctionne

Lorsque n'importe quelle méthode de sélection (règles statiques, pools ou DNS) renvoie un UPF qui n'est pas déjà enregistré, le système automatiquement :

1. **Crée un pair PFCP** : Génère une configuration de pair par défaut pour l'UPF inconnu
2. **Initie l'association PFCP** : Tente d'établir une association PFCP avec l'UPF
3. **Enregistre dans le registre des pairs** : Ajoute l'UPF au système de suivi interne des pairs
4. **Commence la surveillance des battements de cœur** : Commence les échanges de battements de cœur périodiques (intervalles de 10 secondes)
5. **Suit la vivacité** : Surveille l'UPF pour des échecs et des récupérations

Configuration par défaut pour les UPF dynamiques

Lorsqu'un UPF est enregistré dynamiquement, il reçoit la configuration par défaut suivante :

```
%{
  name: "Dynamic-UPF-<IP>",              # par exemple, "Dynamic-
UPF-10-100-1-21"
  remote_ip_address: <discovered_ip>,     # IP du DNS ou de la
sélection
  remote_port: 8805,                      # Port PFCP standard
(remplaçable)
  initiate_pfcp_association_setup: true,  # PGW-C initie
l'association
  heartbeat_period_ms: 10_000             # Intervalle de
battement de cœur de 10 secondes
}
```

> **Remarque** : Les UPF dynamiques sont enregistrés uniquement pour la gestion de l'association. Ils sont utilisés comme cibles dans les règles de `upf_selection`, pas comme sources de logique de sélection.

Exemple : DNS renvoie un UPF inconnu

```
# La requête DNS renvoie : upf-edge-2.example.com -> 10.200.5.99
# Cet UPF n'est PAS dans votre configuration upf_selection

# Flux d'enregistrement dynamique :
# 1. Le système détecte l'UPF inconnu 10.200.5.99
# 2. Journaux : "UPF {10, 200, 5, 99} non préconfiguré, tentative
d'enregistrement dynamique..."
# 3. Envoie une demande de configuration d'association PFCP à
10.200.5.99:8805
# 4. Si l'UPF répond : Association établie, la session continue
normalement
# 5. Si l'UPF ne répond pas : La session échoue proprement avec un
message d'erreur clair
```

Avantages

 Véritable découverte dynamique : La sélection UPF basée sur DNS fonctionne maintenant sans préconfiguration
 Mise à l'échelle automatique : Ajouter des UPF à votre réseau sans redémarrer le PGW-C
 Dégradation en douceur : Si l'association échoue, les sessions échouent

proprement (pas de plantages)
 Compatible avec les versions antérieures : Les UPF préconfigurés continuent de fonctionner exactement comme avant
 Surveillance complète : Les UPF dynamiques bénéficient de la même surveillance des battements de cœur que les pairs statiques

Gestion des échecs

Si un UPF découvert dynamiquement ne répond pas à la configuration d'association PFCP :

```
[error] Échec de la configuration d'association PFCP pour l'UPF
dynamique {10, 200, 5, 99} : :timeout
[error] Échec de l'enregistrement dynamique de l'UPF {10, 200, 5,
99} : :timeout.
        La création de session échouera. Envisagez d'ajouter cet
UPF à la configuration upf_selection.
```

La création de session échouera, mais le PGW-C reste stable et continue de traiter d'autres sessions.

Quand préconfigurer vs enregistrement dynamique

Scénario	Recommandation
UPF de cœur de production	Préconfigurer dans `upf_selection` (configuration explicite, surveillée dès le démarrage)
UPF de bord découvert par DNS	Utiliser l'enregistrement dynamique (mise à l'échelle automatique avec l'infrastructure)
UPF de test/développement	Les deux approches fonctionnent (dynamique est plus pratique)
UPF critiques pour la mission	Préconfigurer dans `upf_selection` (assure la surveillance dès le démarrage)
UPF éphémères/auto-scalés	Utiliser l'enregistrement dynamique (les UPF vont et viennent dynamiquement)

Surveillance des UPF dynamiques

Les UPF dynamiques apparaissent dans les journaux avec leurs noms générés automatiquement :

```
[info] Création de la configuration de pair PFCP dynamique pour
Dynamic-UPF-10-200-5-99 ({10, 200, 5, 99}:8805)
[info] Pair UPF dynamique Dynamic-UPF-10-200-5-99 enregistré avec
succès avec PID #PID<0.1234.0>
```

Vous pouvez interroger le registre des pairs pour voir tous les pairs enregistrés (à la fois statiques et dynamiques) :

```
# Obtenir tous les pairs enregistrés
PGW_C.PFCP_Node.registered_peer_count()

# Vérifier si un UPF spécifique est enregistré
PGW_C.PFCP_Node.get_peer({10, 200, 5, 99})
# Renvoie : {:ok, #PID<0.1234.0>} si enregistré, :error sinon
```

Port personnalisé pour les UPF dynamiques

Si vos UPF utilisent un port PFCP non standard, vous pouvez déclencher manuellement l'enregistrement :

```
# Enregistrer l'UPF à un port personnalisé
PGW_C.PFCP_Node.register_dynamic_peer({10, 200, 5, 99}, 9999)
```

Cependant, la sélection basée sur DNS et l'enregistrement automatique utilisent toujours le port 8805 (port PFCP standard).

Mode Dry-Run de sélection UPF :

Testez et validez votre configuration de sélection UPF sans affecter les sessions réelles :

```
config :pgw_c,
  # Activer le mode dry-run pour les tests (désactivé par défaut)
  upf_selection_dry_run: true
```

Lorsque le mode dry-run est activé :

1. **Aucune affectation réelle** : Les sessions ne sont pas réellement affectées aux UPF
2. **Journalisation détaillée** : Les décisions de sélection sont enregistrées avec tous les détails
3. **Retour d'erreur** : `assign_sxb_peer/1` renvoie `{:error, :dry_run_mode}`
4. **Prévention de session** : Le retour d'une erreur empêche la création de session
5. **Les deux méthodes** : Fonctionne avec les règles statiques et la sélection basée sur DNS

Exemple de sortie de journal :

```
[warning] ⚠  MODE DRY-RUN DE SÉLECTION UPF ACTIVÉ - Aucune
affectation réelle ne se produira

[info]  DRY-RUN : Règle statique correspondante
  Méthode : Règle statique
  Champ de correspondance : :apn
  Expression régulière de correspondance : ~r/^internet\./
  Priorité : 10
  UPF sélectionné : 10.0.1.21:8805

[warning]  DRY-RUN : Affectation UPF prévue mais sautant
l'affectation réelle
  Session IMSI : 001001000000670
  Session APN : internet.apn
  Méthode de sélection : statique
  Lien prévu vers : 10.0.1.21:8805
  ⚠  Retourne une erreur pour empêcher la création de session
```

Tests via l'interface LiveView :

La page LiveView de sélection UPF (`/api/upf_selection`) comprend une interface de test interactive :

1. Naviguez vers la page de sélection UPF dans le panneau web
2. Faites défiler jusqu'à la section "Tester la sélection UPF"
3. Entrez les attributs de session de test :
   - IMSI (par exemple, `001001000000670`)
   - APN (par exemple, `internet.apn`)
   - ID PLMN du réseau de service (par exemple, `001001`)
4. Cliquez sur "Tester la sélection" pour simuler la sélection
5. Affichez les résultats détaillés montrant :
   - Quelle règle a correspondu (ou si DNS serait utilisé)
   - UPF sélectionné et nom du pair
   - Détails du champ de correspondance et du motif
   - Priorité de la règle

L'interface de test LiveView simule la logique de sélection **sans nécessiter que le mode dry-run soit activé globalement**, ce qui la rend sûre à tester

dans des environnements de production sans affecter le trafic réel.

Cas d'utilisation :

- **Tests de configuration** : Valider les règles de routage avant de les déployer en production
- **Dépannage** : Comprendre pourquoi des sessions spécifiques sont routées vers des UPF spécifiques
- **Formation** : Démontrer la logique de sélection UPF aux équipes d'exploitation
- **Développement** : Tester de nouvelles règles de sélection pendant le développement

Champs de correspondance :

- `:imsi` - Identité de l'abonné mobile international
- `:apn` - Nom du point d'accès (APN/DNN)
- `:serving_network_plmn_id` - ID PLMN du réseau de service
- `:sgw_ip_address` - Adresse IP SGW
- `:uli_tai_plmn_id` - ID PLMN de la zone de suivi
- `:uli_ecgi_plmn_id` - ID PLMN de la cellule E-UTRAN

Ajustement du heartbeat :

```
# Aggressif (détecter rapidement les pannes)
heartbeat_period_ms: 2_000  # 2 secondes

# Standard (recommandé)
heartbeat_period_ms: 5_000  # 5 secondes

# Détendu (réseaux à latence élevée)
heartbeat_period_ms: 10_000  # 10 secondes
```

Voir : Documentation de l'interface PFCP

Configuration du pool IP UE

Objectif

Configurer des pools d'adresses IP pour l'allocation UE, organisés par APN.

Bloc de configuration

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # Pools APN "internet"
      "internet" => [
        "100.64.0.0/20"  # 4094 IPs utilisables
      ],

      # Pools APN "ims"
      "ims" => [
        "100.64.16.0/22"  # 1022 IPs utilisables
      ],

      # Pool par défaut pour les APN inconnus
      default: [
        "42.42.42.0/24"  # 254 IPs utilisables
      ]
    }
  }
```

Paramètres

Paramètre	Type	Requis	Description
`subnet_map`	Map	Oui	Mappe les noms d'APN aux listes de sous-réseaux
`default`	Liste (Sous-réseaux)	Oui	Pool de secours pour les APN inconnus

Format de sous-réseau

Notation CIDR : `<network_address>/<prefix_length>`

Calcul des IP utilisables :

CIDR	Total IPs	IPs utilisables	Plage d'exemple
/24	256	254	100.64.1.1 - 100.64.1.254
/23	512	510	100.64.0.1 - 100.64.1.254
/22	1024	1022	100.64.0.1 - 100.64.3.254
/21	2048	2046	100.64.0.1 - 100.64.7.254
/20	4096	4094	100.64.0.1 - 100.64.15.254
/16	65536	65534	100.64.0.1 - 100.64.255.254

Exemples

Configuration simple :

```
ue: %{
  subnet_map: %{
    "internet" => ["100.64.1.0/24"],
    default: ["42.42.42.0/24"]
  }
}
```

Configuration de production :

```
ue: %{
  subnet_map: %{
    # Internet général - grand pool
    "internet" => [
      "100.64.0.0/18"  # 16,382 IPs
    ],

    # IMS (VoLTE)
    "ims" => [
      "100.64.64.0/22"  # 1,022 IPs
    ],

    # APN d'entreprise
    "enterprise.corp" => [
      "10.100.0.0/16"  # 65,534 IPs
    ],

    # Dispositifs IoT
    "iot.m2m" => [
      "100.64.72.0/20"  # 4,094 IPs
    ],

    # Secours par défaut
    default: [
      "42.42.42.0/24"  # 254 IPs
    ]
  }
}
```

Équilibrage de charge (plusieurs sous-réseaux par APN) :

```
ue: %{
  subnet_map: %{
    "internet" => [
      "100.64.0.0/22",   # 1022 IPs
      "100.64.4.0/22",   # 1022 IPs
      "100.64.8.0/22",   # 1022 IPs
      "100.64.12.0/22"   # 1022 IPs
    ],
    # Total : 4088 IPs, distribuées aléatoirement
    default: ["42.42.42.0/24"]
  }
}
```

Correspondance de motifs Regex :

Pour la correspondance d'APN avec des jokers, utilisez des clés commençant par `^` pour indiquer des motifs regex :

```
ue: %{
  subnet_map: %{
    # Regex : APNs commençant par "ims" (correspond à "ims",
"ims.apn", "ims.something")
    "^ims" => [
      "100.64.10.0/24"
    ],

    # Regex : APNs commençant par "m2m." (correspond à "m2m.test",
"m2m.prod")
    "^m2m\." => [
      "100.64.20.0/24"
    ],

    # Correspondance exacte uniquement
    "enterprise.corp" => [
      "10.100.0.0/16"
    ],

    default: ["42.42.42.0/24"]
  }
}
```

Règles de motifs :

- Les clés commençant par `^` sont des motifs regex
- Les clés sans `^` sont des correspondances exactes (compatible avec les versions antérieures)
- Les motifs regex sont compilés au moment de l'analyse de la configuration
- Le premier motif correspondant l'emporte
- Les barres obliques doivent être échappées (`\` pour `\`)

Motifs courants :

- **Commence par :** `"^ims"` correspond à `ims`, `ims.apn`, `ims.foo.bar`
- **Se termine par :** `"^.*\.corp$"` correspond à `foo.corp`, `bar.corp`
- **Contient :** `"^.*test.*"` correspond à `test`, `foo.test.bar`

Support IPv6 :

```
ue: %{
  subnet_map: %{
    # Pools IPv4
    "internet" => [
      "100.64.0.0/20"
    ],

    # Pools IPv6 (délégation de préfixe)
    "internet.ipv6" => [
      "2001:db8:1::/48"
    ],

    default: [
      "42.42.42.0/24"
    ]
  }
}
```

Plages IP recommandées

RFC 6598 (NAT de niveau opérateur) :

- **Plage :** `100.64.0.0/10`
- **Taille :** ~4 millions d'IPs
- **Objectif :** Conçu pour le NAT des fournisseurs de services

Plages IP privées (RFC 1918) :

- `10.0.0.0/8` - 16 millions d'IPs
- `172.16.0.0/12` - 1 million d'IPs
- `192.168.0.0/16` - 65,534 IPs

Voir : Documentation sur l'allocation de pools IP UE

Configuration PCO

Objectif

Configurer les options de configuration de protocole (PCO) envoyées à l'UE.

Bloc de configuration

```
config :pgw_c,
  pco: %{
    # Serveurs DNS
    primary_dns_server_address: "8.8.8.8",
    secondary_dns_server_address: "8.8.4.4",

    # Serveurs DNS IPv6 (optionnel, conteneur 0x0003)
    primary_dns_server_ipv6_address: nil,
    secondary_dns_server_ipv6_address: nil,

    # Serveurs NBNS (optionnel, pour les dispositifs Windows)
    primary_nbns_server_address: nil,
    secondary_nbns_server_address: nil,

    # Adresses P-CSCF pour IMS
    p_cscf_ipv4_address_list: ["10.0.0.50", "10.0.0.51"],   #
conteneur 0x000C
    p_cscf_ipv6_address_list: [],                           #
conteneur 0x0001

    # Taille MTU IPv4
    ipv4_link_mtu_size: 1400
  }
```

Paramètres

Paramètre	Type	Par défaut	Description
`primary_dns_server_address`	String (IPv4)	Requis	Serveur DNS principal
`secondary_dns_server_address`	String (IPv4)	Optionnel	Serveur DNS secondaire
`primary_dns_server_ipv6_address`	String (IPv6)	`nil`	Serveur DNS IPv6 principal (conteneur 0x0003)
`secondary_dns_server_ipv6_address`	String (IPv6)	`nil`	Serveur DNS IPv6 secondaire (conteneur 0x0003)
`primary_nbns_server_address`	String (IPv4)	`nil`	Serveur NBNS (NetBIOS) principal
`secondary_nbns_server_address`	String (IPv4)	`nil`	Serveur NBNS secondaire
`p_cscf_ipv4_address_list`	Liste (IPv4)	`[]`	Adresses P-CSCF IPv4 pour IMS (conteneur 0x000C)

Paramètre	Type	Par défaut	Description
`p_cscf_ipv6_address_list`	Liste (IPv6)	`[]`	Adresses P-CSCF IPv6 pour IMS (conteneur 0x0001)
`ipv4_link_mtu_size`	Entier	`1400`	Taille de l'unité de transmission maximale

Exemples

DNS public (Google) :

```
pco: %{
  primary_dns_server_address: "8.8.8.8",
  secondary_dns_server_address: "8.8.4.4",
  ipv4_link_mtu_size: 1400
}
```

DNS privé :

```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11",
  ipv4_link_mtu_size: 1400
}
```

Configuration IMS :

```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11",

  # P-CSCF pour IMS/VoLTE
  p_cscf_ipv4_address_list: [
    "10.0.0.50",  # P-CSCF principal
    "10.0.0.51"   # P-CSCF secondaire
  ],

  ipv4_link_mtu_size: 1400
}
```

NBNS (Compatibilité Windows) :

```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11",
  primary_nbns_server_address: "10.0.0.20",
  secondary_nbns_server_address: "10.0.0.21",
  ipv4_link_mtu_size: 1400
}
```

Ajustement MTU :

```
# Ethernet standard
ipv4_link_mtu_size: 1500

# Réduit pour la surcharge de tunneling
ipv4_link_mtu_size: 1400

# Jumbo frames (si supporté)
ipv4_link_mtu_size: 9000
```

Voir : Documentation sur la configuration PCO

Configuration de l'API OAM

Objectif

Configurer les points de terminaison de l'API REST OAM pour gérer et surveiller OmniPGW. Toute la surveillance et la gestion sont effectuées via cette API.

> **Remarque :** La configuration de l'API n'est active que dans les environnements non-test. La configuration est automatiquement ignorée lorsque `config_env()` est `:test`, `:test_mock`, ou `:test_impl`.

Configuration de l'API REST

```
config :api_ex,
  api: %{
    # Paramètres réseau
    port: 8443,
    listen_ip: "0.0.0.0",

    # Métadonnées de l'API
    product_name: "PGW-C",
    title: "API - PGW-C",
    hostname: "localhost",

    # Paramètres TLS
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem",

    # Définitions de routes
    routes: [
      %{
        path: "/status",
        module: ApiEx.Api.StatusController,
        actions: [:index]
      }
    ]
  }
```

Paramètres de l'API

Paramètre	Type	Par défaut	Description
`port`	Entier	`8443`	Port HTTPS pour l'API REST
`listen_ip`	String (IP)	`"0.0.0.0"`	Adresse d'écoute de l'API (0.0.0.0 = toutes les interfaces)
`product_name`	String	`"PGW-C"`	Nom du produit (pour les métadonnées de l'API)
`title`	String	`"API - PGW-C"`	Titre de la documentation de l'API
`hostname`	String	`"localhost"`	Nom d'hôte du serveur API
`enable_tls`	Boolean	`true`	Activer HTTPS pour l'API
`tls_cert_path`	String (Path)	Requis	Chemin vers le fichier de certificat SSL/TLS
`tls_key_path`	String (Path)	Requis	Chemin vers le fichier de clé privée SSL/TLS
`routes`	Liste (Maps)	`[]`	Définitions de routes de l'API

Format de définition de route

Chaque route dans la liste `routes` est une carte avec :

Champ	Type	Description	Exemple
`path`	String	Chemin URL pour la route	`"/status"`
`module`	Module	Module de contrôleur gérant les requêtes	`ApiEx.Api.StatusController`
`actions`	Liste (Atoms)	Actions de contrôleur autorisées	`[:index, :show]`

Exemples

Configuration par défaut (Développement) :

```
# API sur localhost:8443
config :api_ex,
  api: %{
    port: 8443,
    listen_ip: "127.0.0.1",  # localhost uniquement
    hostname: "localhost",
    enable_tls: true,
    tls_cert_path: "priv/cert/dev-cert.crt",
    tls_key_path: "priv/cert/dev-key.pem",
    routes: [
      %{path: "/status", module: ApiEx.Api.StatusController,
actions: [:index]}
    ]
  }
```

Configuration de production :

```
# API sur l'interface de gestion
config :api_ex,
  api: %{
    port: 8443,
    listen_ip: "10.0.0.20",  # Réseau de gestion
    product_name: "OmniPGW-C",
    hostname: "pgw-c-api.example.com",
    enable_tls: true,
    tls_cert_path: "/etc/ssl/certs/pgw-c-api.crt",
    tls_key_path: "/etc/ssl/private/pgw-c-api.key",
    routes: [
      %{path: "/status", module: ApiEx.Api.StatusController,
actions: [:index]},
      %{path: "/sessions", module: ApiEx.Api.SessionController,
actions: [:index, :show]}
    ]
  }
```

Accéder à l'API

Interroger un point de terminaison :

```
# Point de terminaison de statut (certificat auto-signé,
développement)
curl -k https://localhost:8443/api/status

# Avec un certificat de confiance (production)
curl https://pgw-c-api.example.com:8443/api/status
```

Toutes les routes sont servies sous le préfixe `/api`. Les points de terminaison disponibles incluent `/api/status`, `/api/sessions`, `/api/upf`, `/api/diameter`, `/api/charging`, `/api/ip_pools`, `/api/session_history`, `/api/topology`, `/api/ue_search` (POST), `/api/pcscf_monitor`, et `/api/upf_selection`.

Documentation API interactive (Swagger UI) :

```
https://localhost:8443/api/docs
```

Configuration du certificat TLS

Générer un certificat auto-signé (Développement) :

```
# Générer la clé privée et le certificat
openssl req -x509 -newkey rsa:4096 -keyout priv/cert/omnitouch.pem
\
  -out priv/cert/omnitouch.crt -days 365 -nodes \
  -subj "/CN=localhost"
```

Certificat de production :

Pour la production, utilisez des certificats d'une autorité de certification (CA) de confiance :

- Let's Encrypt (gratuit, automatisé)
- CA commerciale (DigiCert, GlobalSign, etc.)
- CA interne pour les déploiements d'entreprise

Considérations de sécurité

1. **Utilisez toujours TLS en production** - Définissez `enable_tls: true`
2. **Restreindre listen_ip** - Utilisez des adresses IP spécifiques en production (pas 0.0.0.0)
3. **Utilisez des certificats valides** - Évitez les certificats auto-signés en production
4. **Protection par pare-feu** - Restreindre l'accès au port de gestion (8443)
5. **Fichiers de clé forts** - Utilisez RSA 4096 bits ou équivalent
6. **Mises à jour régulières** - Faites tourner les certificats avant leur expiration

Dépannage

Problème : Impossible d'accéder à l'API

```
# Vérifiez si le port est à l'écoute
netstat -tulpn | grep 8443

# Vérifiez que les fichiers de certificat existent
ls -la priv/cert/

# Vérifiez les journaux pour les erreurs de démarrage
tail -f /var/log/pgw_c/application.log
```

Problème : Erreurs SSL/TLS

- Vérifiez que les chemins des certificats et des clés sont corrects
- Assurez-vous que le certificat correspond au nom d'hôte
- Vérifiez l'expiration du certificat : `openssl x509 -in cert.crt -text -noout`
- Vérifiez les permissions du fichier de clé (doit être lisible par le processus PGW-C)

Voir : Guide de surveillance pour les détails d'utilisation de l'API OAM

Exemple complet

Configuration prête pour la production

```
# config/runtime.exs
import Config

# Configuration du journal
config :logger, level: :info

config :pgw_c,
  # Métriques (Prometheus)
  metrics: %{
    enabled: true,
    ip_address: "10.0.0.20",  # Réseau de gestion
    port: 9090,
    registry_poll_period_ms: 5_000
  },

  # Diameter/Gx (interface PCRF)
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omnipgw.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf-primary.epc.mnc001.mcc001.3gppnetwork.org",
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        ip: "10.0.1.30",
        initiate_connection: true
      },
      %{
        host: "pcrf-backup.epc.mnc001.mcc001.3gppnetwork.org",
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        ip: "10.0.2.30",
        initiate_connection: true
      }
    ]
  },

  # S5/S8 (interface SGW-C)
  s5s8: %{
```

```
    local_ipv4_address: "10.0.0.20"
  },

  # Sxb/PFCP (interface PGW-U)
  sxb: %{
    local_ip_address: "10.0.0.20"
  },

  # Sélection UPF
  upf_selection: %{
    rules: [
      %{
        name: "Internet par défaut",
        priority: 1,
        match_field: :apn,
        match_regex: ~r/^internet/,
        upf_pool: [
          %{remote_ip_address: "10.0.0.21", remote_port: 8805,
weight: 100},
          %{remote_ip_address: "10.0.0.22", remote_port: 8805,
weight: 0}  # En attente
        ]
      }
    ],
    fallback_pool: [
      %{remote_ip_address: "10.0.0.21", remote_port: 8805, weight:
100}
    ]
  },

  # Pools IP UE
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.0.0/18"  # 16,382 IPs
      ],
      "ims" => [
        "100.64.64.0/22"  # 1,022 IPs
      ],
      "enterprise.corp" => [
        "10.100.0.0/16"  # 65,534 IPs
      ],
      default: [
        "100.64.127.0/24"  # 254 IPs
```

```
      ]
    }
  },

  # Options de configuration de protocole
  pco: %{
    primary_dns_server_address: "8.8.8.8",
    secondary_dns_server_address: "8.8.4.4",
    p_cscf_ipv4_address_list: ["10.0.0.50", "10.0.0.51"],
    ipv4_link_mtu_size: 1400
  }
```

Validation de la configuration

Validation au démarrage

OmniPGW valide la configuration au démarrage. Vérifiez les journaux :

```
[info] Chargement de la configuration depuis runtime.exs
[info] Validation de la configuration...
[info] Configuration valide
[info] Démarrage d'OmniPGW...
```

Erreurs de validation courantes

Adresse IP invalide :

```
[error] Adresse IP invalide dans s5s8.local_ipv4_address: "10.0.0"
```

Champ requis manquant :

```
[error] Configuration requise manquante : sxb.local_ip_address
```

CIDR invalide :

```
[error] Sous-réseau invalide dans ue.subnet_map: "100.64.1.0/33"
```

Identité Diameter invalide :

```
[error] L'hôte Diameter doit être un FQDN, pas une IP :
"10.0.0.20"
```

Test de la configuration

Tester la configuration sans démarrer :

```
# Valider la syntaxe
mix compile

# Vérifier le chargement de la configuration
iex -S mix
iex> Application.get_env(:pgw_c, :metrics)
```

Configuration spécifique à

l'environnement

Développement

```
# config/dev.exs
import Config

config :logger, level: :debug

config :pgw_c,
  metrics: %{
    enabled: true,
    ip_address: "127.0.0.1",
    port: 42069,
    registry_poll_period_ms: 10_000
  },
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omnipgw-dev.local",
    realm: "local",
    peer_list: [
      %{
        host: "pcrf-dev.local",
        realm: "local",
        ip: "127.0.0.1",
        initiate_connection: true
      }
    ]
  },
  s5s8: %{
    local_ipv4_address: "127.0.0.10"
  },
  sxb: %{
    local_ip_address: "127.0.0.20"
  },
  upf_selection: %{
    fallback_pool: [
      %{remote_ip_address: "127.0.0.21", remote_port: 8805,
weight: 100}
    ]
  },
  ue: %{
```

```
    subnet_map: %{
      "internet" => ["100.64.1.0/24"],
      default: ["42.42.42.0/24"]
    }
  },
  pco: %{
    primary_dns_server_address: "8.8.8.8",
    secondary_dns_server_address: "8.8.4.4",
    ipv4_link_mtu_size: 1400
  }
```

Utilisation des variables d'environnement

```
# config/runtime.exs
import Config

config :pgw_c,
  metrics: %{
    enabled: System.get_env("METRICS_ENABLED", "true") == "true",
    ip_address: System.get_env("METRICS_IP", "0.0.0.0"),
    port: String.to_integer(System.get_env("METRICS_PORT",
"9090")),
    registry_poll_period_ms: 10_000
  },
  diameter: %{
    listen_ip: System.get_env("DIAMETER_LISTEN_IP", "0.0.0.0"),
    host: System.get_env("DIAMETER_HOST") || raise("DIAMETER_HOST
requis"),
    realm: System.get_env("DIAMETER_REALM") ||
raise("DIAMETER_REALM requis"),
    peer_list: [
      %{
        host: System.get_env("PCRF_HOST") || raise("PCRF_HOST
requis"),
        realm: System.get_env("PCRF_REALM") ||
System.get_env("DIAMETER_REALM"),
        ip: System.get_env("PCRF_IP") || raise("PCRF_IP requis"),
        initiate_connection: true
      }
    ]
  }
  # ... reste de la config
```

Utilisation :

```
export DIAMETER_HOST="omnipgw.epc.mnc001.mcc001.3gppnetwork.org"
export DIAMETER_REALM="epc.mnc001.mcc001.3gppnetwork.org"
export PCRF_HOST="pcrf.epc.mnc001.mcc001.3gppnetwork.org"
export PCRF_IP="10.0.0.30"

mix run --no-halt
```

Documentation connexe

Configuration de l'interface

- **Interface PFCP** - Configuration Sxb/PFCP, communication UPF, établissement de session
- **Interface Diameter Gx** - Contrôle de politique PCRF, règles PCC, gestion de la QoS
- **Interface Diameter Gy** - Facturation en ligne OCS, gestion de quota, contrôle de crédit
- **Interface S5/S8** - Configuration GTP-C v2, communication SGW-C (4G/LTE)
- **Interface Gn/Gp** - Configuration GTP-C v1, communication SGSN (2G/3G GGSN)

Configuration réseau

- **Allocation IP UE** - Gestion des pools IP, allocation basée sur l'APN, DHCP
- **Configuration PCO** - Options de configuration de protocole, DNS, P-CSCF, MTU
- **Surveillance P-CSCF** - Surveillance de la découverte P-CSCF, suivi de la santé IMS

Guides opérationnels

- **Gestion des sessions** - Cycle de vie de la session PDN, gestion des porteurs
- **Guide de surveillance** - Métriques Prometheus, alertes, tableaux de bord
- **Format CDR de données** - Enregistrements de facturation hors ligne, intégration de facturation

---

Retour au guide des opérations

Guide de configuration OmniPGW - *par Omnitouch Network Services*

Format du Dossier de Données de Facturation (CDR)

Facturation Hors Ligne pour PGW-C

OmniPGW par Omnitouch Network Services

Table des Matières

1. Aperçu
2. Format de Fichier CDR
3. Champs CDR
4. Événements CDR
5. Structure de Fichier
6. Configuration
7. Flux de Génération CDR
8. Détails des Champs
9. Exemples
10. Intégration

Aperçu

Le **format CDR (Dossier de Données de Facturation)** fournit des capacités de facturation hors ligne pour le Plan de Contrôle de la Passerelle de Paquet (PGW-C). Les CDR sont générés pour enregistrer les événements de session de porteur, l'utilisation des données et les informations sur les abonnés à des fins de facturation et d'analytique.

Ce format commun est compatible avec les CDR SGW-C, garantissant la cohérence des enregistrements de facturation à travers l'infrastructure EPC.

Caractéristiques Clés

- **Format basé sur CSV** - Valeurs séparées par des virgules, simples et lisibles par l'homme
- **Enregistrement basé sur des événements** - Capture les événements de début, de mise à jour et de fin de porteur
- **Mesure de volume** - Enregistre l'utilisation des données en amont et en aval
- **Rotation automatique** - Rotation de fichier configurable basée sur des intervalles de temps
- **Conforme à 3GPP** - Suit 3GPP TS 32.251 (facturation du domaine PS) et TS 32.298 (encodage CDR)

Cas d'Utilisation

Cas d'Utilisation	Description
Facturation Hors Ligne	Générer des CDR pour la facturation postpayée
Analytique	Analyser les modèles d'utilisation des abonnés
Piste d'Audit	Suivre tous les événements de session de porteur
Planification de Capacité	Surveiller l'utilisation des ressources réseau
Dépannage	Déboguer les problèmes de session et de porteur

Format de Fichier CDR

Convention de Nommage de Fichier

```
<epoch_timestamp>
```

Exemple :

```
1726598022
```

Le nom de fichier est le timestamp Unix epoch (en secondes) du moment où le fichier a été créé.

Emplacement du Fichier

Répertoire par défaut :

- PGW-C : `/var/log/pgw_c/cdrs/`

Configuré via le paramètre `cdr_directory` dans config/runtime.exs.

En-tête de Fichier

Chaque fichier CDR commence par un en-tête multi-lignes contenant des métadonnées :

```
# Fichier CDR de Données :
# Heure de Début du Fichier : HH:MM:SS (unix_timestamp)
# Heure de Fin du Fichier : HH:MM:SS (unix_timestamp)
# Nom de la Passerelle : <gateway_name>
#
epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,eci
```

Champs d'En-tête :

- **Heure de Début du Fichier** - Quand le fichier CDR a été créé (lisible par l'homme et timestamp Unix)
- **Heure de Fin du Fichier** - Quand la rotation de fichier aura lieu (lisible par l'homme et timestamp Unix)
- **Nom de la Passerelle** - Identifiant pour l'instance PGW-C (configuré via le paramètre `pgw_name`)
- **En-têtes de Colonne** - Noms de champs CSV pour les enregistrements de données

---

Champs CDR

Résumé des Champs

Position	Nom du Champ	Type	Description
0	epoch	entier	Timestamp de l'événement (secondes epoch Unix)
1	imsi	chaîne	Identité Internationale de l'Abonné Mobile
2	event	chaîne	Type d'événement CDR (par exemple, "default_bearer_start")
3	charging_id	entier	Identifiant de facturation unique pour le porteur
4	msisdn	chaîne	Numéro ISDN de la Station Mobile (numéro de téléphone)
5	ue_imei	chaîne	Identité Internationale de l'Équipement Mobile
6	timezone_raw	chaîne	Fuseau horaire UE (réservé, actuellement vide)
7	plmn	entier	Identifiant du Réseau Mobile Terrestre Public
8	tac	entier	Code de Zone de Suivi
9	eci	entier	Identifiant de Cellule E-UTRAN

Position	Nom du Champ	Type	Description
10	sgw_ip	chaîne	Adresse IP du Plan de Contrôle SGW-C S5/S8
11	ue_ip	chaîne	Adresse IP de l'UE (format IPv4|IPv6)
12	pgw_ip	chaîne	Adresse IP du Plan de Contrôle PGW-C S5/S8
13	apn	chaîne	Nom du Point d'Accès
14	qci	entier	Identifiant de Classe QoS
15	octets_in	entier	Volume de données en aval (octets)
16	octets_out	entier	Volume de données en amont (octets)

Événements CDR

Types d'Événements

Les CDR sont générés pour trois types d'événements :

Type d'Événement	Format	Description	Quand Généré
Début de Porteur	`<type>_bearer_start`	Établissement du porteur	Réponse de Création de Session envoyée
Mise à Jour de Porteur	`<type>_bearer_update`	Rapport d'utilisation pendant la session	Rapports d'utilisation périodiques du plan utilisateur
Fin de Porteur	`<type>_bearer_end`	Terminaison du porteur	Demande/Répons de Suppression d Session

Types de Porteurs :

- `default` - Porteur par défaut (un par connexion PDN)
- `dedicated` - Porteur dédié (zéro ou plusieurs par connexion PDN)

Exemples d'Événements

```
default_bearer_start      - Porteur par défaut établi
default_bearer_update     - Mise à jour de l'utilisation du
porteur par défaut
default_bearer_end        - Porteur par défaut terminé
dedicated_bearer_start    - Porteur dédié établi
dedicated_bearer_update   - Mise à jour de l'utilisation du
porteur dédié
dedicated_bearer_end      - Porteur dédié terminé
```

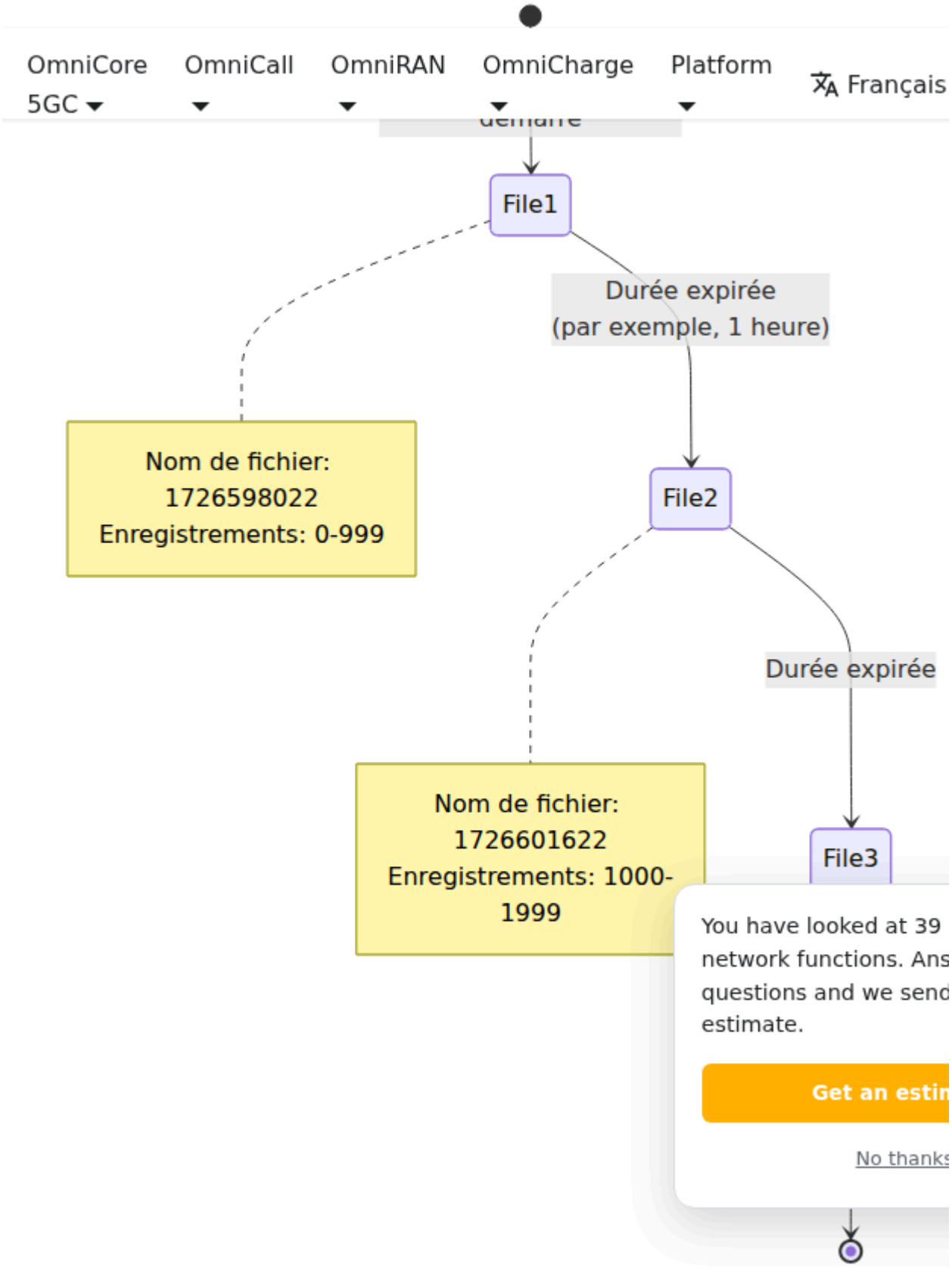
Structure de Fichier

Exemple de Fichier CDR

```
# Fichier CDR de Données :
# Heure de Début du Fichier : 18:53:42 (1726598022)
# Heure de Fin du Fichier : 19:53:42 (1726601622)
# Nom de la Passerelle : sgw-c-prod-01
# epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,e
1726598022,310260123456789,default_bearer_start,12345,15551234567,123
1726598322,310260123456789,default_bearer_update,12345,15551234567,12
1726598622,310260123456789,default_bearer_update,12345,15551234567,12
1726598922,310260123456789,default_bearer_end,12345,15551234567,12345
```

Rotation de Fichier

Les fichiers CDR sont automatiquement tournés en fonction de la durée configurée :



Processus de Rotation :

1. Fermer le fichier CDR actuel
2. Créer un nouveau fichier avec le timestamp actuel

3. Écrire l'en-tête dans le nouveau fichier
4. Continuer à enregistrer les CDR dans le nouveau fichier

Configuration

Paramètres de Configuration

La génération de CDR PGW-C est configurée dans `config/runtime.exs` :

Paramètre	Type	Description	Par Défaut	Recc
`pgw_name`	chaîne	Identifiant de l'instance PGW (apparaît dans les en-têtes CDR)	"omni-pgw01"	Utiliser l ou l'ID d'
`cdr_file_duration`	entier	Intervalle de rotation de fichier (ms)	3600000	3600000
`cdr_directory`	chaîne	Chemin du répertoire de sortie CDR	"/tmp/pgw_c"	`/var/lo`
`usage_report_interval`	entier	Intervalle de rapport URR (ms) - fréquence d'envoi des rapports d'utilisation par PGW-U	60000	60000 (1

Exemples de Configuration

Configuration Minimale (config/runtime.exs) :

```
config :pgw_c,
  # Configuration du fichier CDR
  pgw_name: "omni-pgw01",
  cdr_file_duration: 3_600_000,           # 1 heure
  cdr_directory: "/var/log/pgw_c/cdrs",

  # Configuration URR (déclenche les rapports d'utilisation de
PGW-U)
  usage_report_interval: 60_000           # 60 secondes
```

Production :

```
config :pgw_c,
  pgw_name: "pgw-c-prod-01",
  cdr_file_duration: 3_600_000,           # rotation de 1 heure
  cdr_directory: "/var/log/pgw_c/cdrs",
  usage_report_interval: 60_000           # mises à jour de 1
minute
```

Développement :

```
config :pgw_c,
  pgw_name: "pgw-c-dev",
  cdr_file_duration: 300_000,             # rotation de 5 minutes
pour les tests
  cdr_directory: "/tmp/pgw_c_cdrs",
  usage_report_interval: 30_000           # mises à jour de 30
secondes pour des tests plus rapides
```

Haut Volume :

```
config :pgw_c,
  pgw_name: "pgw-c-prod-heavy",
  cdr_file_duration: 1_800_000,           # rotation de 30 minutes
  cdr_directory: "/mnt/fast-storage/cdrs",
  usage_report_interval: 300_000          # mises à jour de 5
minutes (réduire la surcharge)
```

URR (Règles de Rapport d'Utilisation)

PGW-C utilise **PFCP URRs (Règles de Rapport d'Utilisation)** pour déclencher des rapports d'utilisation de PGW-U. Lorsqu'un seuil URR est atteint ou que le temps expire, PGW-U envoie une Demande de Rapport de Session contenant des données d'utilisation, ce qui déclenche la génération de CDR.

Comment Fonctionne la Configuration URR :

1. `usage_report_interval` (en ms) est converti en secondes pour le seuil de temps PFCP
2. PGW-C crée URR avec seuil de temps lors de l'établissement de session
3. PGW-U envoie des rapports d'utilisation périodiques à l'intervalle configuré
4. Chaque rapport d'utilisation déclenche un événement CDR `bearer_update`
5. Le dernier rapport d'utilisation (lors de la suppression de session) déclenche un événement CDR `bearer_end`

Exemple : `usage_report_interval: 60_000` signifie :

- PGW-U rapporte l'utilisation toutes les 60 secondes
- Événements de mise à jour CDR générés toutes les 60 secondes
- Suivi granulaire de l'utilisation pour la facturation

Définition du Type URR :

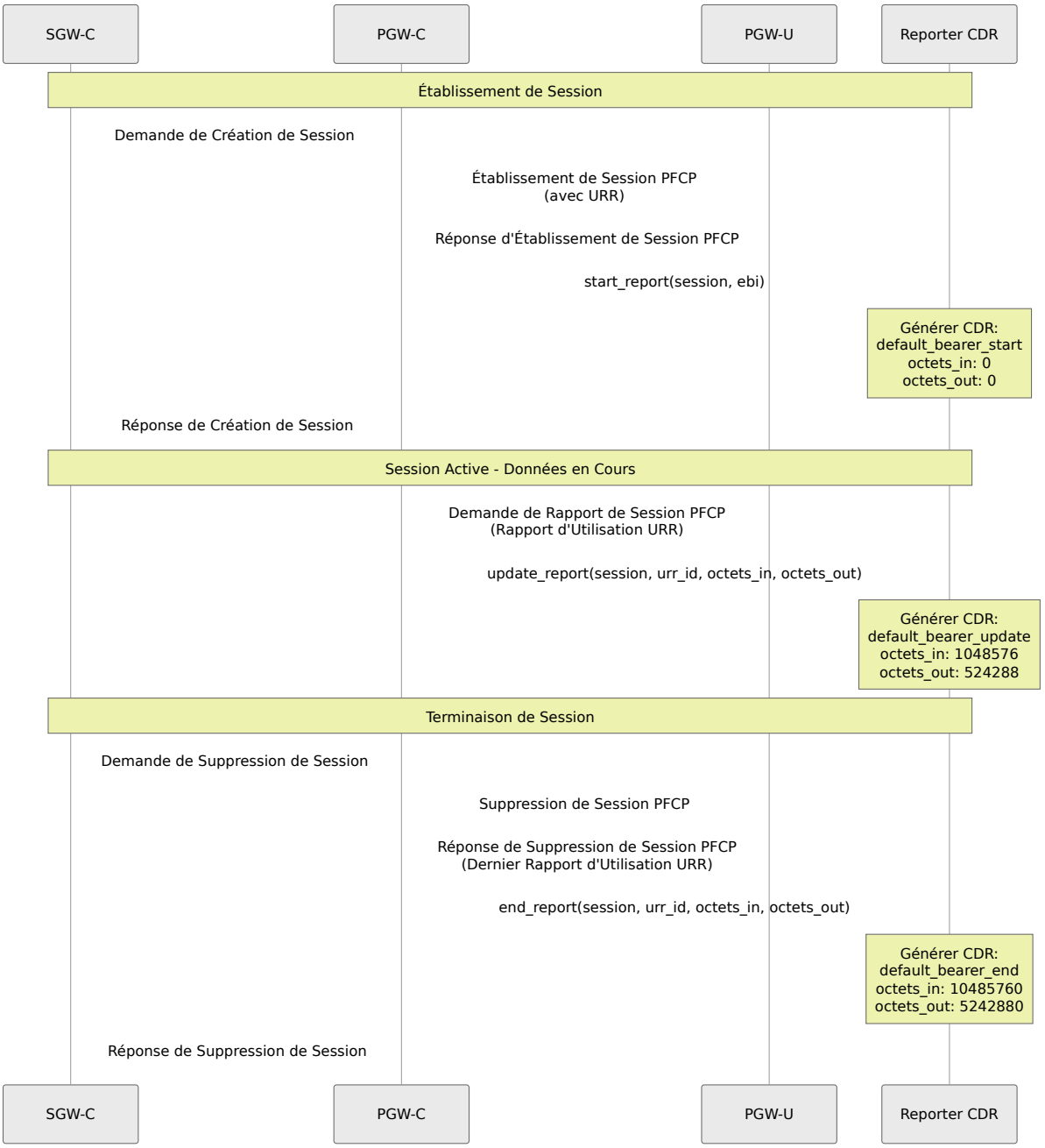
```
defmodule PGW_C.Session.Types.URR do
  typedstruct do
    field :urr_id, non_neg_integer()
    field :measurement_method, :duration | nil
    field :reporting_triggers, :time_threshold | nil
    field :time_threshold, non_neg_integer() | nil  # secondes
  end
end
```

Voir Documentation de l'Interface PFCP pour les détails PFCP URR, y compris comment les URRs sont créés lors de l'établissement de session.

Flux de Génération CDR

Événements CDR du Cycle de Vie du Porteur

Génération CDR PGW-C :



Événements de Génération CDR

1. Début de Porteur :

- **Quand :** La Réponse de Création de Session est envoyée
- **But :** Enregistre l'établissement du porteur avec une utilisation nulle
- **octets_in :** 0
- **octets_out :** 0

2. Mise à Jour de Porteur :

- **Quand :** Demande de Rapport de Session PFCP reçue de PGW-U (rapport d'utilisation URR)
- **But :** Enregistre l'utilisation des données incrémentales
- **octets_in :** Octets en aval cumulés depuis le début du porteur
- **octets_out :** Octets en amont cumulés depuis le début du porteur
- **Déclencheur :** Le seuil de temps URR expire (configuré via `usage_report_interval`)

3. Fin de Porteur :

- **Quand :** Réponse de Suppression de Session PFCP reçue de PGW-U (avec le dernier rapport d'utilisation)
- **But :** Enregistre l'utilisation finale des données avant la terminaison de session
- **octets_in :** Total final des octets en aval
- **octets_out :** Total final des octets en amont

Détails des Champs

1. epoch (Timestamp)

Type : Timestamp Unix epoch (secondes)

Description : Le moment où l'événement CDR s'est produit

Exemple :

```
1726598022  → 2025-09-17 18:53:42 UTC
```

---

2. imsi (Identité de l'Abonné)

Type : Chaîne (jusqu'à 15 chiffres)

Format : MCCMNC + MSIN

Description : Identité Internationale de l'Abonné Mobile identifiant de manière unique l'abonné

Exemple :

```
310260123456789
  └──┘ └──┘ └─────────┘
   MCC  MNC     MSIN
   (310)(260) (123456789)
```

Source : Contexte UE, reçu dans la Demande de Création de Session

---

3. event (Type d'Événement CDR)

Type : Chaîne

Format : `<bearer_type>_bearer_<event>`

Valeurs :

- `default_bearer_start`
- `default_bearer_update`
- `default_bearer_end`
- `dedicated_bearer_start`
- `dedicated_bearer_update`
- `dedicated_bearer_end`

Détermination :

- Si EBI (ID de Porteur EPS) est égal à LBI (ID de Porteur Lié) : `default`
- Si EBI n'est pas égal à LBI : `dedicated`

Source : Contexte du porteur (comparaison EBI vs LBI)

---

4. charging_id (Identifiant de Facturation)

Type : Entier non signé 32 bits

Description : Identifiant unique pour la corrélation de facturation à travers les éléments du réseau

Exemple :

```
12345
```

Source : Assigné par PGW-C, reçu dans la Réponse de Création de Session

Utilisation :

- Corrèle les événements de facturation à travers SGW et PGW
- Utilisé dans les interfaces de facturation Diameter Gy/Gz
- Unique par porteur

---

5. msisdn (Numéro de Téléphone)

Type : Chaîne (format E.164)

Description : Numéro ISDN de la Station Mobile (numéro de téléphone de l'abonné)

Format : Code pays + numéro national

Exemple :

```
15551234567
 └┘└──────┘
  CC  National
  (1) (5551234567)
```

Source : Contexte UE, typiquement du HSS via MME

---

6. ue_imei (Identité de l'Équipement)

Type : Chaîne (15 chiffres)

Format : TAC (8) + SNR (6) + Réserve (1)

Description : Identité Internationale de l'Équipement Mobile (identifiant de l'appareil)

Exemple :

```
123456789012345
└──────┘└────┘└
   TAC   SNR S
```

Source : Contexte UE, reçu de MME

---

7. timezone_raw (Fuseau Horaire UE)

Type : Chaîne (actuellement réservé/vide)

Description : Champ réservé pour les informations de fuseau horaire de l'UE

Statut Actuel : Non peuplé (champ vide dans le CSV)

Utilisation Future : Peut inclure le décalage horaire et le drapeau d'heure d'été

Exemple :

```
,  (champ vide)
```

---

8. plmn (Identifiant du Réseau)

Type : Entier (format hérité)

Description : Identifiant du Réseau Mobile Terrestre Public encodé en hexadécimal little-endian

Processus d'Encodage :

```
MCC: 505, MNC: 57
  ↓
"50557"
  ↓
Échanger les paires : "055570"
  ↓
Hex à décimal : 0x055570 = 349552
```

Exemple :

```
349552  → MCC: 505, MNC: 57
```

Source : Informations de localisation de l'UE provenant de MME

Remarque : Il s'agit d'un format d'encodage hérité pour la compatibilité ascendante

---

9. tac (Code de Zone de Suivi)

Type : Entier non signé 16 bits

Description : Le Code de Zone de Suivi identifie la zone de suivi où se trouve l'UE

Plage : 0 - 65535

Exemple :

```
1234
```

Source : Informations de localisation de l'UE, reçues de MME dans la Demande de Création de Session

Utilisation :

- Identifie la zone de gestion de mobilité
- Utilisé pour la recherche et les mises à jour de localisation
- Partie de l'IAZ (Identité de Zone de Suivi)

---

10. eci (Identifiant de Cellule E-UTRAN)

Type : Entier non signé 28 bits

Description : L'Identifiant de Cellule E-UTRAN identifie de manière unique la cellule servant l'UE

Format : ID eNodeB (20 bits) + ID de Cellule (8 bits)

Plage : 0 - 268,435,455

Exemple :

```
5678
```

Source : Informations de localisation de l'UE provenant de MME

Utilisation :

- Identifie une tour de cellule et un secteur spécifiques
- Utilisé pour le transfert et la gestion de mobilité

- Informations de localisation granulaires

---

11. sgw_ip (IP du Plan de Contrôle SGW)

Type : Chaîne (adresse IPv4 ou IPv6)

Description : Adresse IP du Plan de Contrôle S5/S8 de SGW-C (F-TEID)

Format : Décimal pointé (IPv4) ou hexadécimal à deux-points (IPv6)

Exemple :

```
10.0.0.15       (IPv4)
2001:db8::15    (IPv6)
```

Source : Configuration locale, assignée à l'interface S5/S8

---

12. ue_ip (Adresse IP de l'UE)

Type : Chaîne (format IPv4|IPv6)

Description : Adresse IP assignée à l'UE pour la connexion PDN

Format : `<ipv4>|<ipv6>`

Exemples :

```
172.16.1.100|                      (IPv4 uniquement)
|2001:db8::1                       (IPv6 uniquement)
172.16.1.100|2001:db8::1          (Dual-stack)
```

Source : Allocation d'Adresse PDN (PAA) de PGW-C

Remarques :

- IPv4 vide : Pas d'adresse IPv4 allouée

- IPv6 vide : Pas d'adresse IPv6 allouée
- Les deux présents : connexion PDN en double pile

---

13. pgw_ip (IP du Plan de Contrôle PGW)

Type : Chaîne (adresse IPv4 ou IPv6)

Description : Adresse IP du Plan de Contrôle S5/S8 de PGW-C (F-TEID distant)

Format : Décimal pointé (IPv4) ou hexadécimal à deux-points (IPv6)

Exemple :

```
10.0.0.20       (IPv4)
2001:db8::20    (IPv6)
```

Source : Reçu dans la Réponse de Création de Session de PGW-C

---

14. apn (Nom du Point d'Accès)

Type : Chaîne (jusqu'à 100 caractères)

Description : Nom du Point d'Accès identifiant le réseau externe (PDN)

Format : Format de label semblable à DNS

Exemples :

```
internet
ims
mms
enterprise.corporate
```

Source : Reçu dans la Demande de Création de Session de MME

Utilisation :

- Détermine à quel réseau externe se connecter
- Conduit les règles de politique et de facturation
- Peut déterminer le pool d'adresses IP

---

15. qci (Identifiant de Classe QoS)

Type : Entier non signé 8 bits

Description : L'Identifiant de Classe QoS définit la qualité de service du porteur

Plage : 1 - 9 (standardisé), 128-254 (spécifique à l'opérateur)

Valeurs QCI Standardisées :

QCI	Type de Ressource	Priorité	Délai de Paquet	Perte de Paquet	Service Exemple
1	GBR	2	100 ms	10^-2	Voix Conversationnelle
2	GBR	4	150 ms	10^-3	Vidéo Conversationnelle
3	GBR	3	50 ms	10^-3	Jeux en Temps Réel
4	GBR	5	300 ms	10^-6	Vidéo Non-Conversationnelle
5	Non-GBR	1	100 ms	10^-6	Signalisation IMS
6	Non-GBR	6	300 ms	10^-6	Vidéo (tamponnée)
7	Non-GBR	7	100 ms	10^-3	Voix, Vidéo, Jeux
8	Non-GBR	8	300 ms	10^-6	Vidéo (tamponnée)
9	Non-GBR	9	300 ms	10^-6	Porteur par Défaut

Exemple :

```
9  → Porteur par défaut (meilleur effort)
```

Source : Paramètres QoS du porteur de PGW-C

---

16. octets_in (Volume en Aval)

Type : Entier non signé 64 bits

Description : Nombre d'octets transmis dans la direction en aval (réseau → UE)

Unités : Octets

Exemple :

```
1048576  → 1 Mo en aval
```

Source : Mesure de Volume PFCP de PGW-U (via rapports d'utilisation URR)

Remarques :

- Cumulatif pour les événements `update`
- Total final pour les événements `end`
- Toujours 0 pour les événements `start`
- Rapports déclenchés par le seuil de temps URR (configuré via `usage_report_interval`)

---

17. octets_out (Volume en Amont)

Type : Entier non signé 64 bits

Description : Nombre d'octets transmis dans la direction en amont (UE → réseau)

Unités : Octets

Exemple :

```
524288  → 512 Ko en amont
```

Source : Mesure de Volume PFCP de PGW-U (via rapports d'utilisation URR)

Remarques :

- Cumulatif pour les événements `update`
- Total final pour les événements `end`
- Toujours 0 pour les événements `start`
- Rapports déclenchés par le seuil de temps URR (configuré via `usage_report_interval`)

Exemples

Exemple 1 : Session de Base avec Mise à Jour Unique

Chronologie :

1. Porteur établi
2. 5 minutes plus tard : Mise à jour d'utilisation (10 Mo en aval, 5 Mo en amont)
3. Session terminée

Sortie CDR :

```
# Fichier CDR de Données :
# Heure de Début du Fichier : 10:00:00 (1726570800)
# Heure de Fin du Fichier : 11:00:00 (1726574400)
# Nom de la Passerelle : pgw-c-01
# epoch,imsi,event,charging_id,msisdn,ue_imei,timezone_raw,plmn,tac,e
1726570800,310260111111111,default_bearer_start,10001,15551111111,111
1726571100,310260111111111,default_bearer_update,10001,15551111111,11
1726571400,310260111111111,default_bearer_end,10001,15551111111,11111
```

Exemple 2 : Session Double Pile avec Plusieurs Mises à Jour

Chronologie :

1. Porteur double pile établi (IPv4 + IPv6)
2. Multiples mises à jour d'utilisation
3. Session terminée

Sortie CDR :

```
1726570800,310260222222222,default_bearer_start,10002,15552222222,222
1726571100,310260222222222,default_bearer_update,10002,15552222222,22
1726571400,310260222222222,default_bearer_update,10002,15552222222,22
1726571700,310260222222222,default_bearer_update,10002,15552222222,22
1726572000,310260222222222,default_bearer_end,10002,15552222222,22222
```

Exemple 3 : Session avec Porteur Dédicacé

Chronologie :

1. Porteur par défaut établi (QCI 9)
2. Porteur dédié créé pour la vidéo (QCI 6)
3. Mises à jour d'utilisation pour les deux porteurs
4. Porteur dédié supprimé
5. Porteur par défaut terminé

Sortie CDR :

```
1726570800,310260333333333,default_bearer_start,10003,15553333333,333
1726571100,310260333333333,dedicated_bearer_start,10004,15553333333,3
1726571400,310260333333333,default_bearer_update,10003,15553333333,33
1726571400,310260333333333,dedicated_bearer_update,10004,15553333333,
1726571700,310260333333333,dedicated_bearer_end,10004,15553333333,333
1726572000,310260333333333,default_bearer_end,10003,15553333333,33333
```

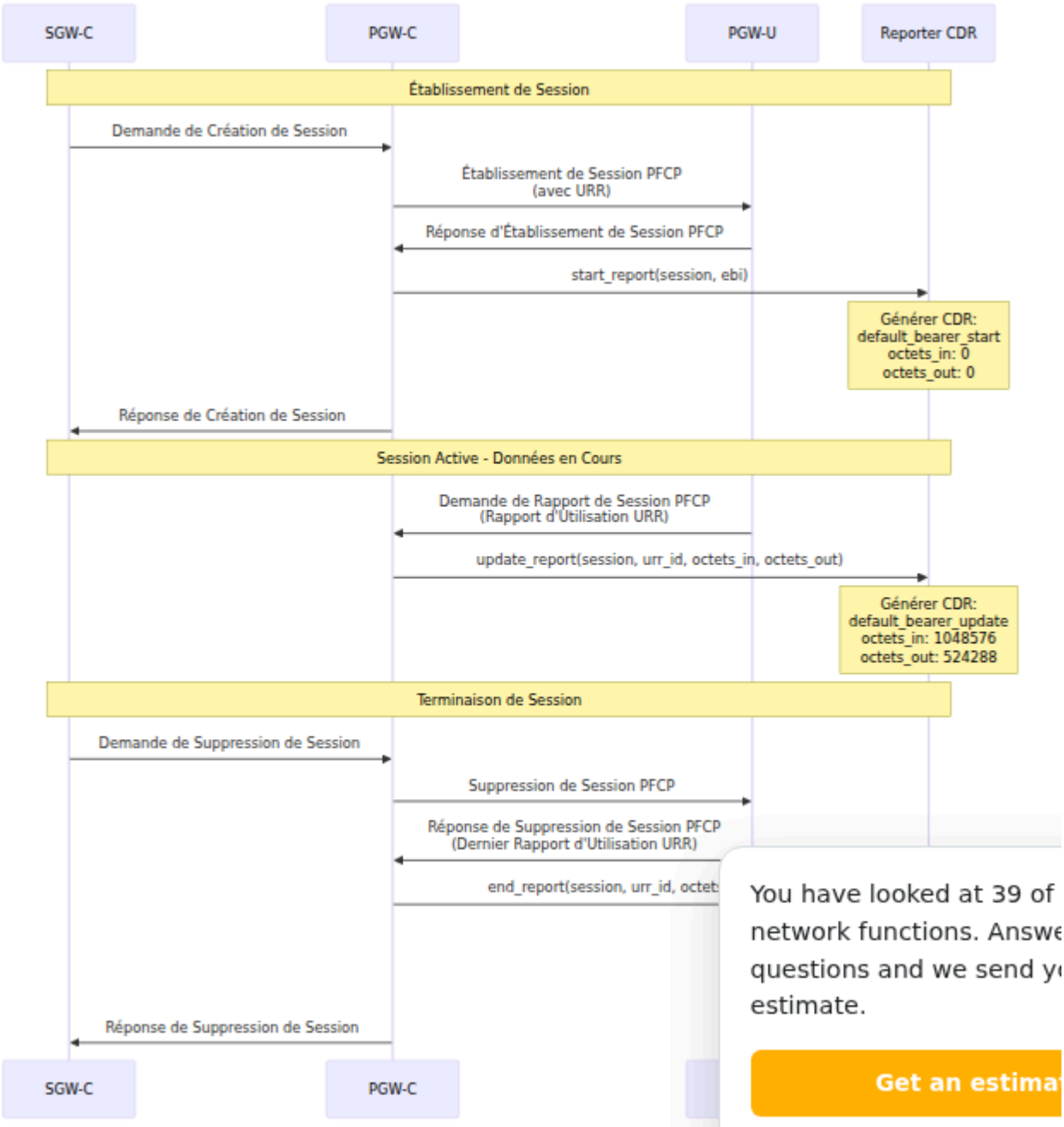
Analyse :

- Le porteur par défaut (10003) transporte le trafic de fond (10 Mo en aval, 4 Mo en amont)
- Le porteur dédié (10004) transporte le trafic vidéo (200 Mo en aval, 2 Mo en amont)
- Différentes valeurs QCI (9 contre 6) reflètent un traitement QoS différent

---

Intégration

Pipeline de Traitement CDR



Méthodes de Collecte CDR

1. Collecte Basée sur Fichier :

```
# Surveiller le répertoire CDR (PGW-C)
inotifywait -m /var/log/pgw_c/cdrs/ -e close_write | while read
path action file; do
    # Rotation de fichier terminée, traiter le CDR
    process_cdr "$path$file"
done
```

2. Streaming en Temps Réel :

```
# Suivre et diffuser vers le pipeline de traitement
tail -F /var/log/pgw_c/cdrs/* | process_cdr_stream
```

Documentation Connexe

- Gestion de Session - Cycle de vie de la session et déclencheurs CDR
- Interface PFCP - Rapport d'utilisation de PGW-U via URRs
- Guide de Surveillance - Métriques de génération CDR et alertes
- Guide de Configuration - Paramètres de configuration CDR et URR
- Interface Diameter Gx - Contrôle de politique pour les valeurs QCI dans les CDR
- Interface Diameter Gy - Intégration de facturation en ligne

---

Références 3GPP

- TS 32.251 - Facturation du domaine de Paquet (PS)
- TS 29.274 - Système de Paquet Évolué 3GPP (EPS) ; protocole GTP-C
- TS 29.244 - Interface entre les nœuds CP et UP (PFCP) - **Support URR**
- TS 32.298 - Encodage CDR

Format CDR - *Enregistrements de Facturation Hors Ligne pour PGW-C*

Développé par Omnitouch Network Services

Version de Documentation : 1.0 **Dernière Mise à Jour :** 2025-12-28

Documentation de l'Interface Gx de Diameter

Interface de la Fonction de Règles de Politique et de Facturation (PCRF)

Table des Matières

1. Aperçu
2. Notions de Base de l'Interface Gx
3. Protocole Diameter
4. Messages de Contrôle de Crédit
5. Règles de Politique et de Facturation
6. Rapport de Changement de Localisation
7. Configuration
8. Flux de Messages
9. Gestion des Erreurs
10. Dépannage

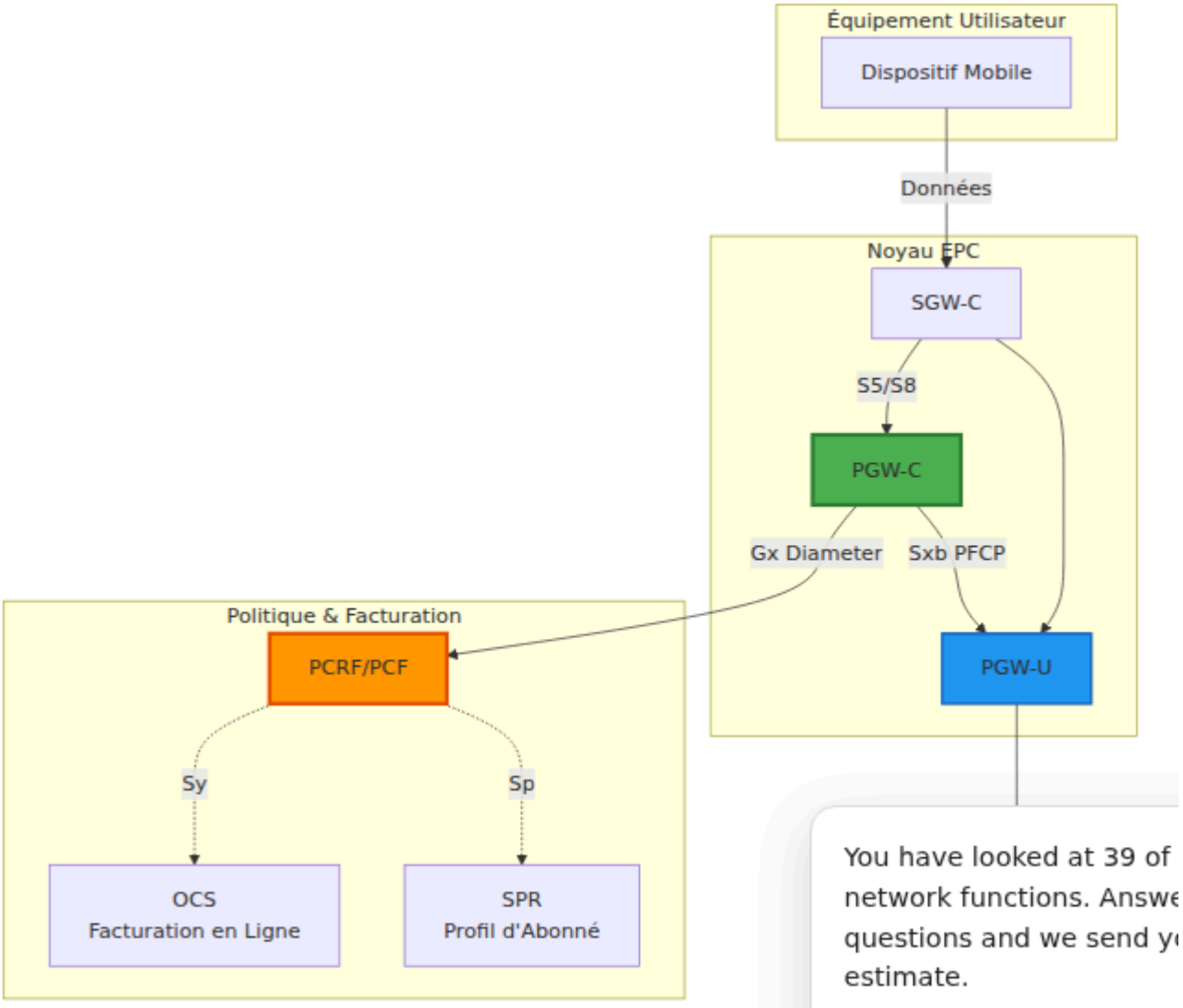
Aperçu

L'**interface Gx** connecte le PGW-C à la **PCRF (Fonction de Règles de Politique et de Facturation)** ou **PCF (Fonction de Contrôle de Politique)** dans les réseaux 5G. Cette interface permet :

- **Contrôle de Politique Dynamique** - Application en temps réel de la QoS et des politiques

- **Contrôle de Facturation** - Autorisation de crédit et suivi de l'utilisation
- **Connaissance du Service** - Différenciation du trafic au niveau de l'application
- **Gestion des Profils d'Abonnés** - Application de politiques par utilisateur

Gx dans l'Architecture Réseau



Fonctions Clés

Fonction	Description
Provisionnement de Politique	La PCRF fournit des règles PCC définissant comment gérer le trafic
Contrôle de QoS	Ajustement dynamique des débits et des paramètres de QoS
Contrôle de Facturation	Autorisation de crédit pour les scénarios prépayés/postpayés
Contrôle de Gating	Activer/désactiver les flux de trafic en fonction de la politique
Surveillance de l'Utilisation	Suivi de la consommation de données par service

---

Notions de Base de l'Interface Gx

Référence 3GPP

- **Spécification :** 3GPP TS 29.212
- **ID d'Application Diameter :** 16777238 (Gx)
- **Protocole :** Protocole de Base Diameter (RFC 6733)

Concept de Session

Chaque connexion PDN de l'UE a une **session Gx** correspondante identifiée par un **Session-ID**. Cette session :

- Créée lorsque l'UE s'attache (CCR-Initial)

- Mise à jour pendant la durée de la connexion (CCR-Update) - optionnelle, par exemple, changements de politique ou rapport de changement de localisation
- Terminée lorsque l'UE se détache (CCR-Termination)

Format de l'ID de Session

```
Session-ID: <Origin-Host>;<high32>;<low32>[;<optional>]
Exemple: omni-
pgw_c.epc.mnc999.mcc999.3gppnetwork.org;1234567890;98765
```

Composants :

- **Origin-Host :** Identité Diameter du PGW-C
- **high32 :** 32 bits supérieurs de l'identifiant unique
- **low32 :** 32 bits inférieurs de l'identifiant unique

Protocole Diameter

Structure des Messages

Les messages Diameter sont encodés en binaire avec la structure suivante :

```
En-tête Diameter (20 octets)
├── Version (1 octet) = 1
├── Longueur du Message (3 octets)
├── Drapeaux (1 octet)
│   ├── R : Demande (1) / Réponse (0)
│   ├── P : Proxiable
│   ├── E : Erreur
│   └── T : Potentiellement retransmis
├── Code de Commande (3 octets)
├── ID d'Application (4 octets) = 16777238 (Gx)
├── ID Hop-by-Hop (4 octets)
└── ID End-to-End (4 octets)

AVPs (Attribut-Valeur Paires)
├── En-tête AVP
│   ├── Code AVP
│   ├── Drapeaux (V, M, P)
│   ├── Longueur AVP
│   └── ID de Fournisseur (optionnel)
└── Données AVP
```

Concepts Clés de Diameter

AVP (Attribut-Valeur Paire) :

- Unité de données de base dans Diameter
- Contient un code, des drapeaux et une valeur
- Peut être imbriqué (AVP Groupé)

Commande :

- Paire Demande/Réponse
- CCR (Demande de Contrôle de Crédit) / CCA (Réponse de Contrôle de Crédit)

Codes de Résultat :

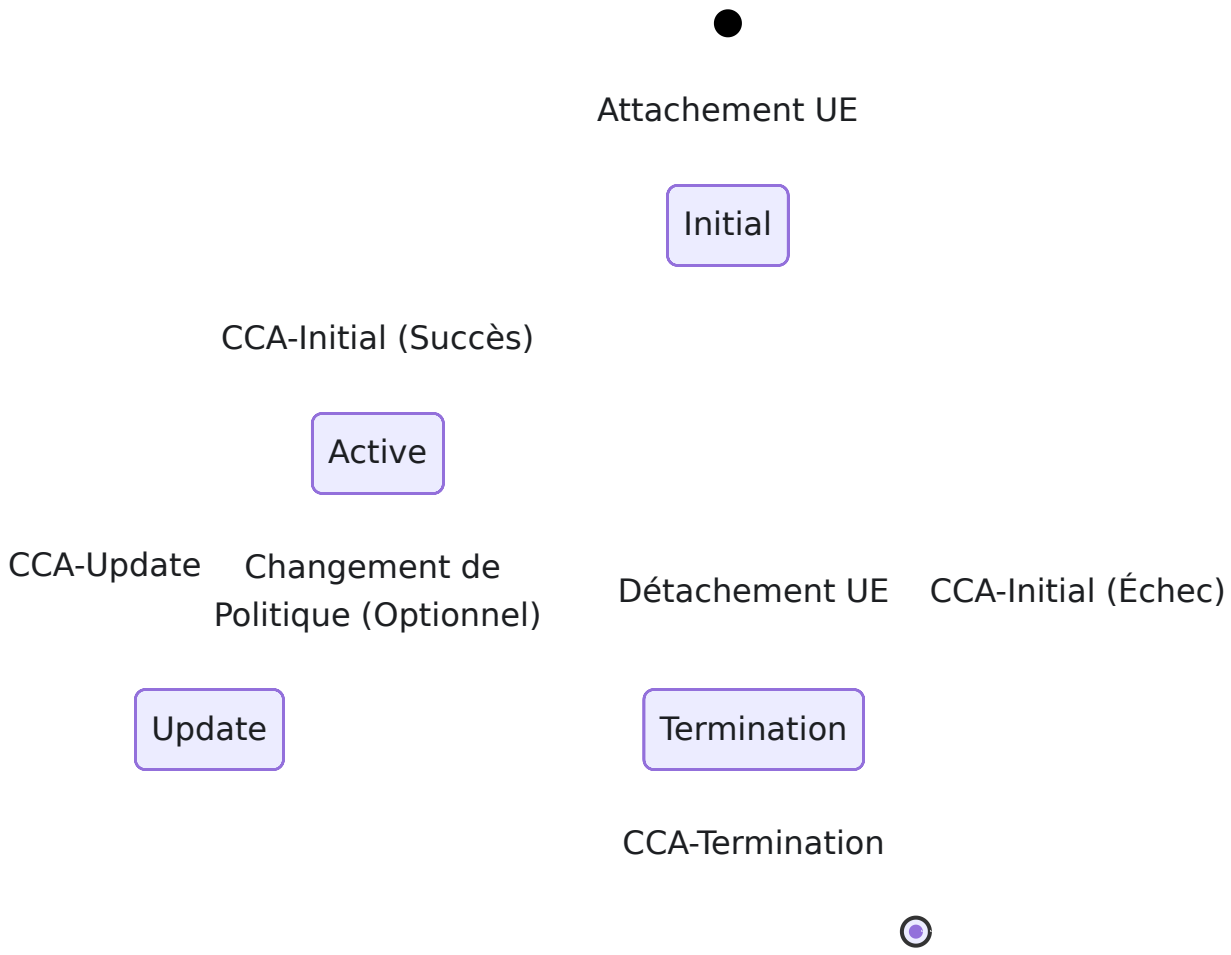
- `2001` - DIAMETER_SUCCESS
- `3xxx` - Erreurs de protocole

- `4xxx` - Échecs transitoires
- `5xxx` - Échecs permanents

Messages de Contrôle de Crédit

Le PGW-C utilise l'**Application de Contrôle de Crédit Diameter** (RFC 4006) pour Gx.

Types de Messages



CCR-Initial (Demande de Contrôle de Crédit - Initial)

Quand : L'UE crée une nouvelle connexion PDN

But :

- Demander des règles de politique et de facturation initiales
- Fournir le contexte de l'UE et du réseau à la PCRF
- Obtenir des paramètres de QoS et une autorisation de facturation

AVPs Clés Envoyés par PGW-C :

Nom de l'AVP	Code de l'AVP	Type	Description
Session-Id	263	UTF8String	Identifiant de session Gx unique
Auth-Application-Id	258	Unsigned32	16777238 (Gx)
Origin-Host	264	DiamIdent	Identité Diameter du PGW-C
Origin-Realm	296	DiamIdent	Domaine Diameter du PGW-C
Destination-Realm	283	DiamIdent	Domaine de la PCRF
CC-Request-Type	416	Enumerated	1 = INITIAL_REQUEST
CC-Request-Number	415	Unsigned32	Numéro de séquence (commence à 0)
Subscription-Id	443	Grouped	Identifiant de l'UE (IMSI/MSISDN)
Called-Station-Id	30	UTF8String	Nom de l'APN
Framed-IP-Address	8	OctetString	Adresse IPv4 allouée à l'UE (PDNs IPv4 et IPv4v6)
Framed-IPv6-Prefix	97	OctetString	Préfixe IPv6 alloué à l'UE (PDNs uniquement IPv6)
3GPP-User-Location-Info	22	OctetString	ULI initial de l'abonné (TS 29.061)

Nom de l'AVP	Code de l'AVP	Type	Description
User-Equipment-Info	458	Grouped	Identité de l'UE (par exemple, IMEISV), si disponible
IP-CAN-Type	1027	Enumerated	5 = 3GPP-EPS
RAT-Type	1032	Enumerated	Mappé à partir du type RAT GTP (par exemple, 1004 = EUTRAN)
QoS-Information	1016	Grouped	QoS actuelle (APN-AMBR UL/DL)
Default-EPS-Bearer-QoS	1049	Grouped	QCI/ARP du porteur par défaut (QCI 5)
Network-Request-Support	1024	Enumerated	Procédures initiées par le réseau
Supported-Features	628	Grouped	Liste des fonctionnalités Gx (Rel-8/9/10 Gx)

> L'AVP framed-address est sélectionné par la famille d'adresses de l'UE : `Framed-IP-Address` pour IPv4 (et l'IPv4 d'un PDN IPv4v6), `Framed-IPv6-Prefix` pour un IPv6 uniquement PDN. Pour IPv4v6, le préfixe IPv6 n'est pas transporté sur Gx. Voir IPv6 / Dual-Stack.

Exemple de Structure CCR-I :

```
CCR (Code de Commande : 272, Demande)
├── Session-Id: "pgw_c.example.com;123;456"
├── Auth-Application-Id: 16777238
├── Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├── Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── CC-Request-Type: INITIAL_REQUEST (1)
├── CC-Request-Number: 0
├── Subscription-Id (Grouped)
│   ├── Subscription-Id-Type: END_USER_IMSI (1)
│   └── Subscription-Id-Data: "310260123456789"
├── Called-Station-Id: "internet"
├── Framed-IP-Address: 100.64.1.42
├── IP-CAN-Type: 3GPP-EPS (5)
├── RAT-Type: EUTRAN (1004)
├── QoS-Information (Grouped)
│   ├── APN-Aggregate-Max-Bitrate-UL: 100000000 (100 Mbps)
│   └── APN-Aggregate-Max-Bitrate-DL: 50000000 (50 Mbps)
├── Network-Request-Support: 1
└── Supported-Features: [...]
```

CCA-Initial (Réponse de Contrôle de Crédit - Initial)

Envoyé par : PCRF en réponse à CCR-I

But :

- Autoriser ou rejeter la session
- Fournir des règles PCC pour la gestion du trafic
- Spécifier les paramètres de QoS

AVPs Clés Reçus par PGW-C :

Nom de l'AVP	Code de l'AVP	Description
Result-Code	268	Succès (2001) ou code d'erreur
Experimental-Result	297	Codes de résultat spécifiques au fournisseur
QoS-Information	1016	QoS autorisée (peut différer de la demande)
Charging-Rule-Install	1001	Règles PCC à activer
Charging-Rule-Definition	1003	Définitions de règles en ligne
Default-EPS-Bearer-QoS	1049	QoS pour le porteur par défaut

Exemple de Réponse de Succès :

```
CCA (Code de Commande : 272, Réponse)
├── Session-Id: "pgw_c.example.com;123;456"
├── Result-Code: DIAMETER_SUCCESS (2001)
├── Origin-Host: "pcrf.example.com"
├── Origin-Realm: "example.com"
├── Auth-Application-Id: 16777238
├── CC-Request-Type: INITIAL_REQUEST (1)
├── CC-Request-Number: 0
├── QoS-Information (Grouped)
│   ├── APN-Aggregate-Max-Bitrate-UL: 50000000 (50 Mbps - réduit)
│   └── APN-Aggregate-Max-Bitrate-DL: 100000000 (100 Mbps -
augmenté)
├── Charging-Rule-Install (Grouped)
│   ├── Charging-Rule-Name: "default_internet_rule"
│   └── Charging-Rule-Name: "video_streaming_rule"
└── Charging-Rule-Definition (Grouped)
    ├── Charging-Rule-Name: "default_internet_rule"
    ├── QoS-Information: {...}
    └── Precedence: 1000
```

CCR-Termination (Demande de Contrôle de Crédit - Terminaison)

Quand : L'UE se détache ou la connexion PDN est supprimée

But :

- Notifier la PCRF de la terminaison de session
- Enregistrement final de comptabilité/facturation

Différences Clés par Rapport à CCR-I :

- `CC-Request-Type: TERMINATION_REQUEST (3)`
- Peut inclure des statistiques d'utilisation
- Ensemble AVP simplifié

Exemple CCR-T :

```
CCR (Code de Commande : 272, Demande)
├── Session-Id: "pgw_c.example.com;123;456"
├── Auth-Application-Id: 16777238
├── Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├── Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── CC-Request-Type: TERMINATION_REQUEST (3)
├── CC-Request-Number: 1
└── Termination-Cause: DIAMETER_LOGOUT (1)
```

CCA-Termination

Envoyé par : PCRF en réponse à CCR-T

But :

- Accuser réception de la terminaison de session
- Aucune règle de politique retournée

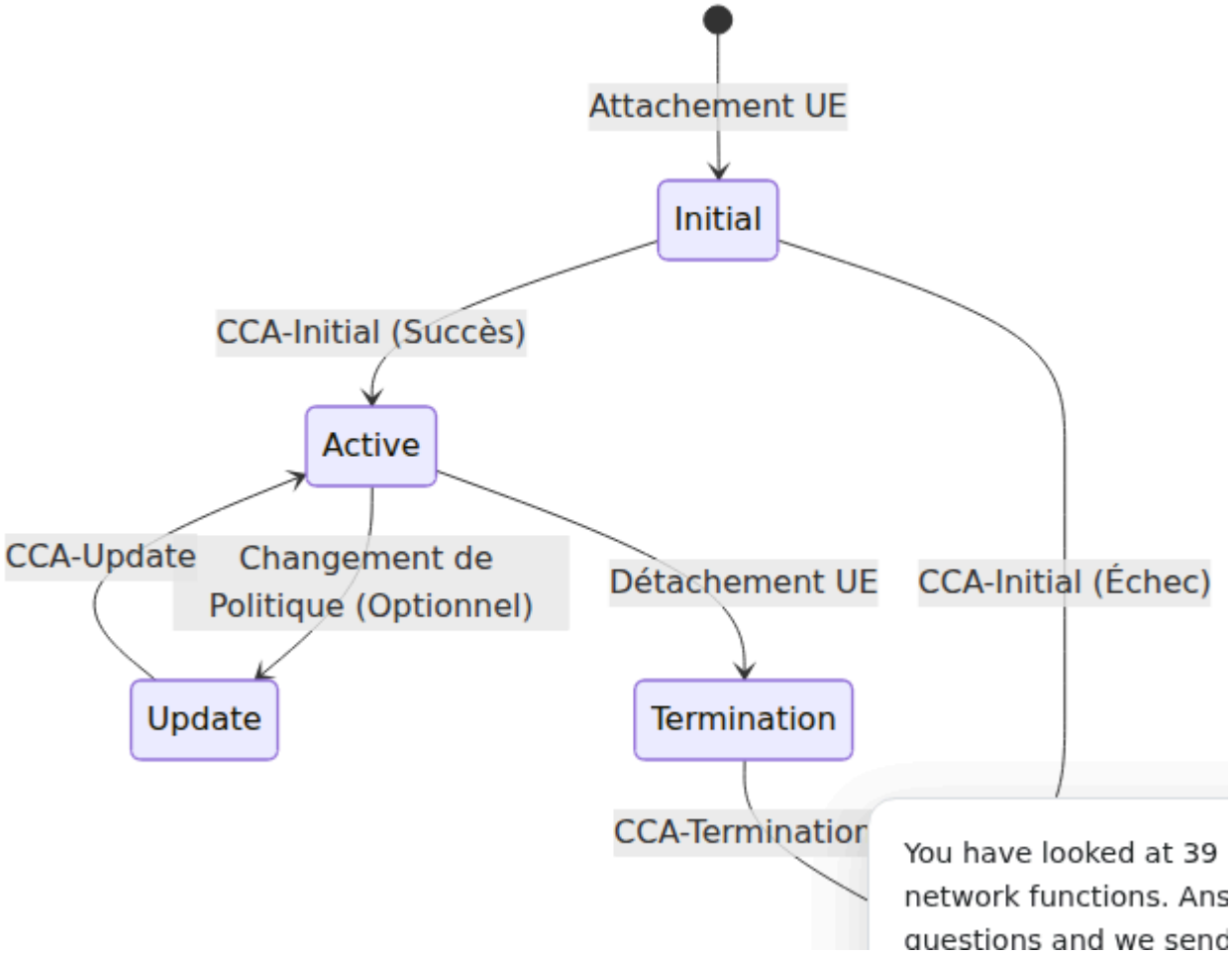
Exemple CCA-T :

```
CCA (Code de Commande : 272, Réponse)
├── Session-Id: "pgw_c.example.com;123;456"
├── Result-Code: DIAMETER_SUCCESS (2001)
├── Origin-Host: "pcrf.example.com"
├── Origin-Realm: "example.com"
├── Auth-Application-Id: 16777238
├── CC-Request-Type: TERMINATION_REQUEST (3)
└── CC-Request-Number: 1
```

Règles de Politique et de Facturation

Structure de la Règle PCC

Une **Règle PCC (Contrôle de Politique et de Facturation)** définit comment gérer des flux de trafic spécifiques :



Composants de la Règle

1. Nom de la Règle :

- Identifiant unique pour la règle
- Exemple : `"video_streaming_rule"`

2. Précedence :

- Un nombre plus bas = une priorité plus élevée
- Plage : 0-65535
- Utilisé lorsque plusieurs règles correspondent

3. Filtres de Flux (TFT - Modèle de Flux de Trafic) :

- Définit quels paquets correspondent à cette règle
- Exemples :
  - IP 5-tuple : Protocole, IP Src/Dst, Port Src/Dst
  - `"permit out ip from any to 8.8.8.8 80"`

4. Informations de QoS :

- **QCI (Identifiant de Classe de QoS) :** 1-9 (standardisé), 128-254 (spécifique à l'opérateur)
  - QCI 1 : Voix Conversationnelle
  - QCI 5 : Signalisation IMS
  - QCI 9 : Internet par D��faut
- **ARP (Priorité d'Allocation et de Rétention) :** Capacité de préemption
- **MBR/GBR :** Débits Maximaux/Garantis

5. Informations de Facturation :

- **Groupe de Tarification :** Identifie la catégorie de facturation (utilisé par OCS - voir Interface Diameter Gy)
- **Méthode de Mesure :** Volume, temps ou basé sur des événements
- **Facturation en Ligne/Hors Ligne :** OCS (prépayé via Diameter Gy) vs. CDRs hors ligne (postpayé - voir Format CDR de Données)

6. Statut de Gating :

- **OUVERT :** Autoriser le trafic
- **FERMÉ :** Bloquer le trafic

Provisionnement Dynamique de Règles

La PCRF peut fournir des règles de deux manières :

1. Règles Prédéfinies (par nom) :

```
Charging-Rule-Install (Grouped)
├── Charging-Rule-Name: "gold_subscriber_internet"
└── Charging-Rule-Name: "video_qos_boost"
```

2. Règles Dynamiques (définition en ligne) :

```
Charging-Rule-Definition (Grouped)
├── Charging-Rule-Name: "dynamic_rule_123"
├── Precedence: 100
├── Flow-Information (Grouped)
│   ├── Flow-Description: "permit out ip from any to 192.0.2.0/24"
│   └── Flow-Direction: DOWNLINK
├── QoS-Information (Grouped)
│   ├── QoS-Class-Identifier: 5
│   ├── Max-Requested-Bandwidth-UL: 10000000
│   └── Max-Requested-Bandwidth-DL: 50000000
└── Rating-Group: 1000
```

AVP d'Informations QoS

APN-AMBR (Débit Maximum Agrégé) :

S'applique à tous les porteurs non-GBR pour cet APN :

```
QoS-Information (Grouped)
├── APN-Aggregate-Max-Bitrate-UL: 100000000  # 100 Mbps
└── APN-Aggregate-Max-Bitrate-DL: 200000000  # 200 Mbps
```

Réponse PGW-C :

- Met à jour l'état interne AMBR
- Envoie une Demande de Modification de Session au PGW-U avec le QER mis à jour

---

Rapport de Changement de Localisation

Le PGW-C peut notifier la PCRF lorsqu'une **localisation**, **technologie d'accès radio (RAT)**, **PLMN de service** ou **fuseau horaire de l'UE** change pendant la durée d'une session. Le changement est transporté dans un **CCR-Update** en tant que **3GPP-User-Location-Info** et AVPs connexes.

Le rapport est opt-in (contrôlé par la PCRF)

Ce rapport est **conditionnel à la demande de la PCRF** — exactement comme requis par 3GPP TS 29.212 §4.5.7 / §5.3.7. La PCRF *arme* les événements dont elle souhaite être informée en incluant un ou plusieurs AVPs **Event-Trigger** dans un **CCA** (ou dans un **RAR**). Le PGW-C stocke l'ensemble armé par session Gx et n'envoie un CCR-Update que lorsque :

1. la condition d'un Event-Trigger armé se produit réellement, **et**
2. une session Gx existe pour l'abonné.

Si la PCRF n'arme **aucun** Event-Trigger lié à la localisation, le PGW-C reste silencieux sur les changements de localisation — il met simplement à jour son état interne et répond à la Demande de Modification de Porteur GTP-C. Cela garde l'interface silencieuse pour les PCRF qui ne se soucient pas de la mobilité, et conforme pour ceux qui le font.

L'ensemble armé peut changer à tout moment pendant la session. Le PGW-C le met à jour à partir de chaque ensemble AVP Event-Trigger que la PCRF envoie — dans le CCA-Initial, dans un CCA-Update, ou dans un **RAR** initié par la PCRF. Un ensemble Event-Trigger présent remplace l'ensemble armé actuel ; un message qui ne transporte pas d'AVP Event-Trigger laisse l'ensemble armé inchangé. Ainsi, une PCRF peut commencer une session sans rapport de localisation et plus tard l'activer (ou la désactiver) avec un RAR, et vice-versa.

Event-Triggers Supportés

Event-Trigger	Valeur	Se déclenche lorsque…	AVP Rapporté
`RAT_CHANGE`	2	Changements de RAT-Type (par exemple, EUTRAN → UTRAN)	`RAT-Type`
`PLMN_CHANGE`	4	Changements de réseau de service (MCC/MNC)	`3GPP-SGSN-MCC-MNC`
`USER_LOCATION_CHANGE`	13	Tout changement dans les Informations de Localisation de l'Utilisateur	`3GPP-User-Location-Info`
`UE_TIME_ZONE_CHANGE`	25	Changements de fuseau horaire de l'UE	`3GPP-MS-TimeZone`
`TAI_CHANGE`	26	Changements d'Identité de Zone de Suivi	`3GPP-User-Location-Info`
`ECGI_CHANGE`	27	Changements d'Identité Globale de Cellule E-UTRAN	`3GPP-User-Location-Info`

Plusieurs déclencheurs peuvent se produire en même temps (par exemple, passer à une nouvelle cellule dans une nouvelle zone de suivi déclenche `USER_LOCATION_CHANGE`, `TAI_CHANGE` **et** `ECGI_CHANGE`) ; tous les déclencheurs armés et déclenchés sont listés dans les AVPs `Event-Trigger` du CCR-Update.

D'où provient la localisation

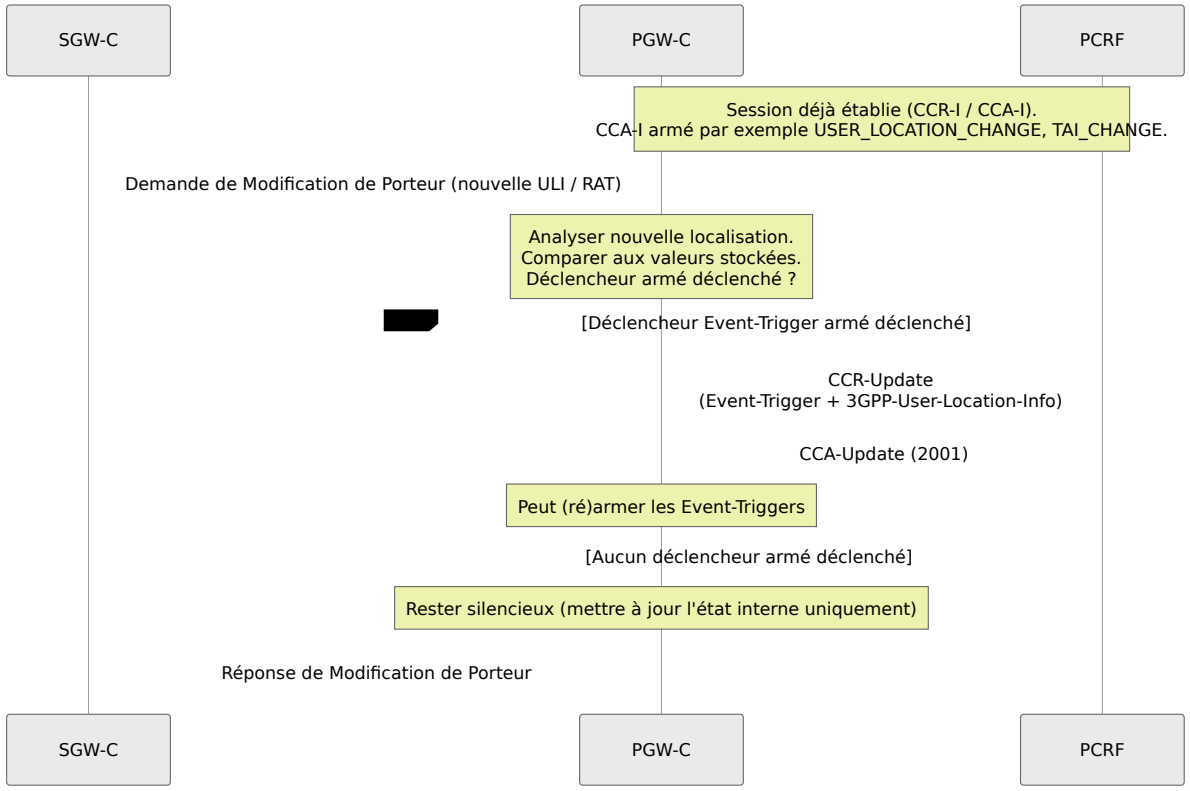
- **CCR-Initial** inclut maintenant l'initial `3GPP-User-Location-Info` de l'abonné, donnant à la PCRF une base de référence.
- Pendant la session, une **Demande de Modification de Porteur GTP-C** (S5/S8) transporte les Informations de Localisation de l'Utilisateur mises à jour, RAT-Type, Réseau de Service et Fuseau Horaire de l'UE. Le PGW-C les analyse, les compare aux valeurs stockées et rapporte tout changement armé.
- Les IEs **absents** d'une Demande de Modification de Porteur conservent leur valeur précédente, donc une Demande de Modification de Porteur partielle ne ressemble jamais à un changement de localisation spurious.

Encodage de 3GPP-User-Location-Info

L'AVP est encodé selon **3GPP TS 29.061 §16.4.7.2** : un type de *Localisation Géographique* d'un octet suivi des champs de localisation, avec des identités PLMN encodées selon 3GPP TS 24.008 §10.5.1.3. Types supportés :

Type	Valeur	Contenu
CGI	0	PLMN + LAC + CI (2G)
SAI	1	PLMN + LAC + SAC (3G)
RAI	2	PLMN + LAC + RAC
TAI	128	PLMN + TAC (LTE)
ECGI	129	PLMN + ECI (LTE)
TAI et ECGI	130	PLMN + TAC + PLMN + ECI (LTE, préféré)

Flux de Messages



> **Robustesse :** rapporter le changement à la PCRF ne bloque jamais la Réponse de Modification de Porteur GTP-C. Si le CCR-Update expire ou si la PCRF renvoie une erreur, le PGW-C le consigne et répond toujours au SGW-C — le chemin de données de l'abonné n'est pas retardé par un échec de rapport de politique.

Inspection des déclencheurs armés via l'API

L'API des Sessions (`GET /api/sessions` et `GET /api/sessions/:imsi`) renvoie un dump complet, sûr pour JSON, de la session sous la clé `state`, en plus des champs de niveau supérieur soigneusement sélectionnés. Les Event-Triggers armés par la PCRF sont visibles à `state.gx.requested_event_triggers`, aux côtés du reste de l'état de la session Gx (`session_id`, `cc_request_number`, hôte/domaine de destination) et l'état actuel `state.uli`. Cela permet de confirmer à partir des outils d'opérations si un abonné est effectivement abonné au rapport de localisation.

Surcharge des déclencheurs armés via l'API

L'ensemble armé peut également être modifié directement sur une session active avec **`PATCH /api/sessions/:imsi`** (utile pour tester la fonctionnalité sans PCRF, ou pour forcer le rapport activé/désactivé). Le corps est **fusionné en profondeur dans l'état de la session**, donc le chemin reflète la structure retournée par `GET` — les déclencheurs armés vivent sous `gx` :

```
PATCH /api/sessions/310260000000001
Content-Type: application/json

{ "gx": { "requested_event_triggers": [13, 26, 27] } }
```

- La valeur est une liste de codes Event-Trigger Gx ; elle remplace l'ensemble armé (d'autres champs `gx` sont laissés intacts par la fusion).
- Une liste vide (`[]`) désarme tous les rapports de localisation pour la session.
- La réponse renvoie l'état de session mis à jour et assaini.

Ce point de terminaison n'a **aucune protection** : le corps est fusionné en profondeur dans la session active, donc *tout* champ peut être modifié — y compris à une valeur qui casse la session. Les modifications qui ont des effets en aval sont propagées au UPF. La mise à jour de l'APN-AMBR ou de la QoS du porteur par défaut recompute le QER du porteur par défaut (QCI / MBR / GBR) et le pousse comme un QER de Mise à Jour dans une Modification de Session PFCP :

```
PATCH /api/sessions/310260000000001
Content-Type: application/json

{ "ambr": { "uplink": 200000, "downlink": 200000 } }
```

```
PATCH /api/sessions/310260000000001
Content-Type: application/json

{ "default_bearer_qos": { "qci": 5, "mbr_ul": 300000, "mbr_dl":
250000 } }
```

Lorsque `default_bearer_qos` fournit des MBR/GBR explicites (porteurs GBR), ceux-ci dirigent les limites de taux du QER ; sinon, l'APN-AMBR est utilisé comme MBR du porteur par défaut.

> Remarque : la surcharge des déclencheurs armés change la vue locale du PGW-C sur ce que la PCRF a armé. La PCRF peut toujours réarmer l'ensemble plus tard via un CCA ou un RAR, ce qui écrasera une surcharge manuelle.

Configuration

Configuration de Base de Gx

Modifier `config/runtime.exs` :

```
config :pgw_c,
  diameter: %{
    # Adresse IP à écouter pour les connexions Diameter
    listen_ip: "0.0.0.0",

    # Identité Diameter du PGW-C (Origin-Host)
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",

    # Domaine Diameter du PGW-C (Origin-Realm)
    realm: "epc.mnc999.mcc999.3gppnetwork.org",

    # Liste des pairs PCRF
    peer_list: [
      %{
        # Identité Diameter de la PCRF
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",

        # Domaine PCRF (généralement le même que le domaine PGW-C)
        realm: "epc.mnc999.mcc999.3gppnetwork.org",

        # Adresse IP de la PCRF
        ip: "10.0.0.30",

        # Si le PGW-C initie la connexion à la PCRF
        # true = PGW-C se connecte à la PCRF
        # false = Attendre que la PCRF se connecte
        initiate_connection: true
      }
    ]
  }
```

Plusieurs Pairs PCRF

Pour la redondance ou la distribution géographique :

```
config :pgw_c,
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf-primary.example.com",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.1.30",
        initiate_connection: true
      },
      %{
        host: "pcrf-backup.example.com",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.2.30",
        initiate_connection: true
      }
    ]
  }
```

Équilibrage de Charge :

- Le protocole Diameter gère la sélection des pairs
- Les demandes sont distribuées en fonction de la disponibilité
- Basculement automatique en cas de défaillance d'un pair

Résolution de Noms d'Hôtes

Les Identités Diameter doivent être des FQDNs (Noms de Domaine Entièrement Qualifiés) :

```
# CORRECT - format FQDN
host: "pgw_c.epc.mnc999.mcc999.3gppnetwork.org"

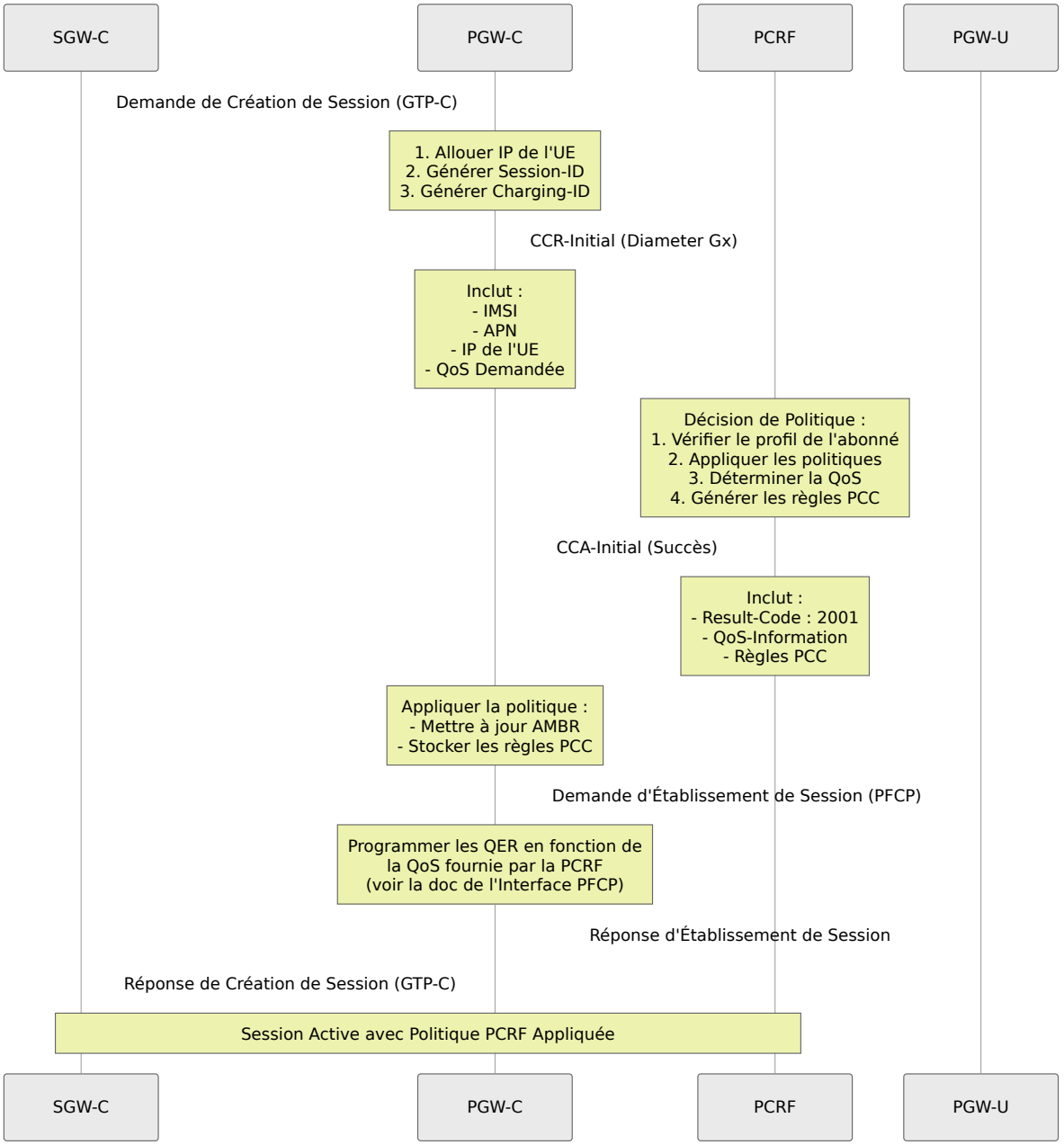
# INCORRECT - Pas une identité Diameter valide
host: "pgw_c"
host: "10.0.0.20"  # Les adresses IP ne sont pas autorisées
```

Format de Domaine :

- Doit être un nom de domaine valide
- Correspond généralement au format PLMN 3GPP : `epc.mncXXX.mccYYY.3gppnetwork.org`

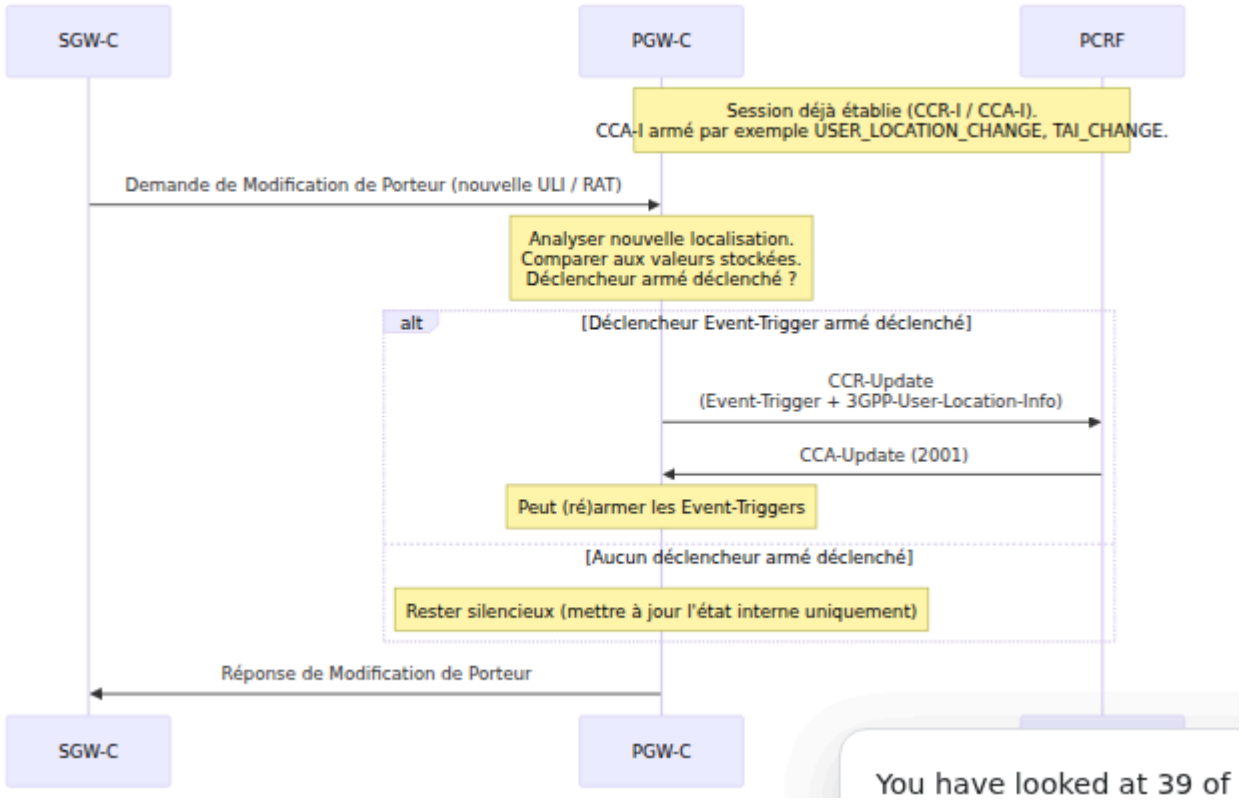
Flux de Messages

Établissement de Session Réussi



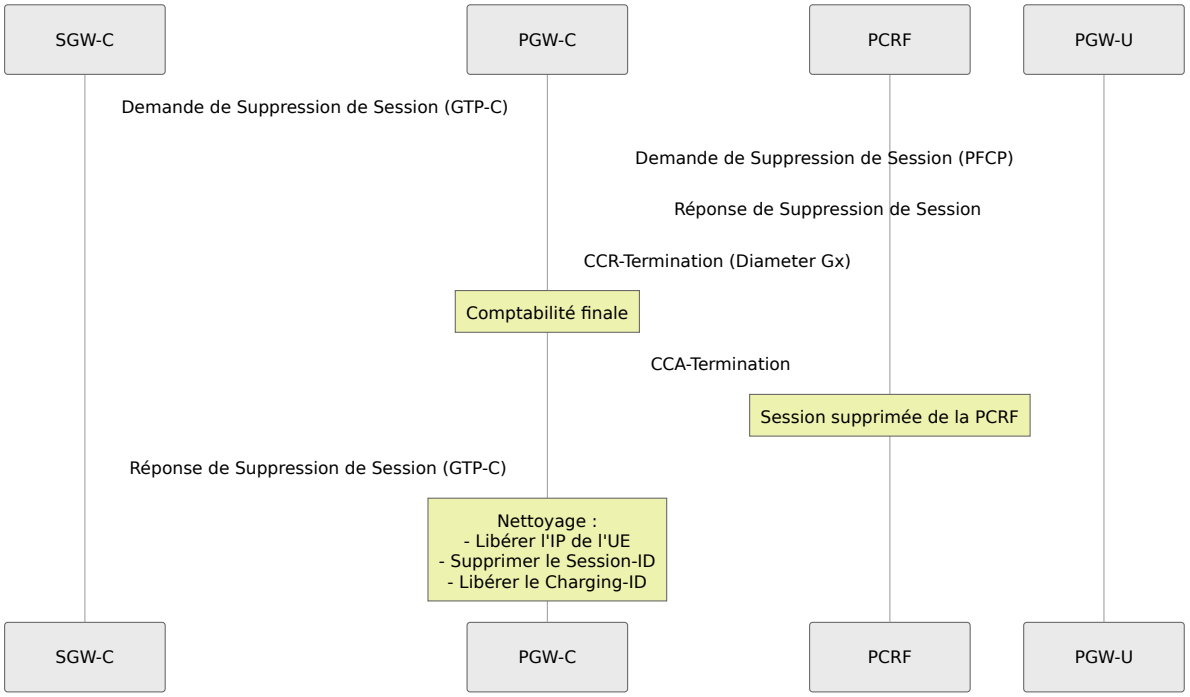
Remarque : Les paramètres de QoS de la PCRF sont traduits en QERs (Règles d'Application de QoS) et programmés dans le PGW-U via PFCP. Voir Interface PFCP pour les détails des QER.

Mise à Jour de Politique (Initiée par le Réseau)



You have looked at 39 of

Terminaison de Session



Gestion des Erreurs

Codes de Résultat

Le PGW-C gère divers codes de résultat Diameter dans les messages CCA :

Codes de Succès :

Code	Nom	Action
2001	DIAMETER_SUCCESS	Continuer l'établissement de la session

Échecs Permanents (5xxx) :

Code	Nom	Action PGW-C
5002	DIAMETER_UNKNOWN_SESSION_ID	Consigner l'erreur, échouer la session
5030	DIAMETER_USER_UNKNOWN	Rejeter la session (Utilisateur Inconnu)
5140	DIAMETER_ERROR_INITIAL_PARAMETERS	Consigner l'erreur, réessayer ou échouer
5003	DIAMETER_AUTHORIZATION_REJECTED	Rejeter la session (Non Autorisé)

Échecs Transitoires (4xxx) :

Code	Nom	Action PGW-C
4001	DIAMETER_AUTHENTICATION_REJECTED	Réessayer ou échouer la session
4010	DIAMETER_TOO_BUSY	Réessayer avec retour
4012	DIAMETER_UNABLE_TO_COMPLY	Consigner l'erreur, peut réessayer

Codes de Résultat Expérimentaux

Codes d'erreur spécifiques au fournisseur :

```
Experimental-Result (Grouped)
├── Vendor-Id: 10415 (3GPP)
└── Experimental-Result-Code: <code spécifique au fournisseur>
```

Codes Expérimentaux 3GPP Communs :

Code	Nom	Signification
5065	IP_CAN_SESSION_NOT_AVAILABLE	La PCRF ne peut pas établir de session
5143	INVALID_SERVICE_INFORMATION	Données de service invalides

Gestion des Délais

Délai CCR-I :

Si la PCRF ne répond pas au CCR-Initial dans le délai :

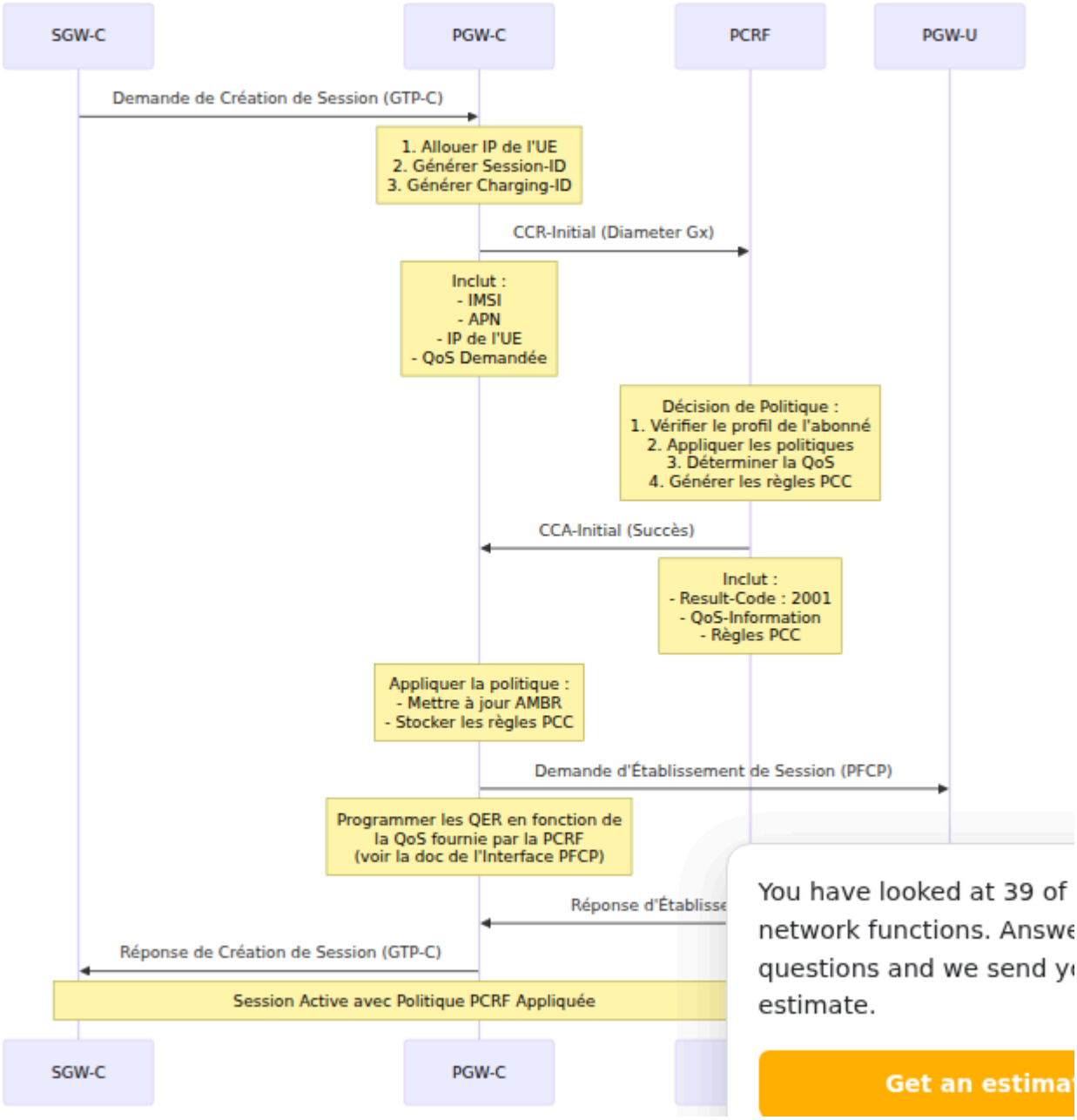
```
1. PGW-C attend le délai configuré (par exemple, 5 secondes)
2. Si aucun CCA reçu :
   - Consigner : "Délai CCR-Initial pour Session-ID : ..."
   - Répondre au SGW-C avec un code d'erreur
   - Nettoyer les ressources allouées
3. SGW-C reçoit : Réponse de Création de Session (Cause : Pair
Distant Non Répondant)
```

Réponse d'Erreur au SGW-C :

Lorsque le CCR-Initial expire, le PGW-C envoie une Réponse de Création de Session avec le code de cause `:remote_peer_not_responding` au SGW-C.

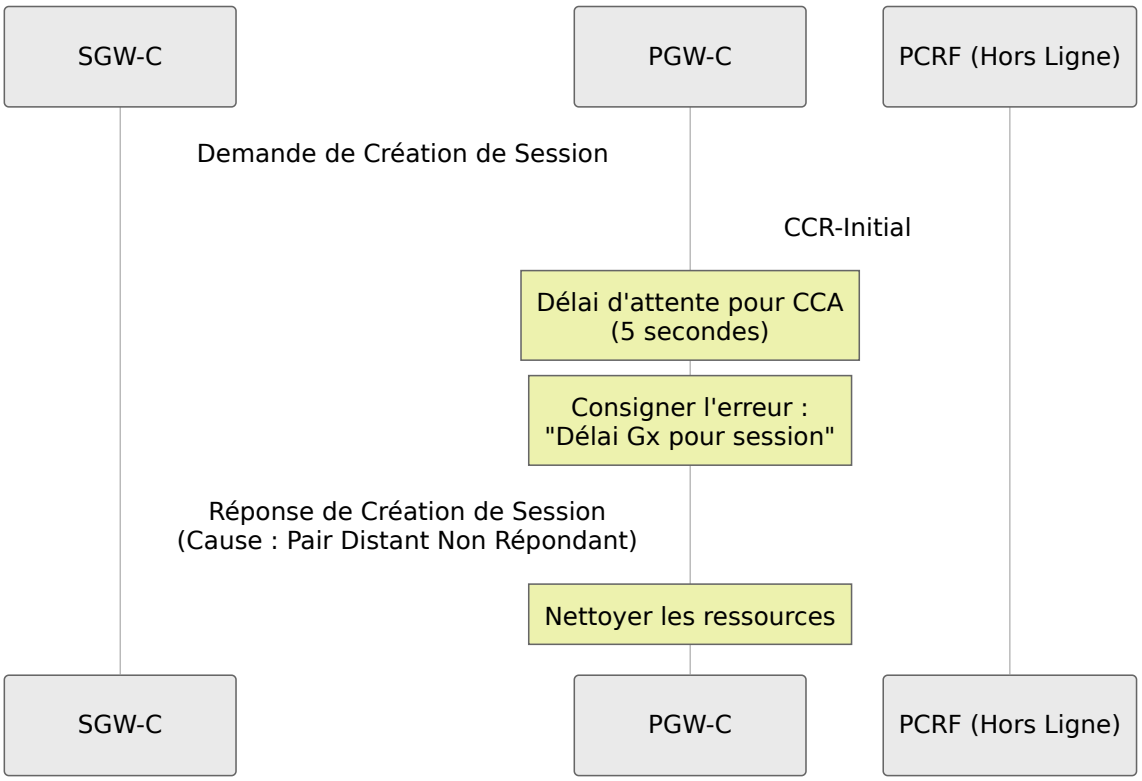
Scénarios d'Échec

Scénario 1 : La PCRF Rejette la Session (Utilisateur Inconnu)



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima

Scénario 2 : PCRF Temporairement Indisponible



Dépannage

Problèmes Courants

1. Échec de Connexion au Pair Diameter

Symptômes :

- Journal : "Pair Diameter non connecté"
- Aucun CCR-Initial envoyé

Causes Possibles :

- PCRF non accessible
- IP PCRF incorrect dans la configuration
- Pare-feu bloquant le port Diameter (3868)
- Identités Diameter incorrectes (hôte/domaine)

Résolution :

```
# Tester la connectivité réseau
ping <pcrf_ip>

# Tester le port Diameter (TCP 3868)
telnet <pcrf_ip> 3868

# Vérifier la configuration de l'identité Diameter
# Assurez-vous que l'hôte et le domaine sont des FQDN, pas des IP
```

Vérifier la Configuration :

```
config :pgw_c,
  diameter: %{
    # Doit être un FQDN, pas une IP
    host: "pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.0.30"
      }
    ]
  }
```

2. Délais CCR-Initial

Symptômes :

- Échec de la Demande de Création de Session
- Journal : "Délai CCR-Initial"

Causes Possibles :

- PCRF surchargée
- Latence réseau
- PCRF ne répond pas à cet Session-ID

Résolution :

1. Vérifier les journaux de la PCRF pour des erreurs
2. Vérifier que la PCRF traite les demandes
3. Vérifier la latence réseau : `ping <pcrf_ip>`
4. Augmenter le délai si la latence réseau est élevée

3. Sessions Rejetées par la PCRF

Symptômes :

- CCA-Initial avec Result-Code != 2001
- Échec de la Réponse de Création de Session

Codes de Résultat Communs :

Code de Résultat	Cause Probable	Résolution
5030	IMSI non dans la base de données des abonnés	Provisionner l'abonné dans HSS/SPR
5003	Autorisation rejetée	Vérifier les permissions de l'abonné
4010	PCRF trop occupée	Réessayer ou ajouter de la capacité à la PCRF

Vérifier les Journaux :

```
# Les journaux PGW-C montrent :
[error] Erreur Diameter Gx : Result-Code 5030
(DIAMETER_USER_UNKNOWN)
[error] IMSI 310260999999999 rejeté par la PCRF
```

4. QoS Non Appliquée

Symptômes :

- Session établie mais mauvaise QoS

- Les débits ne correspondent pas aux valeurs attendues

Étapes de Débogage :

1. **Vérifier CCA-Initial :**
    - Vérifier que l'AVP `QoS-Information` est présent
    - Vérifier les valeurs `APN-Aggregate-Max-Bitrate-UL/DL`
2. **Vérifier l'Établissement de Session PFCP :**
    - Vérifier que le QER a été créé avec les bonnes valeurs de MBR
    - Vérifier les journaux PGW-U pour l'installation du QER
3. **Vérifier la Politique de la PCRF :**
    - Vérifier la configuration de la PCRF
    - Vérifier que le profil de l'abonné inclut la bonne QoS

5. Problèmes de Routage Diameter

Symptômes :

- Les messages Diameter n'atteignent pas la PCRF
- Journal : "Aucun itinéraire vers Destination-Realm"

Cause :

- Incohérence de domaine entre la configuration et les messages

Résolution :

Assurer la cohérence :

```
# Tous doivent correspondre
config :pgw_c,
  diameter: %{
    realm: "epc.mnc999.mcc999.3gppnetwork.org",  # Domaine du PGW-
C
    peer_list: [
      %{
        realm: "epc.mnc999.mcc999.3gppnetwork.org"  # Domaine de
la PCRF (généralement le même)
      }
    ]
  }
```

Dans CCR-Initial :

```
Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
```

Surveillance de la Santé de Gx

Métriques Clés :

```
# Taux de messages Gx
rate(gx_inbound_messages_total{message_type="gx_CCA"}[5m])
rate(gx_outbound_messages_total{message_type="gx_CCR"}[5m])

# Taux d'erreurs Gx
rate(gx_inbound_errors_total[5m])

# Taux de succès des réponses Gx (nouvelle métrique)
sum(rate(gx_outbound_responses_total{result_code_class="2xxx"}
[5m])) /
sum(rate(gx_outbound_responses_total[5m])) * 100

# Échecs de réponse Gx par hôte PCRF
rate(gx_outbound_responses_total{result_code_class!="2xxx"}[5m])
by (diameter_host)

# Compte des sessions Gx
session_id_registry_count

# Durée de traitement des messages Gx
histogram_quantile(0.95,
rate(gx_inbound_handling_duration_bucket[5m]))
```

Métriques de Réponse par Classe de Code de Résultat :

La métrique `gx_outbound_responses_total` fournit une visibilité détaillée sur les réponses Diameter envoyées aux pairs PCRF, catégorisées par :

- **`message_type`** : Type de message de réponse (`gx_RAA`, `gx_CCA`)
- **`result_code_class`** : Catégorie de code de résultat (`2xxx`, `3xxx`, `4xxx`, `5xxx`)
- **`diameter_host`** : Pair PCRF recevant la réponse

Codes de Résultat Communs :

- **2001** (DIAMETER_SUCCESS) - Réponse réussie
- **3001** (DIAMETER_COMMAND_UNSUPPORTED) - Erreur de protocole
- **5012** (DIAMETER_UNABLE_TO_COMPLY) - Impossible d'exécuter la demande
- **5030** (DIAMETER_USER_UNKNOWN) - Abonné non trouvé

Exemples d'Alerte :

```
# Alerte sur un taux d'erreur Gx élevé
- alert: GxErrorRateHigh
  expr: rate(gx_inbound_errors_total[5m]) > 0.1
  for: 5m
  annotations:
    summary: "Taux d'erreur Gx élevé détecté"

# Alerte sur un taux d'échec de réponse Gx élevé
- alert: GxResponseFailureRate
  expr: |

sum(rate(gx_outbound_responses_total{result_code_class!="2xxx"}
[5m])) /
    sum(rate(gx_outbound_responses_total[5m])) > 0.1
  for: 5m
  annotations:
    summary: "Taux d'échec de réponse Gx élevé"
    description: "Plus de 10 % des réponses Gx sont des échecs"

# Alerte sur des échecs spécifiques à la PCRF
- alert: GxPCRFFailures
  expr:
rate(gx_outbound_responses_total{result_code_class=~"4xxx|5xxx"}
[5m]) by (diameter_host) > 0.05
  for: 3m
  annotations:
    summary: "Échecs de réponses pour la PCRF {{
$labels.diameter_host }}"
    description: "Taux d'échec élevé pour l'hôte PCRF"

# Alerte sur le rejet de session
- alert: GxSessionRejection
  expr: rate(gx_inbound_errors_total{result_code="5030"}[5m]) >
0.01
  for: 5m
  annotations:
    summary: "PCRF rejetant des sessions (USER_UNKNOWN)"
```

Journalisation de Débogage

Activer la journalisation verbose de Diameter :

```
# config/runtime.exs
config :logger, level: :debug

# Ou à l'exécution
iex> Logger.configure(level: :debug)
```

Rechercher :

- `[debug] Envoi CCR-Initial pour Session-ID : ...`
- `[debug] Reçu CCA-Initial : Result-Code 2001`
- `[error] Erreur Diameter : ...`

Surveillance des Pairs Diameter

La connectivité Diameter Gx au pair PCRF est surveillée via l'**API OAM REST**. L'API est servie sur HTTPS/TLS à `https://<host>:8443` avec toutes les routes sous le préfixe `/api`. La documentation interactive de l'API (Swagger UI) est disponible à `https://<host>:8443/api/docs`.

Points de Terminaison Diameter Peer

```
GET /api/diameter                    # tous les pairs configurés
+ état en direct
GET /api/diameter?status=disconnected # filtrer par état de
connexion
GET /api/diameter/<origin-host>      # un pair par Origin-Host
```

Exemple :

```
curl -k https://localhost:8443/api/diameter
```

`GET /api/diameter` renvoie chaque pair Diameter configuré et son état de connexion actuel. Ajoutez `?status=connected` ou `?status=disconnected` pour filtrer par état de connexion, ou ajoutez l'Origin-Host du pair à `GET /api/diameter/<origin-host>` pour récupérer un seul pair.

Pour chaque pair, les champs suivants sont renvoyés :

Champ	Description
`host`	Identité Diameter (Origin-Host) du pair
`status`	État de connexion (par exemple, Connecté / Déconnecté / En Cours de Connexion)
`ip`	Adresse IP du pair
`port`	Port du pair
`transport`	TCP ou SCTP
`connection_initiation`	Si OmniPGW initie (sortant) ou accepte (entrant) la connexion
`realm`	Domaine Diameter rapporté par le pair dans CEA
`product_name`	Chaîne de produit du fournisseur rapportée par le pair dans CEA
`application_ids`	IDs d'application Diameter négociés dans CEA (Gx = 16777238)

Un déploiement sain montre tous les pairs PCRF configurés comme **Connectés**. La perte de tous les pairs Gx empêche l'installation de politiques et bloque la création de nouvelles sessions.

Cas d'Utilisation Opérationnels

Surveiller la Connectivité PCRF :

```
curl -k https://localhost:8443/api/diameter
```

Vérifiez que tous les pairs PCRF affichent `Connecté`, que `connection_initiation` correspond à la configuration, et que `application_ids` incluent Gx (16777238).

Dépanner les Échecs de Création de Session (Problèmes Gx) : Lorsque les sessions utilisateur échouent avec des erreurs "délai PCRF", interrogez le pair affecté :

```
curl -k https://localhost:8443/api/diameter/<origin-host>
```

- `status` est `Déconnecté` → vérifiez la connectivité réseau, vérifiez que la PCRF fonctionne, et confirmez que les règles de pare-feu autorisent TCP 3868.
- `status` est `Connecté` mais les sessions échouent toujours → le problème est au niveau de l'application (vérifiez les journaux) ; la PCRF peut rejeter les abonnés.

Vérifier la Configuration Diameter : Après avoir configuré un nouveau pair PCRF, confirmez qu'il apparaît et atteint `Connecté` :

```
curl -k https://localhost:8443/api/diameter/<origin-host>
```

Vérifiez que `realm` correspond à la configuration, que `application_ids` incluent Gx, et que `product_name` affiche l'identifiant PCRF attendu.

Surveiller le Basculement : Lorsqu'un PCRF principal échoue, `GET /api/diameter?status=disconnected` le liste comme `Déconnecté` tandis que le PCRF de secours reste `Connecté`. De nouvelles sessions utilisent automatiquement le secours ; lorsque le principal se rétablit, son `status` revient à `Connecté`.

Détecter les Problèmes de Routage Diameter :

- Le pair affiche `Connecté` mais `realm` est incorrect
- `application_ids` n'inclut pas Gx (16777238)
- `product_name` ne correspond pas à la PCRF attendue

Identifier les Incohérences de Configuration : Si `connection_initiation` rapporte que le pair initie alors que la configuration définit `initiate_connection: true`, les deux extrémités tentent d'initier, ce qui peut provoquer des conditions de course de connexion.

Intégration avec les Métriques

L'API OAM fournit l'état des pairs en temps réel ; combinez-la avec Prometheus pour :

- Taux d'erreurs Gx historiques
- Comptes de messages CCR/CCA
- Tendances de latence

`GET /api/diameter` répond à "Est-ce que ça fonctionne en ce moment ?" ; les métriques répondent à "Comment ça a fonctionné dans le temps ?"

Documentation Connexe

Configuration et Politique

- **Guide de Configuration** - Configuration Diameter, configuration des pairs PCRF
- **Interface PFCP** - Application de QoS via QERs à partir des règles PCC
- **Gestion des Sessions** - Cycle de vie de la session avec intégration de la politique
- **Gestion de la QoS & des Porteurs** - Configuration détaillée de la QoS et configuration des porteurs

Intégration de la Facturation

- **Interface Diameter Gy** - Facturation en ligne déclenchée par des règles PCC
- **Format CDR de Données** - Enregistrements de facturation hors ligne avec informations de politique
- **Configuration PCO** - Livraison P-CSCF pour le contrôle de politique IMS

Opérations

- **Guide de Surveillance** - Métriques Gx, suivi des politiques, alertes de connectivité PCRF
- **Interface S5/S8** - Intégration de la gestion des porteurs avec la politique

---

Retour au Guide des Opérations

Chargement en Ligne par Diamètre (Interface Gy/Ro)

Interface du Système de Chargement en Ligne (OCS)

Table des Matières

1. Aperçu
2. Architecture de Chargement 3GPP
3. Bases de l'Interface Gy/Ro
4. Messages de Contrôle de Crédit
5. Flux de Chargement en Ligne
6. Contrôle de Chargement de Support
7. Contrôle de Crédit pour Services Multiples
8. Configuration
9. Flux de Messages
10. Gestion des Erreurs
11. Intégration avec Gx
12. Dépannage

Aperçu

L'**interface Gy** (également appelée **interface Ro** dans les contextes IMS) connecte PGW-C au **Système de Chargement en Ligne (OCS)** pour le contrôle de crédit en temps réel. Cela permet :

- **Chargement Prépayé** - Autorisation et déduction de crédit en temps réel

- **Contrôle de Crédit en Temps Réel** - Accorder un quota avant la livraison du service
- **Chargement Basé sur le Service** - Chargement différent pour la voix, les données, les SMS, etc.
- **Mises à Jour de Compte Immédiates** - Mises à jour du solde de crédit en temps réel
- **Refus de Service** - Bloquer le service lorsque le crédit est épuisé

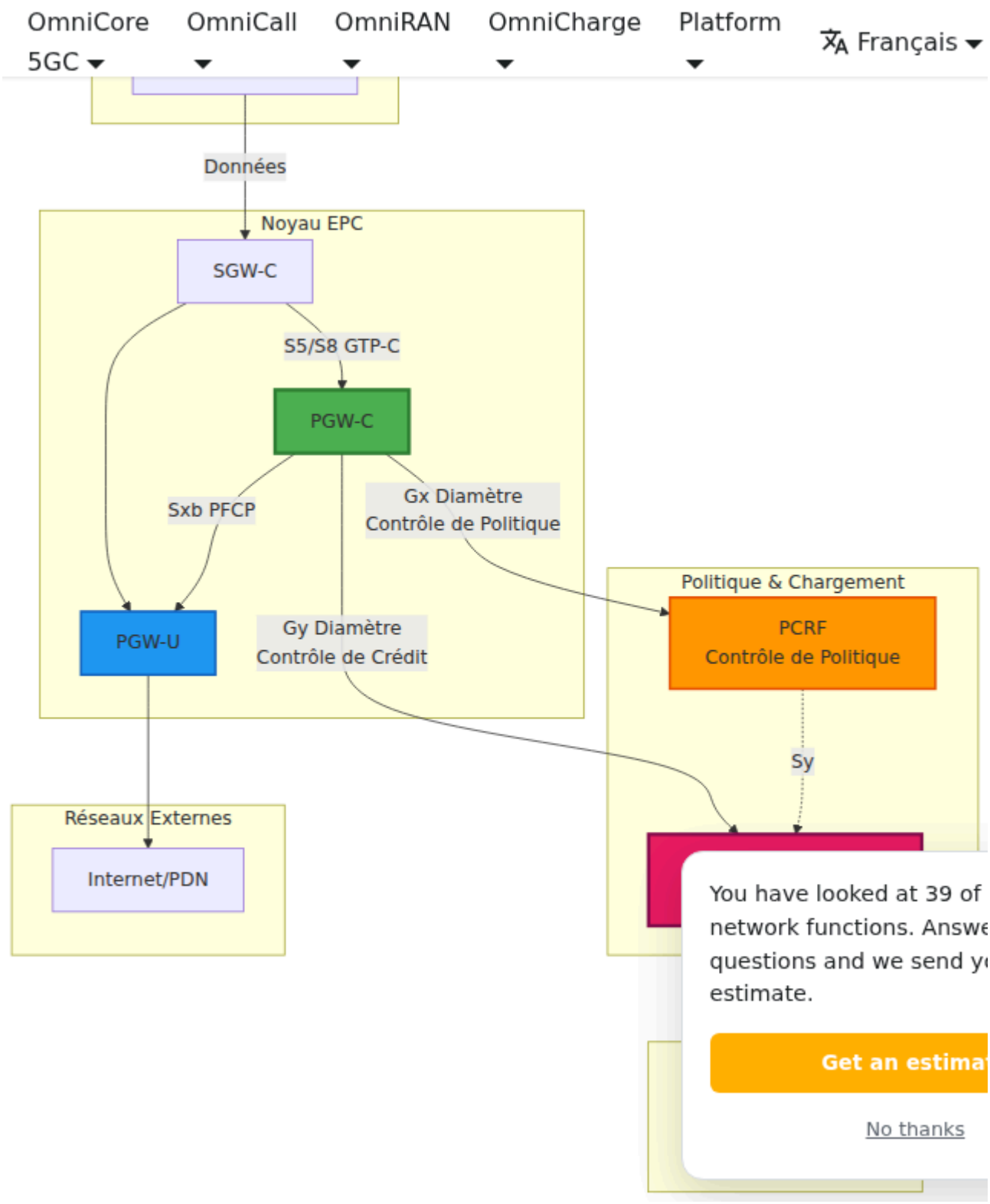
Chargement en Ligne vs. Chargement Hors Ligne

Aspect	Chargement en Ligne (Gy/Ro)	Chargement Hors Ligne (Gz/Rf)
Temps	En temps réel, avant le service	Après la livraison du service
Cas d'utilisation	Abonnés prépayés	Abonnés postpayés
Vérification de Crédit	Oui, avant d'accorder le service	Non, facture générée plus tard
Système	OCS (Système de Chargement en Ligne)	CGF/CDF (Fonction de Données de Chargement)
Risque	Pas de perte de revenus	Risque de factures impayées
Complexité	Élevée (exigences en temps réel)	Inférieure (traitement par lots)
Impact sur l'utilisateur	Service refusé si pas de crédit	Service toujours disponible

Voir aussi : Format CDR de Données pour les enregistrements de chargement hors ligne (facturation postpayée)

Voir aussi : Gestion des Sessions pour le cycle de vie complet de la session PDN, y compris l'intégration de chargement

Gy dans l'Architecture Réseau



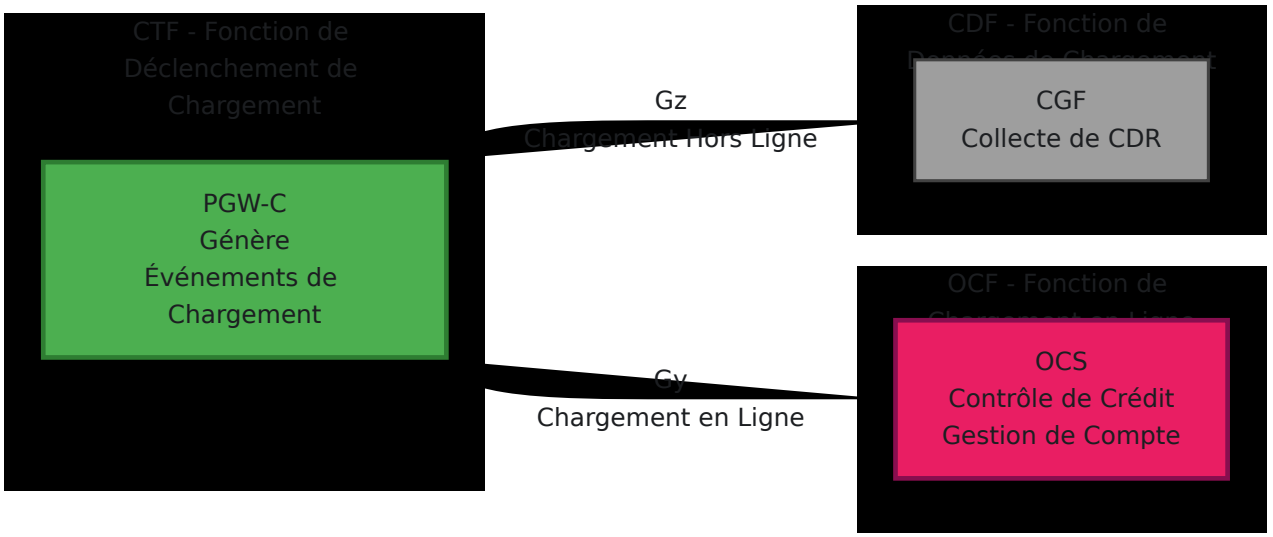
Fonctions Clés

Fonction	Description
Autorisation de Crédit	Demander un quota à l'OCS avant de permettre le trafic
Gestion de Quota	Suivre les unités accordées (octets, temps, événements)
Détection de D��plétion de Crédit	Surveiller le quota restant
Ré-autorisation	Demander un quota supplémentaire lorsque le seuil est atteint
Résiliation de Service	Arrêter le service lorsque le crédit est épuisé
Règlement Final	Rapporter l'utilisation réelle à la fin de la session

---

Architecture de Chargement 3GPP

Points de Référence de Chargement



Fonction de Déclenchement de Chargement (CTF)

PGW-C agit comme un **CTF (Fonction de Déclenchement de Chargement)**, responsable de :

1. **Détecter les événements facturables** - Début de session, utilisation des données, fin de session
2. **Demander une autorisation de crédit** - Avant de permettre le service
3. **Suivre la consommation de quota** - Surveiller les unités accordées
4. **Générer des événements de chargement** - Déclencher des demandes de crédit
5. **Appliquer le contrôle de crédit** - Bloquer le trafic lorsque le quota est épuisé

Fonction de Chargement en Ligne (OCF)

L'OCS implémente la **OCF (Fonction de Chargement en Ligne)** :

1. **Gestion du solde de compte** - Suivre le crédit de l'abonné

2. **Tarification** - Déterminer le prix par unité (par Mo, par seconde, etc.)
3. **Réservation de Crédit** - Réserver du crédit pour le quota accordé
4. **Déduction de Crédit** - Déduire lors du rapport d'utilisation
5. **Décisions de Politique** - Accorder ou refuser en fonction du solde

---

Bases de l'Interface Gy/Ro

Référence 3GPP

- **Spécification :** 3GPP TS 32.299 (Architecture de chargement)
- **Protocole :** 3GPP TS 32.251 (Chargement du domaine PS)
- **ID d'Application Diamètre :** 4 (Gy/Ro - Application de Contrôle de Crédit)
- **Protocole de Base :** RFC 4006 (Application de Contrôle de Crédit Diamètre)

Concept de Session

Chaque connexion PDN UE nécessitant un chargement en ligne a une **session Gy/Ro** identifiée par un **Session-ID**. Cette session :

- Créée lorsque le support nécessite un chargement en ligne (CCR-Initial)
- Mise à jour lorsque le quota est consommé (CCR-Update)
- Terminée lorsque la session se termine (CCR-Termination)

Format de l'ID de Session

```
Session-ID: <Origin-Host>;<high32>;<low32>[;<optional>]
Exemple : omni-
pgw_c.epc.mnc999.mcc999.3gppnetwork.org;9876543210;12345;gy
```

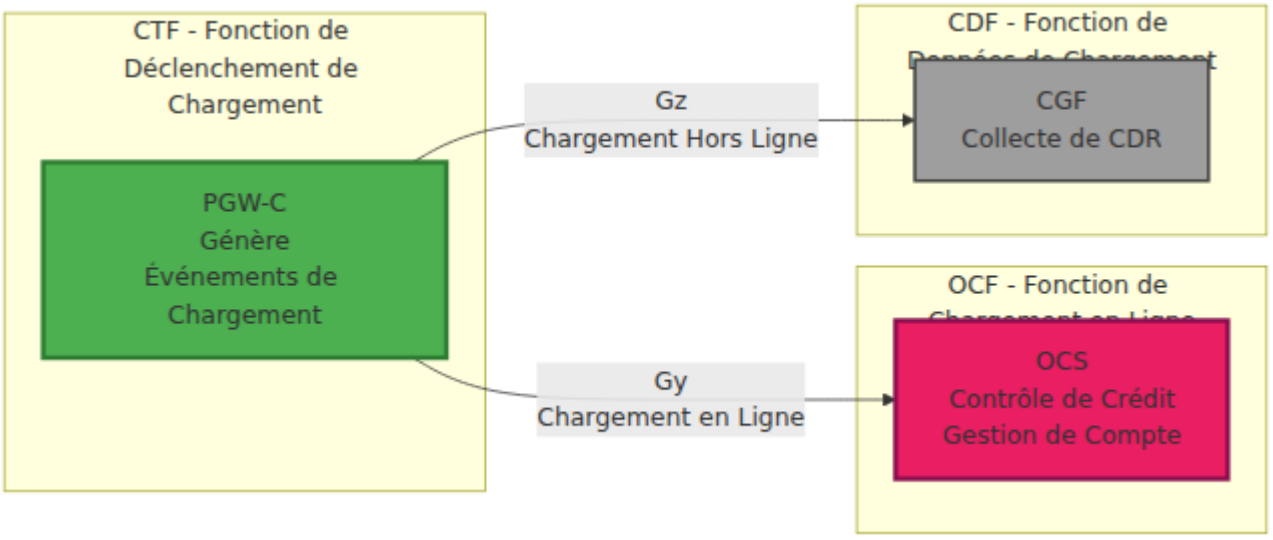
Composants :

- **Origin-Host :** Identité Diamètre de PGW-C

- **high32 :** 32 bits supérieurs de l'identifiant unique
- **low32 :** 32 bits inférieurs de l'identifiant unique
- **optional :** Identifiant supplémentaire (par exemple, "gy" pour distinguer de Gx)

Messages de Contrôle de Crédit

Types de Messages



CCR-Initial (Demande de Contrôle de Crédit - Initial)

Quand : UE crée une connexion PDN et le support nécessite un chargement en ligne

But :

- Demander une autorisation de crédit initiale à l'OCS
- Réserver un quota pour la livraison du service
- Établir une session Gy/Ro

AVPs Clés Envoyés par PGW-C :

Nom de l'AVP	Code de l'AVP	Type	Description
Session-Id	263	UTF8String	Identifiant unique de session Gy
Auth-Application-Id	258	Unsigned32	4 (Contrôle de Crédit)
Origin-Host	264	DiamIdent	Identité Diamètre de PGW-C
Origin-Realm	296	DiamIdent	Domaine Diamètre de PGW-C
Destination-Realm	283	DiamIdent	Domaine de l'OCS
CC-Request-Type	416	Enumerated	1 = INITIAL_REQUEST
CC-Request-Number	415	Unsigned32	Numéro de séquence (commence à 0)
Subscription-Id	443	Grouped	Identifiant UE (IMSI/MSISDN)
Service-Context-Id	461	UTF8String	Identifiant de contexte de chargement
Multiple-Services-Credit-Control	456	Grouped	Demandes de crédit spécifiques au service
Requested-Service-Unit	437	Grouped	Quota demandé (octets, temps, etc.)
Used-Service-Unit	446	Grouped	Quota utilisé (0 pour initial)

Nom de l'AVP	Code de l'AVP	Type	Description
Service-Identifier	439	Unsigned32	Identifiant de type de service
Rating-Group	432	Unsigned32	Identifiant de catégorie de chargement

Exemple de Structure CCR-I :

```
CCR (Code de Commande : 272, Demande)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Auth-Application-Id: 4
├── Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├── Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── CC-Request-Type: INITIAL_REQUEST (1)
├── CC-Request-Number: 0
├── Subscription-Id (Grouped)
│   ├── Subscription-Id-Type: END_USER_IMSI (1)
│   └── Subscription-Id-Data: "310260123456789"
├── Subscription-Id (Grouped)
│   ├── Subscription-Id-Type: END_USER_E164 (0)
│   └── Subscription-Id-Data: "15551234567"
├── Service-Context-Id: "32251@3gpp.org"
├── Multiple-Services-Credit-Control (Grouped)
│   ├── Service-Identifier: 1
│   ├── Rating-Group: 100
│   └── Requested-Service-Unit (Grouped)
│       └── CC-Total-Octets: 10000000 (demande 10 Mo)
└── Used-Service-Unit (Grouped)
    └── CC-Total-Octets: 0 (aucune utilisation pour l'instant)
```

CCA-Initial (Réponse de Contrôle de Crédit - Initial)

Envoyé par : OCS en réponse à CCR-I

But :

- Accorder ou refuser l'autorisation de crédit
- Fournir un quota pour la livraison du service
- Spécifier les paramètres de tarification et de chargement

AVPs Clés Reçus par PGW-C :

Nom de l'AVP	Code de l'AVP	Description
Result-Code	268	Succès (2001) ou code d'erreur
Multiple-Services-Credit-Control	456	Accords de crédit spécifiques au service
Granted-Service-Unit	431	Quota accordé (octets, temps, etc.)
Validity-Time	448	Période de validité du quota (secondes)
Result-Code	268	Code de résultat par service
Final-Unit-Indication	430	Action lorsque le quota est épuisé
Volume-Quota-Threshold	-	Seuil pour ré-autorisation

Exemple de Réponse de Succès :

```
CCA (Code de Commande : 272, Réponse)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Result-Code: DIAMETER_SUCCESS (2001)
├── Origin-Host: "ocs.example.com"
├── Origin-Realm: "example.com"
├── Auth-Application-Id: 4
├── CC-Request-Type: INITIAL_REQUEST (1)
├── CC-Request-Number: 0
└── Multiple-Services-Credit-Control (Grouped)
    ├── Result-Code: DIAMETER_SUCCESS (2001)
    ├── Service-Identifier: 1
    ├── Rating-Group: 100
    ├── Granted-Service-Unit (Grouped)
    │   └── CC-Total-Octets: 10000000 (accordé 10 Mo)
    ├── Validity-Time: 3600 (quota valide pendant 1 heure)
    └── Volume-Quota-Threshold: 8000000 (ré-auth à 8 Mo utilisés,
80%)
```

CCR-Update (Demande de Contrôle de Crédit - Mise à Jour)

Quand :

- Seuil de quota accordé atteint (par exemple, 80 % consommé)
- Temps de validité expiré
- Changement de service nécessitant une ré-autorisation
- Changement de temps de tarification

But :

- Demander un quota supplémentaire
- Rapporter l'utilisation du quota précédemment accordé
- Mettre à jour les paramètres de chargement

Différences Clés par Rapport à CCR-I :

- `CC-Request-Type: UPDATE_REQUEST (2)`
- `CC-Request-Number` incrémenté

- `Used-Service-Unit` contient l'utilisation réelle
- `Requested-Service-Unit` pour plus de quota

Exemple de Structure CCR-U :

```
CCR (Code de Commande : 272, Demande)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Auth-Application-Id: 4
├── Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├── Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── CC-Request-Type: UPDATE_REQUEST (2)
├── CC-Request-Number: 1
└── Multiple-Services-Credit-Control (Grouped)
    ├── Service-Identifier: 1
    ├── Rating-Group: 100
    ├── Used-Service-Unit (Grouped)
    │   └── CC-Total-Octets: 8000000 (8 Mo utilisés jusqu'à
présent)
    └── Requested-Service-Unit (Grouped)
        └── CC-Total-Octets: 10000000 (demande encore 10 Mo)
```

CCA-Update (Réponse de Contrôle de Crédit - Mise à Jour)

Envoyé par : OCS en réponse à CCR-U

But :

- Accorder un quota supplémentaire (si crédit disponible)
- Accuser réception de l'utilisation
- Mettre à jour les paramètres de chargement

Résultats Possibles :

1. Plus de Quota Accordé :

```
CCA (Mise à Jour)
└── Multiple-Services-Credit-Control
    ├── Result-Code: DIAMETER_SUCCESS (2001)
    ├── Granted-Service-Unit
    │   └── CC-Total-Octets: 10000000 (encore 10 Mo)
    └── Validity-Time: 3600
```

2. Quota Final (Crédit Épuisé) :

```
CCA (Mise à Jour)
└── Multiple-Services-Credit-Control
    ├── Result-Code: DIAMETER_SUCCESS (2001)
    ├── Granted-Service-Unit
    │   └── CC-Total-Octets: 1000000 (il reste seulement 1 Mo)
    └── Final-Unit-Indication
        └── Final-Unit-Action: TERMINATE (0)
```

3. Pas de Crédit Disponible :

```
CCA (Mise à Jour)
├── Result-Code: DIAMETER_CREDIT_LIMIT_REACHED (4012)
└── Multiple-Services-Credit-Control
    ├── Result-Code: DIAMETER_CREDIT_LIMIT_REACHED (4012)
    └── Final-Unit-Indication
        └── Final-Unit-Action: TERMINATE (0)
```

CCR-Termination (Demande de Contrôle de Crédit - Résiliation)

Quand :

- UE se détache
- Connexion PDN supprimée
- Session terminée pour une raison quelconque

But :

- Rapport final d'utilisation
- Fermer la session Gy/Ro
- Règlement final

Différences Clés :

- `CC-Request-Type: TERMINATION_REQUEST (3)`
- `Used-Service-Unit` contient l'utilisation finale
- Pas de `Requested-Service-Unit` (plus de quota nécessaire)
- Inclut `Termination-Cause`

Exemple de Structure CCR-T :

```
CCR (Code de Commande : 272, Demande)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Auth-Application-Id: 4
├── Origin-Host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org"
├── Origin-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── Destination-Realm: "epc.mnc999.mcc999.3gppnetwork.org"
├── CC-Request-Type: TERMINATION_REQUEST (3)
├── CC-Request-Number: 5
├── Termination-Cause: DIAMETER_LOGOUT (1)
└── Multiple-Services-Credit-Control (Grouped)
    ├── Service-Identifier: 1
    ├── Rating-Group: 100
    └── Used-Service-Unit (Grouped)
        └── CC-Total-Octets: 18500000 (18,5 Mo d'utilisation
totale)
```

CCA-Termination (Réponse de Contrôle de Crédit - Résiliation)

Envoyé par : OCS en réponse à CCR-T

But :

- Accuser réception de la résiliation de session
- Compléter la comptabilité

- Libérer le crédit réservé

Exemple CCA-T :

```
CCA (Code de Commande : 272, Réponse)
├── Session-Id: "pgw_c.example.com;123;456;gy"
├── Result-Code: DIAMETER_SUCCESS (2001)
├── Origin-Host: "ocs.example.com"
├── Origin-Realm: "example.com"
├── Auth-Application-Id: 4
├── CC-Request-Type: TERMINATION_REQUEST (3)
└── CC-Request-Number: 5
```

Flux de Chargement en Ligne

Types d'Unités de Service

L'OCS peut accorder un quota dans différentes unités :

Type d'Unité	AVP	Description	Cas d'utilisation
Temps	CC-Time	Secondes	Appels vocaux, durée de session
Volume	CC-Total-Octets	Octets (total montant + montant)	Services de données
Volume (séparé)	CC-Input-Octets, CC-Output-Octets	Octets (séparés)	Chargement asymétrique
Spécifique au Service	CC-Service-Specific-Units	Unités personnalisées	SMS, MMS, appels API
Événements	-	Événements comptés	Services payants à l'utilisation

Gestion des Seuils de Quota

Problème : Comment PGW-C sait-il quand demander plus de quota ?

Solution : L'OCS fournit un **Volume-Quota-Threshold** ou **Time-Quota-Threshold**. PGW-C surveille l'utilisation via les Rapports de Session PFCP de PGW-U (voir Interface PFCP).

Exemple de Flux :

```
1. L'OCS accorde un quota de 10 Mo avec un seuil de 80 % (8 Mo)
2. PGW-C surveille l'utilisation via les rapports d'utilisation de
PGW-U (Rapports de Session PFCP)
3. Lorsque l'utilisation atteint 8 Mo :
   → PGW-C envoie CCR-Update
   → Continue à permettre le trafic (ne pas attendre la réponse)
4. L'OCS répond avec plus de quota
5. Si le quota est épuisé avant l'envoi de CCR-Update :
   → PGW-C doit bloquer le trafic
```

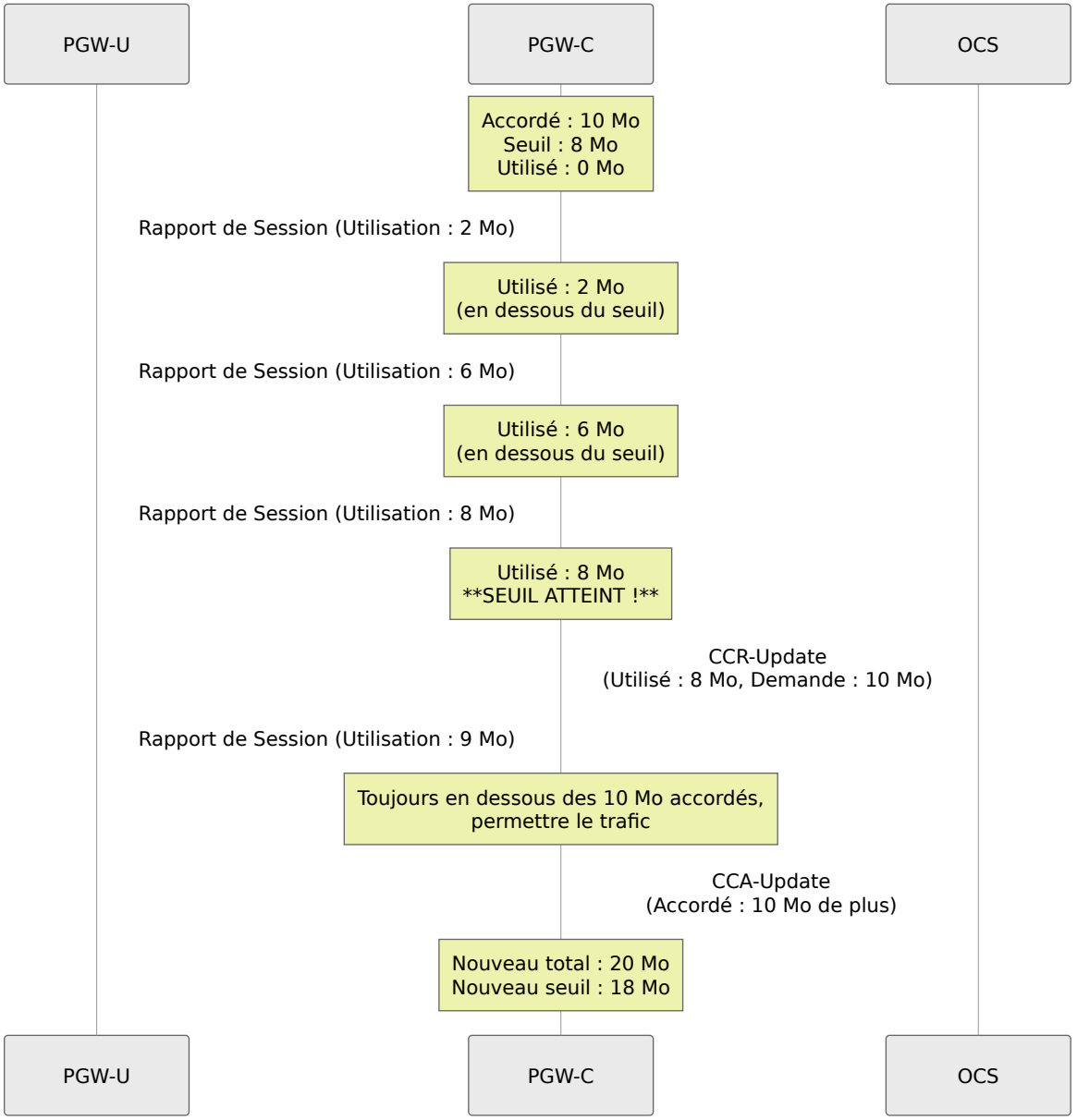
Calcul du Seuil :

```
Granted-Service-Unit: 10000000 octets (10 Mo)
Volume-Quota-Threshold: 8000000 octets (8 Mo)

Lorsque 8 Mo consommés → Déclencher CCR-Update
Tampon restant : 2 Mo (permet du temps pour la réponse de l'OCS)
```

Surveillance de PGW-C :

PGW-C surveille l'utilisation via **Rapports de Session PFCP** de PGW-U :



Indication de l'Unité Finale

Que se passe-t-il lorsque le crédit est épuisé ?

L'OCS inclut l'AVP **Final-Unit-Indication** dans CCA pour spécifier l'action :

Final-Unit-Action	Valeur	Comportement de PGW-C
TERMINATE	0	Bloquer tout le trafic, initier la résiliation de session
REDIRECT	1	Rediriger le trafic vers un portail (par exemple, page de recharge)
RESTRICT_ACCESS	2	Autoriser l'accès uniquement à des services spécifiques (par exemple, serveur de recharge)

Exemple : Unité Finale avec Redirection

```
CCA (Mise à Jour)
└── Multiple-Services-Credit-Control
    ├── Result-Code: DIAMETER_SUCCESS (2001)
    ├── Granted-Service-Unit
    │   └── CC-Total-Octets: 1000000 (dernier 1 Mo)
    └── Final-Unit-Indication
        ├── Final-Unit-Action: REDIRECT (1)
        └── Redirect-Server (Grouped)
            ├── Redirect-Address-Type: URL (2)
            └── Redirect-Server-Address:
"http://topup.example.com"
```

Actions de PGW-C :

1. **TERMINATE :** Envoyer CCR-T, supprimer le support
2. **REDIRECT :** Installer une règle PFCP pour rediriger HTTP vers l'URL de recharge
3. **RESTRICT_ACCESS :** Installer des règles PFCP permettant uniquement les IP sur liste blanche

---

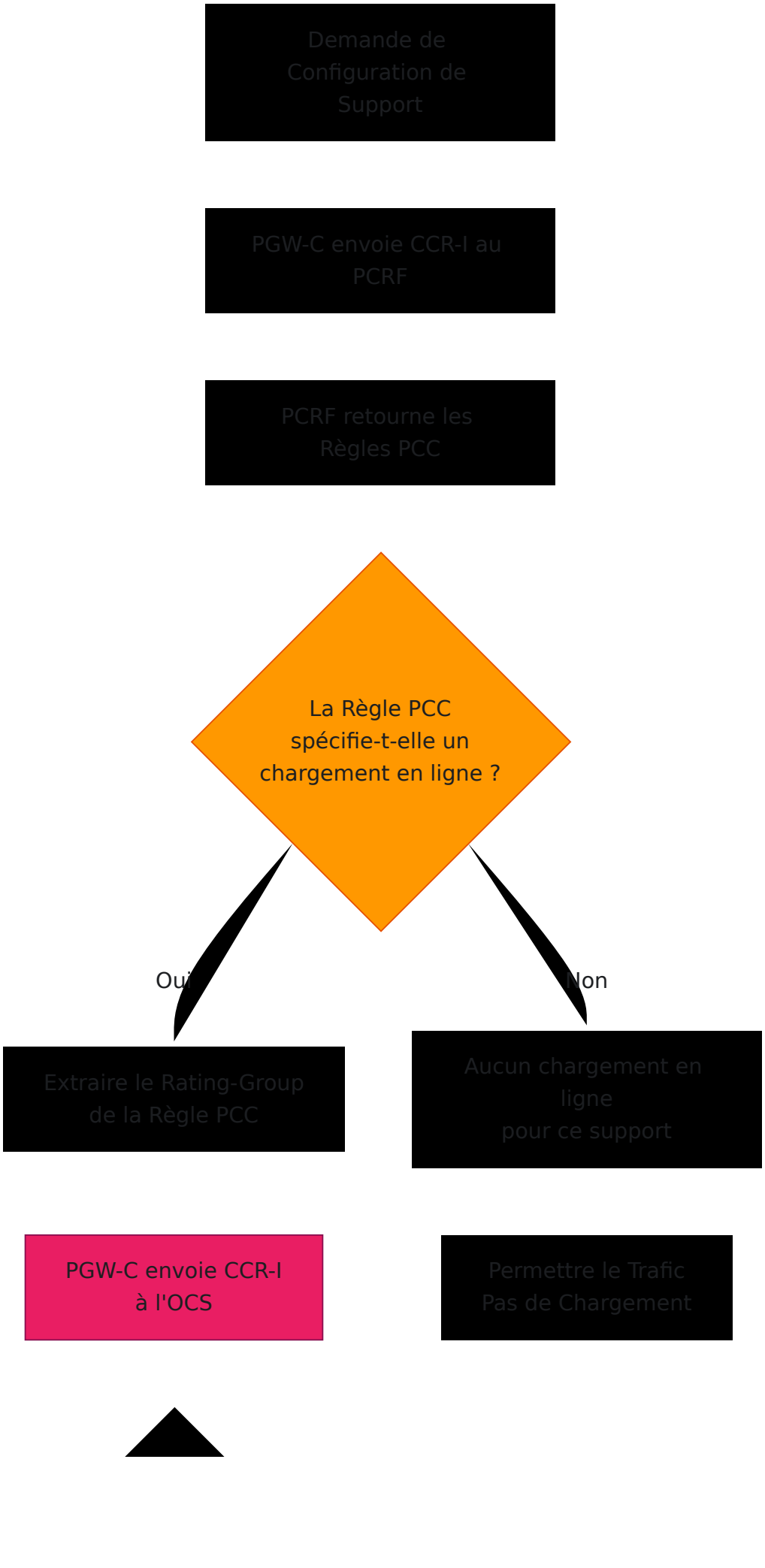
Contrôle de Chargement de Support

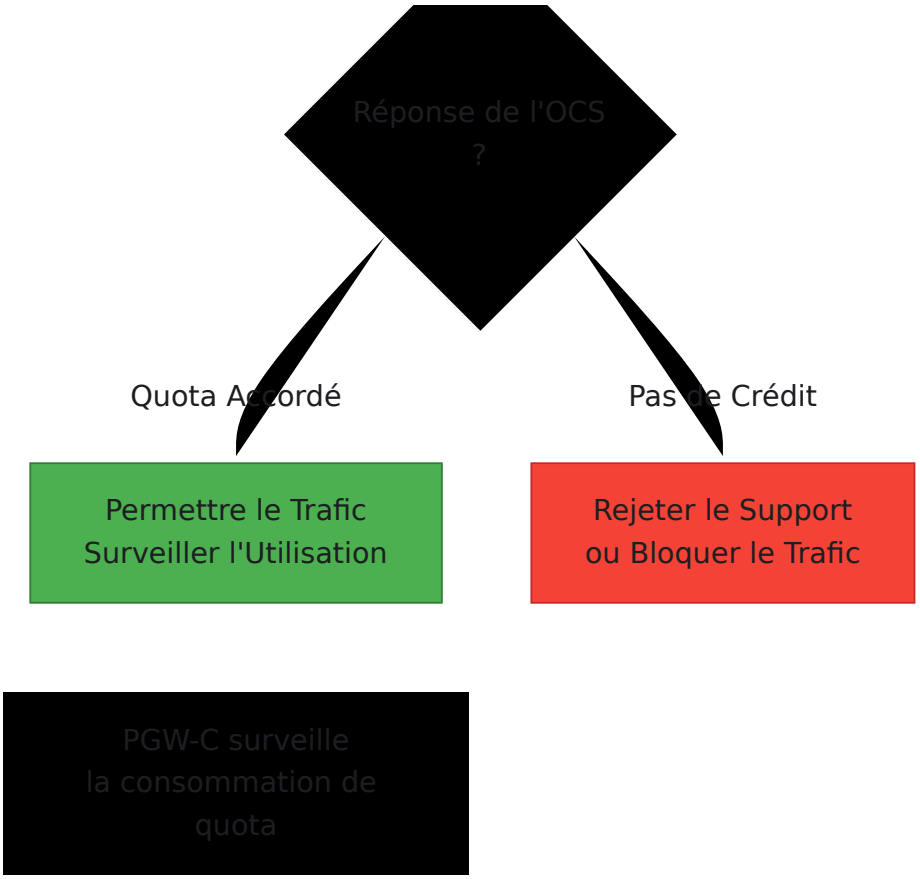
Qu'est-ce qui Contrôle si un Support est Chargé ?

Spécification 3GPP : TS 23.203, TS 29.212, TS 32.251

Le chargement de support est contrôlé par des **Règles PCC** fournies par le PCRF via l'interface Gx. Voir Interface Gx Diamètre pour la documentation complète des règles PCC.

Flux de Décision de Chargement :


Demande de Configuration de Support
PGW-C envoie CCR-I au PCRF
PCRF retourne les Règles PCC
La Règle PCC spécifie-t-elle un chargement en ligne ?
Oui
Non
Extraire le Rating-Group de la Règle PCC
Aucun chargement en ligne pour ce support
PGW-C envoie CCR-I à l'OCS
Permettre le Trafic Pas de Chargement



Règle PCC avec Informations de Chargement

Réponse PCRF (CCA-I sur Gx) :

```
CCA (Interface Gx)
└── Charging-Rule-Definition (Grouped)
    ├── Charging-Rule-Name: "prepaid_data_rule"
    ├── Rating-Group: 100
    ├── Online: 1 (activer le chargement en ligne)
    ├── Offline: 0 (désactiver le chargement hors ligne)
    ├── Metering-Method: VOLUME (1)
    ├── Precedence: 100
    ├── Flow-Information: [...]
    └── QoS-Information: [...]
```

AVPs de Chargement Clés dans les Règles PCC :

Nom de l'AVP	Code de l'AVP	Valeurs	Description
Rating-Group	432	Unsigned32	Catégorie de chargement (correspond à la tarification dans l'OCS)
Online	1009	0=Désactiver, 1=Activer	Activer le chargement en ligne (Gy)
Offline	1008	0=Désactiver, 1=Activer	Activer le chargement hors ligne (Gz)
Metering-Method	1007	0=Durée, 1=Volume, 2=Les Deux	Ce qu'il faut mesurer
Reporting-Level	1011	0=Service, 1=Groupe de Tarification	Granularité des rapports d'utilisation

Matrice de Décision de Chargement de Support

En Ligne	Hors Ligne	Rating-Group	Comportement
1	0	Présent	Chargement en ligne uniquement (prépayé)
0	1	Présent	Chargement hors ligne uniquement (postpayé)
1	1	Présent	Chargement en ligne et hors ligne (convergent)
0	0	-	Pas de chargement (service gratuit)

```
### Groupes de Tarification Multiples

Une seule connexion PDN peut avoir **plusieurs porteurs avec
différents groupes de tarification** :

**Scénario Exemple :**
```

Porteur par Défaut (Internet) ├── Groupe-de-Tarification : 100 (Données Standard) └── En Ligne : 1

Porteur Dédié 1 (Streaming Vidéo) ├── Groupe-de-Tarification : 200 (Service Vidéo) └── En Ligne : 1

Porteur Dédié 2 (Voix IMS) ├── Groupe-de-Tarification : 300 (Voix) └── En Ligne : 1

```
**Comportement PGW-C Gy :**

- **CCR-I unique** avec plusieurs sections MSCC (Contrôle de
Crédit de Services Multiples) :
```

CCR-Initial ├── Session-Id : "..." └── Contrôle de Crédit de Services Multiples ├── [Groupe-de-Tarification : 100] → Données Standard ├── [Groupe-de-Tarification : 200] → Service Vidéo └── [Groupe-de-Tarification : 300] → Voix

```
**Réponse OCS :**
```

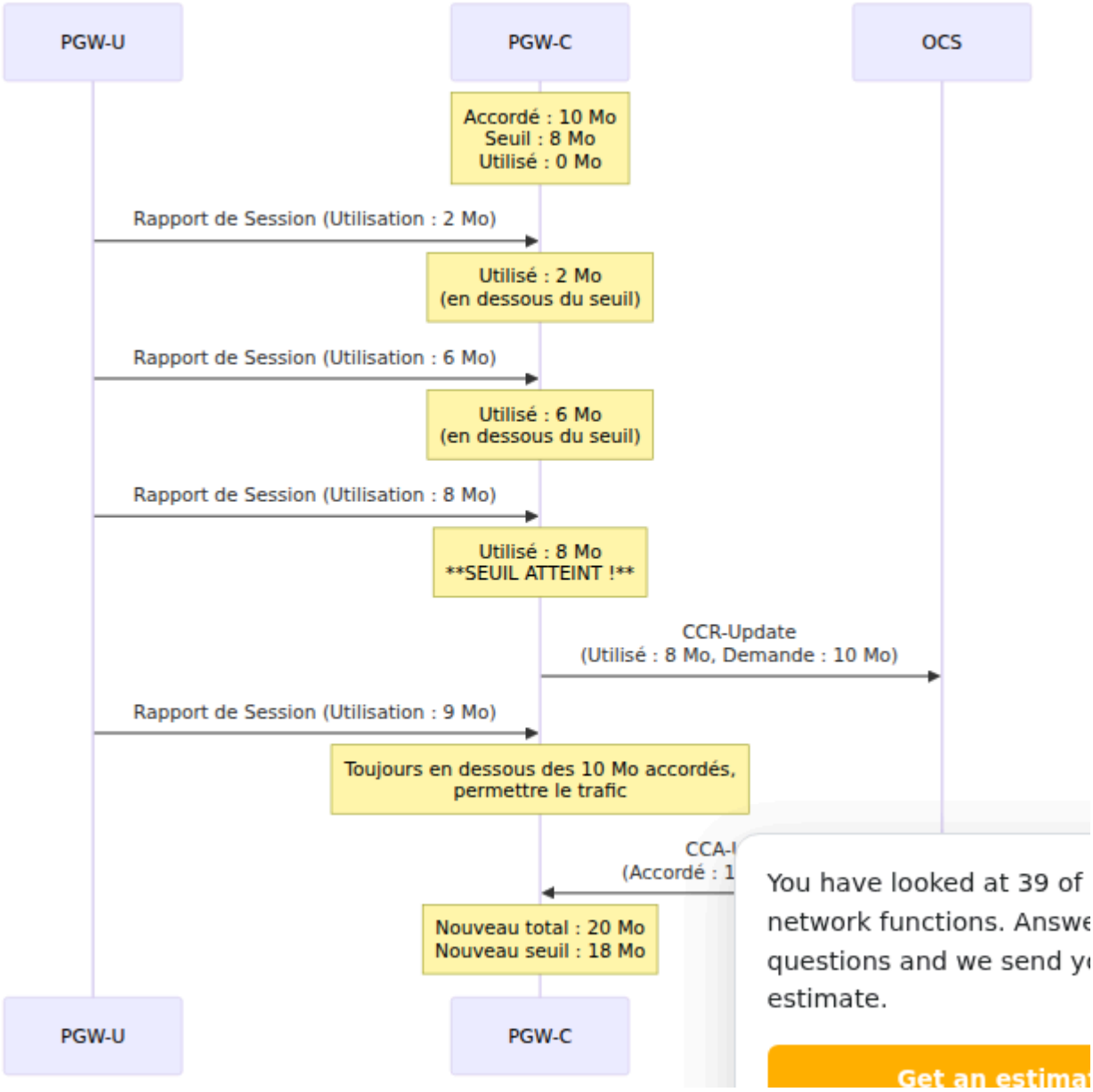
CCA-Initial └── Contrôle de Crédit de Services Multiples ├── [Groupe-de-Tarification : 100] → Accordé : 10 Mo ├── [Groupe-de-Tarification : 200] → Accordé : 5 Mo (vidéo plus coûteux) └── [Groupe-de-Tarification : 300] → Accordé : 60 secondes

```
### Application de la Tarification par Service

**PGW-C suit le quota par Groupe-de-Tarification :**

```elixir
# Pseudocode
state.charging_quotas = %{
  100 => %{granted: 10_000_000, used: 0, threshold:
8_000_000},
  200 => %{granted: 5_000_000, used: 0, threshold:
4_000_000},
  300 => %{granted: 60_000, used: 0, threshold: 48_000}
# millisecondes
}
```

Surveillance de l'Utilisation par Porteur :
```



Contrôle de Crédit de Services Multiples

AVP MSCC (Contrôle de Crédit de Services Multiples)

Objectif : Regrouper les informations de tarification pour un service/groupe de tarification spécifique

Structure :

```
Contrôle de Crédit de Services Multiples (Groupé, AVP 456)
├── Identifiant-Service (Unsigned32, AVP 439)
├── Groupe-de-Tarification (Unsigned32, AVP 432)
├── Unité-Service-Demandée (Groupée, AVP 437)
│   ├── CC-Temps (Unsigned32, AVP 420)
│   ├── CC-Octets-Total (Unsigned64, AVP 421)
│   ├── CC-Octets-Entrants (Unsigned64, AVP 412)
│   └── CC-Octets-Sortants (Unsigned64, AVP 414)
├── Unité-Service-Utilisée (Groupée, AVP 446)
│   └── [Même structure que l'Unité-Service-Demandée]
├── Unité-Service-Accordée (Groupée, AVP 431)
│   └── [Même structure que l'Unité-Service-Demandée]
├── Temps-de-Validité (Unsigned32, AVP 448)
├── Code-Résultat (Unsigned32, AVP 268)
└── Indication-Unit-Finale (Groupée, AVP 430)
    └── Action-Unit-Finale (Enuméré, AVP 449)
```

Identifiant-Service vs. Groupe-de-Tarification

Attribut	Identifiant-Service	Groupe-de-Tarification
Objectif	Identifie le type de service	Identifie la catégorie de tarification
Exemple	1=Données, 2=Voix, 3=SMS	100=Régulier, 200=Premium
Granularité	Classification large	Tarif spécifique
Requis	Optionnel	Requis pour la tarification
Mapping	Peut mapper à plusieurs RG	Tarif unique dans OCS

Exemple :

```
Identifiant-Service : 1 (Service de Données)
├─── Groupe-de-Tarification : 100 (Données Standard - 0,01 $/Mo)
└─── Groupe-de-Tarification : 200 (Données Premium - 0,05 $/Mo)

Identifiant-Service : 2 (Voix)
└─── Groupe-de-Tarification : 300 (Appels Voix - 0,10 $/min)
```

Configuration

Configuration de Base Gy

Modifiez `config/runtime.exs` :

```
config :pgw_c,
  gy: %&#123;
    # Activer ou désactiver la tarification en ligne globalement
    enabled: true,

    # Délai de connexion OCS (millisecondes)
    timeout_ms: 5000,

    # Demande de quota par défaut (octets) si non spécifiée par
PCRF
    default_requested_quota: 10_000_000,  # 10 Mo

    # Pourcentage de seuil pour la ré-autorisation
    # (0.8 = déclencher CCR-Mise à Jour à 80% de quota consommé)
    quota_threshold_percentage: 0.8,

    # Action lorsque le délai d'attente OCS se produit
    # Options : :block, :allow
    timeout_action: :block,

    # Action lorsque l'OCS ne retourne pas de crédit
    # Options : :terminate, :redirect
    no_credit_action: :terminate,

    # URL de redirection pour le rechargement (utilisée si
no_credit_action: :redirect)
    topup_redirect_url: "http://topup.example.com"
  &#125;,
  diameter: %&#123;
    listen_ip: "0.0.0.0",
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",

    # Configuration du pair OCS
    peer_list: [
      # PCRF pour le contrôle de politique (Gx)
      %&#123;
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.0.30",
        initiate_connection: true
      &#125;,
      # OCS pour la tarification en ligne (Gy)
```

```
      %&#123;
        host: "ocs.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "10.0.0.40",
        initiate_connection: true
      &#125;
    ]
  &#125;
```

Explication des Paramètres de Configuration

`enabled`

- `true` : Tarification en ligne active, messages CCR envoyés à OCS
- `false` : Tarification en ligne désactivée, aucun message Gy

`timeout_ms`

- Temps d'attente pour la réponse CCA de l'OCS
- Recommandé : 3000-5000 ms

`default_requested_quota`

- Quota par défaut à demander si le PCRF ne spécifie pas
- Valeurs typiques : 1-100 Mo

`quota_threshold_percentage`

- Déclencher CCR-Mise à Jour lorsque ce % de quota est consommé
- Recommandé : 0.75-0.85 (75%-85%)
- Plus élevé = moins de messages, mais risque d'épuisement du quota
- Plus bas = plus de messages, mais plus sûr

`timeout_action`

- **`:block`** - Bloquer le trafic si l'OCS ne répond pas (plus sûr, empêche la perte de revenus)
- **`:allow`** - Autoriser le trafic si l'OCS ne répond pas (meilleure UX, risque de revenus)

`no_credit_action`

- `:terminate` - Supprimer le porteur lorsque le crédit est épuisé
- `:redirect` - Rediriger vers le portail de rechargement

Configuration Spécifique à l'Environnement

Production (abonnés prépayés) :

```
config :pgw_c,
  gy: %&#123;
    enabled: true,
    timeout_action: :block,
    no_credit_action: :terminate,
    quota_threshold_percentage: 0.8
  &#125;
```

Test/Développement :

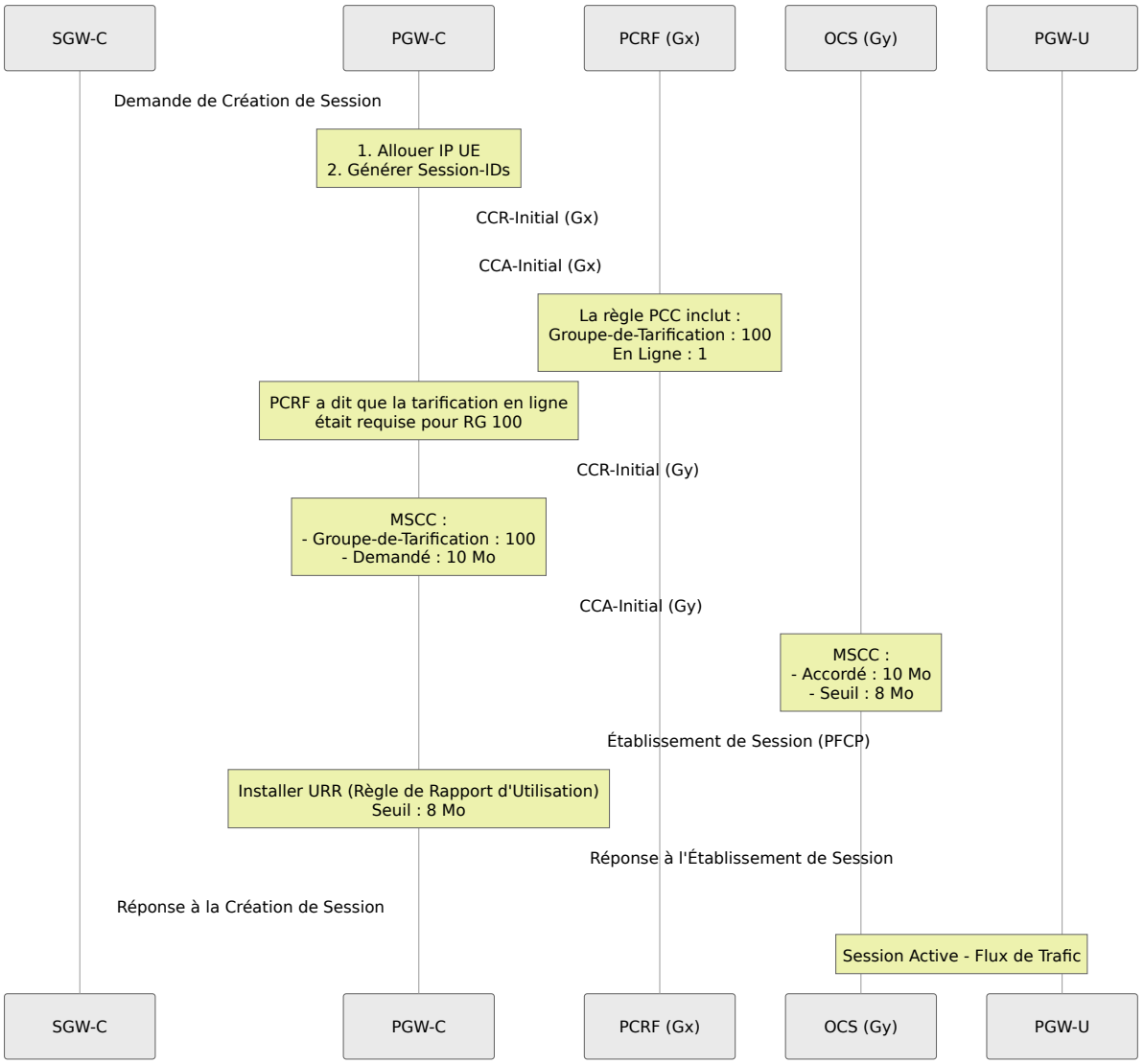
```
config :pgw_c,
  gy: %&#123;
    enabled: false  # Désactiver pour les tests
  &#125;
```

Hybride (certains prépayés, certains postpayés) :

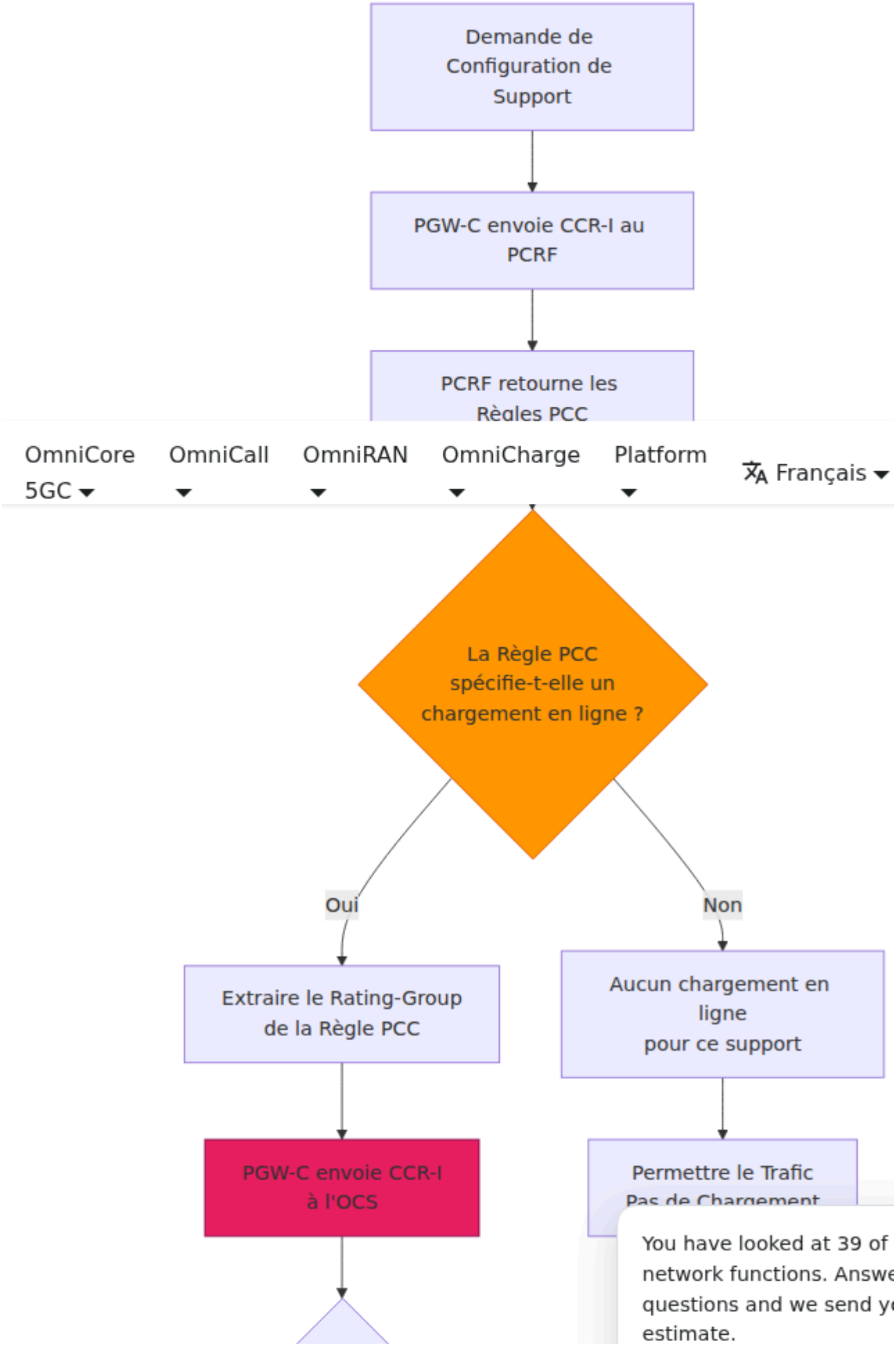
```
config :pgw_c,
  gy: %&#123;
    enabled: true,  # Contrôlé par abonné par le PCRF
    timeout_action: :allow,  # Ne pas bloquer les postpayés en cas
d'échec de l'OCS
    no_credit_action: :terminate
  &#125;
```

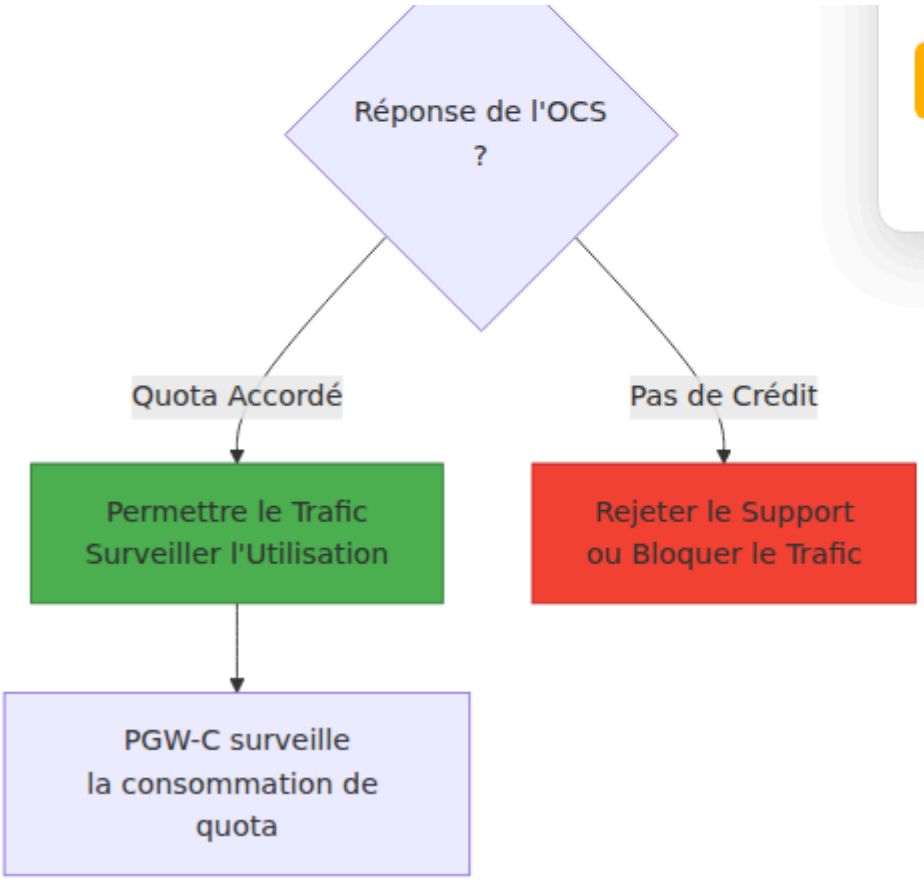
Flux de Messages

Session Réussie avec Tarification en Ligne



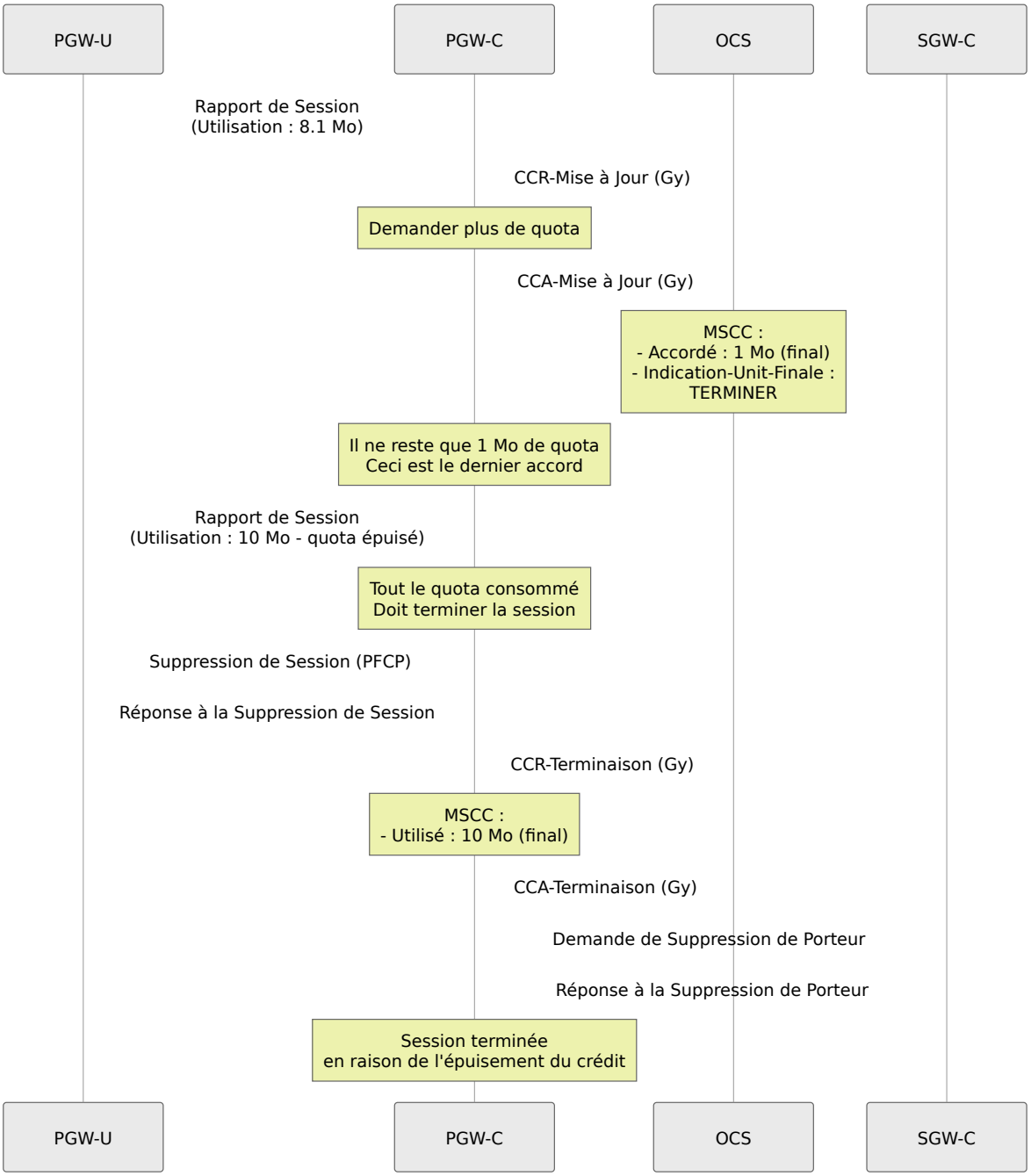
Ré-autorisation de Quota (CCR-Mise à Jour)


Demande de Configuration de Support
PGW-C envoie CCR-I au PCRF
PCRF retourne les Règles PCC
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Français
La Règle PCC spécifie-t-elle un chargement en ligne ?
Oui
Non
Extraire le Rating-Group de la Règle PCC
Aucun chargement en ligne pour ce support
PGW-C envoie CCR-I à l'OCS
Permettre le Trafic
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

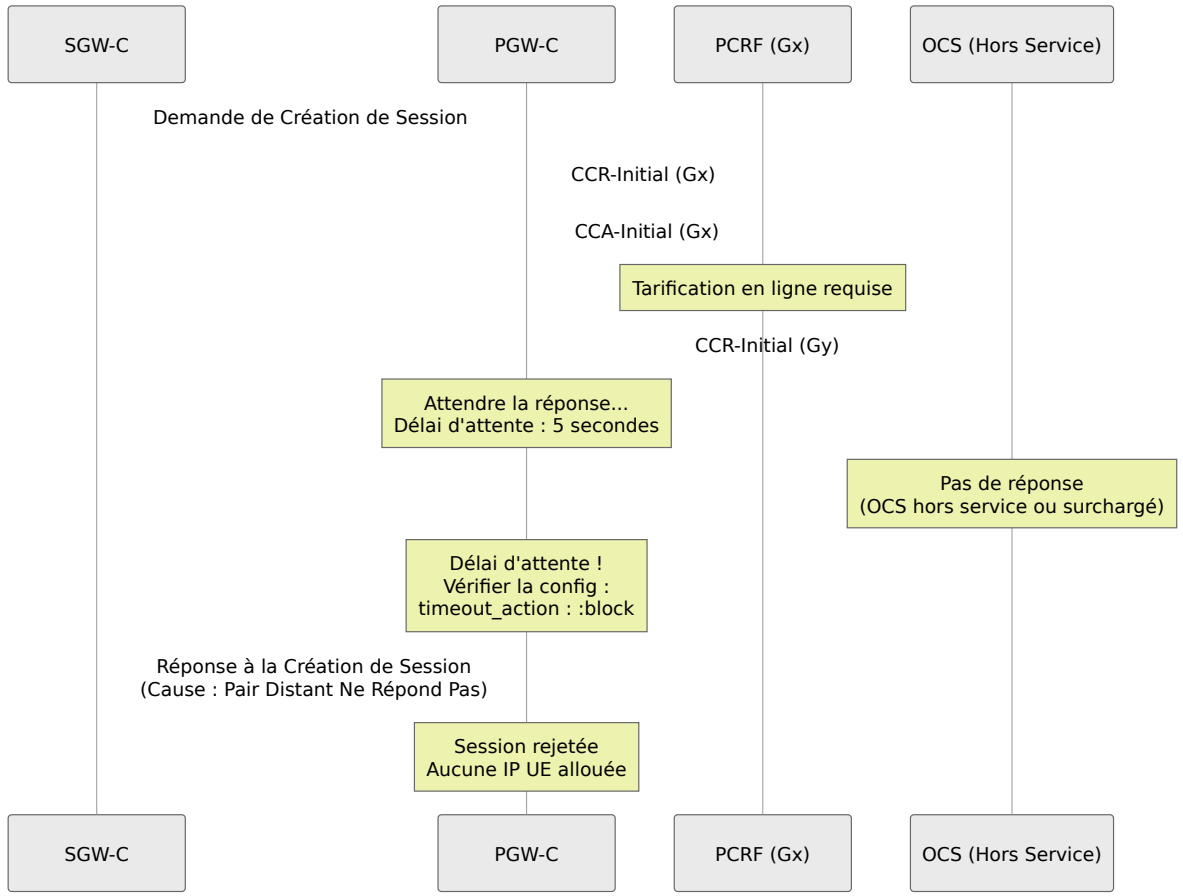


Get an estima
No thanks

Épuisement de Crédit (Unité Finale)


PGW-U
PGW-C
OCS
SGW-C
Rapport de Session
(Utilisation : 8.1 Mo)
CCR-Mise à Jour (Gy)
Demander plus de quota
CCA-Mise à Jour (Gy)
MSCC :
- Accordé : 1 Mo (final)
- Indication-Unit-Finale :
TERMINER
Il ne reste que 1 Mo de quota
Ceci est le dernier accord
Rapport de Session
(Utilisation : 10 Mo - quota épuisé)
Tout le quota consommé
Doit terminer la session
Suppression de Session (PFCP)
Réponse à la Suppression de Session
CCR-Terminaison (Gy)
MSCC :
- Utilisé : 10 Mo (final)
CCA-Terminaison (Gy)
Demande de Suppression de Porteur
Réponse à la Suppression de Porteur
Session terminée
en raison de l'épuisement du crédit
PGW-U
PGW-C
OCS
SGW-C

Gestion des Délais d'Attente OCS



Gestion des Erreurs

Codes de Résultat

Codes de Succès :

Code	Nom	Action
2001	DIAMETER_SUCCESS	Continuer avec le quota accordé

Échecs Transitoires (4xxx) :

Code	Nom	Action PGW-C
4010	DIAMETER_TOO_BUSY	Réessayer avec un délai
4011	DIAMETER_UNABLE_TO_COMPLY	Enregistrer l'erreur, peut réessayer
4012	DIAMETER_CREDIT_LIMIT_REACHED	Terminer ou rediriger

Échecs Permanents (5xxx) :

Code	Nom	Action PGW-C
5003	DIAMETER_AUTHORIZATION_REJECTED	Rejeter la session
5031	DIAMETER_USER_UNKNOWN	Rejeter la session (abonné invalide)

Codes de Résultat par Service

Important : Le Code-Résultat peut apparaître à **deux niveaux** :

1. **Niveau de message** - Résultat global
2. **Niveau MSCC** - Résultat par service

Exemple :

```
CCA-Initial
├── Code-Résultat : DIAMETER_SUCCESS (2001)  ← Niveau de message : OK
└── Contrôle de Crédit de Services Multiples
    ├── [Groupe-de-Tarification : 100]
    │   └── Code-Résultat : DIAMETER_SUCCESS (2001)  ← RG 100 : OK
    └── [Groupe-de-Tarification : 200]
        └── Code-Résultat : DIAMETER_CREDIT_LIMIT_REACHED (4012)
← RG 200 : Pas de crédit
```

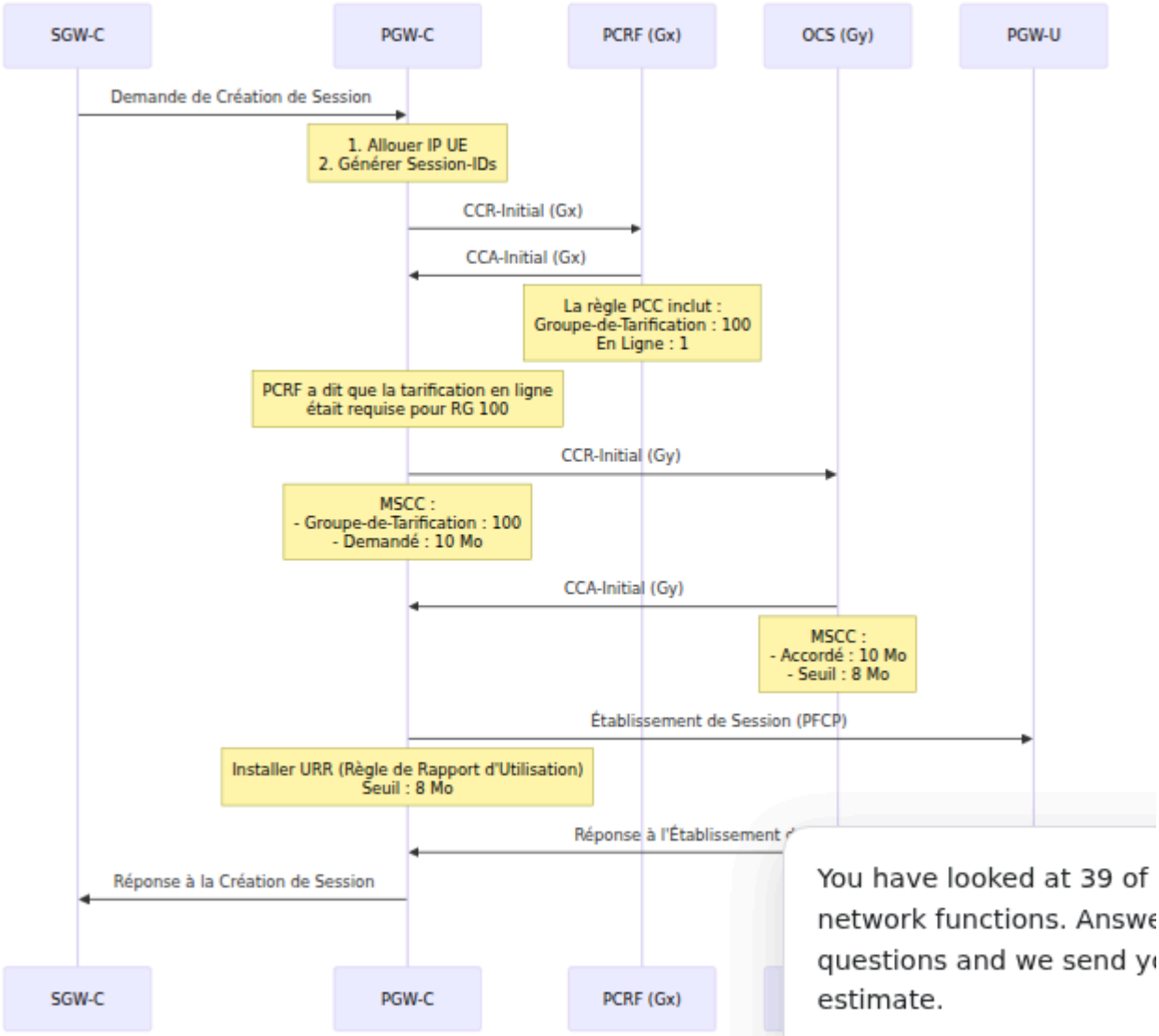
Comportement PGW-C :

- Autoriser le trafic pour le Groupe-de-Tarification 100
- Bloquer le trafic pour le Groupe-de-Tarification 200

Intégration avec Gx

L'interface Gx (contrôle de politique PCRF) détermine si la tarification en ligne est requise et fournit le Groupe-de-Tarification qui pilote la tarification Gy. Voir Interface Diameter Gx pour la documentation complète sur le contrôle de politique.

Relation Gx et Gy



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

Flux d'Intégration

1. Configuration du Porteur :

```
PGW-C reçoit la Demande de Création de Session
    ↓
Envoyer CCR-I au PCRF (Gx)
    ↓
Recevoir CCA-I avec Règles PCC
    ↓
Analyser les Règles PCC :
    - La règle a-t-elle un Groupe-de-Tarification ?
    - Est-ce que En Ligne = 1 ?
    ↓
Si OUI :
    Envoyer CCR-I à l'OCS (Gy) avec le Groupe-de-Tarification
    ↓
    Recevoir CCA-I avec quota
    ↓
    Si quota accordé : Procéder
    Si pas de crédit : Rejeter le porteur
Si NON :
    Procéder sans tarification en ligne
```

2. Mise à Jour de Politique Dynamique (RAR du PCRF) :

```
PCRF envoie RAR (Re-Auth-Request) sur Gx
    ↓
Nouvelle Règle PCC ajoutée avec En Ligne=1, Groupe-de-
Tarification=200
    ↓
PGW-C envoie CCR-U à l'OCS (Gy)
    - Ajouter MSCC pour le Groupe-de-Tarification 200
    ↓
OCS accorde quota pour le nouveau service
    ↓
Installer un porteur dédié avec tarification en ligne
```

Dépannage

Problèmes Courants

1. Délais d'Attente CCR-Initial vers OCS

Symptômes :

- Les sessions échouent avec "délai d'attente OCS"
- Journal : "CCR-Initial (Gy) délai d'attente"

Causes Possibles :

- OCS non accessible
- IP OCS incorrecte dans la configuration
- Pare-feu bloquant le port Diameter (3868)
- OCS surchargé

Résolution :

```
# Tester la connectivité réseau
ping &lt;ocs_ip&gt;

# Tester le port Diameter (TCP 3868)
telnet &lt;ocs_ip&gt; 3868

# Vérifier la configuration
# Assurez-vous que le pair OCS est configuré dans peer_list
```

2. Sessions Rejetées par OCS

Symptômes :

- CCA-I avec Code-Résultat != 2001
- Échec de la Réponse à la Création de Session

Codes de Résultat Courants :

Code de Résultat	Cause Probable	Résolution
4012	Limite de crédit atteinte	L'abonné doit recharger
5003	Autorisation rejetée	Vérifier les permissions de l'abonné
5031	Utilisateur inconnu	Provisionner l'abonné dans l'OCS

Étapes de Débogage :

1. Vérifier les journaux OCS pour la raison du rejet
2. Vérifier le solde de l'abonné dans l'OCS
3. Vérifier que l'IMSI/MSISDN dans CCR-I correspond à l'enregistrement de l'abonné

3. Épuisement de Quota Non Détecté

Symptômes :

- L'utilisateur continue d'utiliser des données après épuisement du solde
- Aucun CCR-Mise à Jour envoyé

Causes Possibles :

- URR (Règle de Rapport d'Utilisation) non installée dans PGW-U
- Seuil non configuré correctement
- Rapports de Session PFCP non reçus

Étapes de Débogage :

1. Vérifier URR dans l'Établissement de Session PFCP :

```
Cr��er URR
├── URR-ID : 1
├── Méthode-de-Mesure : VOLUME
├── Seuil-de-Volume : 8000000 (8 Mo)
└── Déclencheurs-de-Rapport : VOLUME_THRESHOLD
```

2. Vérifier les journaux PGW-U pour les rapports d'utilisation
3. Vérifier `quota_threshold_percentage` dans la config

4. Groupe-de-Tarification Incorrect

Symptômes :

- OCS rejette avec "Groupe-de-Tarification Inconnu"
- Les sessions échouent

Cause :

- Groupe-de-Tarification dans CCR-I ne correspond pas à la configuration OCS
- PCRF a provisionné un Groupe-de-Tarification invalide

Résolution :

1. Vérifier le Groupe-de-Tarification dans la Règle PCC du PCRF
2. Vérifier la configuration OCS pour les Groupes-de-Tarification valides
3. Assurer le mapping entre les Règles PCC et les tarifs OCS

Surveillance

Métriques Clés

La liste complète, avec étiquettes, se trouve dans le Guide de Surveillance.

```
# Taux de messages Gy (type_message est ro_CCR, ro_CCA, ro_RAR ou
ro_RAA)
rate(gy_outbound_messages_total&amp;#123;message_type="ro_CCR"&amp;#125;
[5m])
rate(gy_inbound_messages_total&amp;#123;message_type="ro_CCA"&amp;#125;[5m])

# Résultats CCA par type de demande (initial, mise à jour,
terminaison) et classe de code de résultat
sum by (cc_request_type, result_code_class)
(rate(gy_inbound_responses_total[5m]))

# Taux d'erreurs Gy
rate(gy_inbound_errors_total[5m])

# CCR envoyés sans CCA en retour (délais d'attente OCS ou réponses
perdues)
sum(rate(gy_outbound_messages_total&amp;#123;message_type="ro_CCR"&amp;#125;
[5m]))
  - sum(rate(gy_inbound_responses_total[5m]))

# Temps de réponse aller-retour OCS (CCR à CCA)
histogram_quantile(0.95, sum by (le)
(rate(gy_outbound_transaction_duration_bucket[5m])))

# Sessions Gy ouvertes
gy_session_id_registry_count
```

Alertes

```
# Alerte sur un taux d'erreur Gy élevé
- alert: GyErrorRateHigh
  expr: rate(gy_inbound_errors_total[5m]) > 0.1
  for: 5m
  annotations:
    summary: "Taux d'erreur Gy élevé détecté"

# Alerte lorsque l'OCS ne répond pas aux CCR (délais d'attente ou rép
- alert: OcsNotAnswering
  expr: |
    sum by (instance) (rate(gy_outbound_messages_total&#123;message_t
      - sum by (instance) (rate(gy_inbound_responses_total[5m])) > 0.
  for: 2m
  annotations:
    summary: "Les CCR à l'OCS ne sont pas répondus"

# Alerte lorsque l'OCS refuse de nouvelles sessions
- alert: OcsRejectingCcrI
  expr:
sum(rate(gy_inbound_responses_total&#123;cc_request_type="initial",re
[5m])) > 0.1
  for: 5m
  annotations:
    summary: "L'OCS rejette les CCR-I (nouvelles sessions Gy)"
```

Contrôle de Crédit Gy (API OAM)

La tarification en ligne est inspectée et exercée via l'**API REST OAM**, servie sur HTTPS/TLS à `https://<host>:8443`. La documentation interactive de l'API (Swagger UI) est disponible à `https://<host>:8443/api/docs`.

État de la Tarification en Ligne

```
GET /api/charging                # toutes les sessions de
tarification en ligne + état de crédit en direct
GET /api/charging?search=&lt;imsi&gt;  # filtrer les sessions par
IMSI
GET /api/charging/&lt;imsi&gt;         # état de contrôle de
crédit d'une session par IMSI
```

Exemple :

```
curl -k https://localhost:8443/api/charging/310170123456789
```

`GET /api/charging` retourne l'état de tarification en ligne (quota accordé vs. utilisé, validité, l'OCS Indication-Unit-Finale, et état de redirection jardin clos / hors crédit) pour chaque session active. Ajoutez `?search=<imsi>` pour restreindre la liste, ou ajoutez l'IMSI à `GET /api/charging/<imsi>` pour récupérer l'état de contrôle de crédit d'un seul abonné.

Simulateur de Contrôle de Crédit Gy

```
POST /api/gy_simulator
```

Envoie une Demande de Contrôle de Crédit synthétique à l'OCS configuré et retourne la réponse analysée — Code-Résultat au niveau de la commande, quota accordé, temps de validité, et toute Indication-Unit-Finale — afin que la tarification en ligne puisse être exercée **sans trafic d'abonné en direct**. Ceci est le remplacement de l'API OAM pour l'ancien simulateur Gy basé sur le navigateur. Il nécessite que Gy soit activé (`gy.enabled: true`) ; la demande est envoyée depuis l'hôte d'origine Gy du PGW-C vers le domaine OCS configuré.

Corps de la Demande (`application/json`) :

Champ	Type	Description
`imsi`	string	**Requis.** IMSI de l'abonné.
`msisdn`	string	MSISDN optionnel (E.164). Remplit `User-Name` comme `<msisdn>@<apn>`.
`apn`	string	APN optionnel.
`requested_units`	integer	Quota demandé optionnel en octets (Unité-Service-Demandée → CC-Octets-Total).
`rating_group`	integer	Groupe-de-Tarification / clé de tarification optionnel (par défaut `1`).
`service_id`	integer	Identifiant-Service optionnel.
`cc_request_type`	string	`initial` (par défaut), `update`, ou `terminate` (ou les numériques `1`/`2`/`3`).

Exemple — demander 1 Mo de quota initial pour un abonné :

```
curl -k -X POST https://localhost:8443/api/gy_simulator \
  -H 'content-type: application/json' \
  -d
'&#123;"imsi":"310170123456789","msisdn":"14155551234","apn":"interne
```

Exemple de réponse (CCA-Initial accordant 1 Mo, valide 1 heure) :

```
&amp;#123;
  "session_id": "omni-pgw_c.epc...;734825199;app_gy_sim",
  "cc_request_type": 1,
  "cc_request_type_name": "initial",
  "result_code": "diameter_success",
  "accepted": true,
  "ocs_host": "ocs.epc.example.org",
  "ocs_realm": "epc.example.org",
  "granted": &amp;#123; "total_octets": 1000000, "input_octets": null,
"output_octets": null, "time": null &amp;#125;,
  "validity_time": 3600,
  "final_unit_action": null,
  "final_unit_redirect": null,
  "mscc_result_code": "diameter_success"
&amp;#125;
```


En cas d'échec, le point de terminaison retourne `422` (`imsi` manquant) ou `503` (Gy désactivé, ou l'OCS n'a pas répondu — le champ `reason` porte l'erreur sous-jacente, par exemple `:timeout`).

Cas d'Utilisation Opérationnels

- **Inspecter le quota d'un abonné** — `GET /api/charging/<imsi>` retourne le quota accordé actuel de la session et l'état de crédit.
- **Tester l'intégration OCS sans trafic** — `POST /api/gy_simulator` pour confirmer que l'OCS accorde le quota / Indication-Unit-Finale attendu pour un IMSI, groupe de tarification, ou service donné.
- **Dépanner les problèmes de tarification** — localiser un abonné affecté avec `GET /api/charging?search=<imsi>` et examiner son état de contrôle de crédit en direct.

Pour les tendances historiques (taux de messages, épuisement de quota, délais d'attente OCS), utilisez les métriques Prometheus décrites dans [Monitoring](#monitoring) ci-dessus.

Documentation Connexe

Facturation et Politique
- **[Interface Diameter Gx](./diameter-gx.md)** - Contrôle de politique PCRF, règles PCC qui déclenchent la facturation en ligne
- **[Format CDR de Données](./data-cdr-format.md)** - Enregistrements de facturation hors ligne pour la facturation postpayée
- **[Guide de Configuration](./configuration.md)** - Paramètres complets de configuration de la facturation en ligne

Gestion des Sessions
- **[Gestion des Sessions](./session-management.md)** - Cycle de vie de la session PDN, gestion des porteurs
- **[Interface PFCP](./pfcp-interface.md)** - Rapport d'utilisation depuis PGW-U via URRs
- **[Interface S5/S8](./s5s8-interface.md)** - Configuration et démantèlement du porteur GTP-C

Opérations
- **[Guide de Surveillance](./monitoring.md)** - Métriques Gy, suivi des quotas, alertes de délai d'attente OCS
- **[Allocation IP UE](./ue-ip-allocation.md)** - Configuration du pool IP pour les sessions facturées

---

[Retour au Guide des Opérations](../OPERATIONS.md)

Documentation de l'interface Gn/Gp

Communication GTP-C avec SGSN (réseaux 2G/3G)

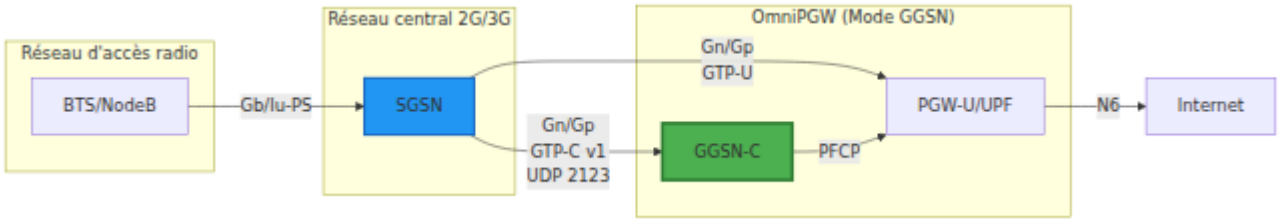
OmniPGW par Omnitouch Network Services

Vue d'ensemble

L'**interface Gn/Gp** permet à OmniPGW de fonctionner comme un **GGSN** (Gateway GPRS Support Node) pour les réseaux 2G/3G en utilisant le protocole **GTP-C v1**. Cette interface gère la gestion du contexte PDP (Packet Data Protocol) entre le GGSN et les SGSN.

- **Interface Gn** : Se connecte aux SGSN au sein du même PLMN (réseau domestique)
- **Interface Gp** : Se connecte aux SGSN dans d'autres PLMN (scénarios de roaming)

Les deux interfaces utilisent des protocoles identiques - la distinction est purement topologique.



Principales différences avec PGW (S5/S8)

Aspect	GGSN (Gn/Gp)	PGW (S5/S8)
Version du protocole	GTP-C v1	GTP-C v2
Spécification 3GPP	TS 29.060	TS 29.274
Concept de session	Contexte PDP	Session PDN + Porteuses
Identifiant de porteuse	NSAPI (0-15)	EBI (5-15)
Interface de politique	Aucune (politique statique)	Gx (PCRF)
Modèle QoS	Profil QoS R99	QoS de porteuse EPS (QCI)
Port	2123 (partagé)	2123

Détails du protocole

GTP-C Version 1

- **Protocole** : GTP-C v1 (3GPP TS 29.060)
- **Transport** : UDP
- **Port** : 2123 (partagé avec S5/S8 GTP-C v2)
- **Type d'interface** : Plan de contrôle

TEID (Identifiant de point de terminaison de tunnel)

Chaque contexte PDP a des TEID uniques pour le plan de contrôle et le plan utilisateur :

- **TEID de contrôle GGSN** : Alloué par OmniPGW pour les messages de contrôle entrants
- **TEID de données GGSN** : Alloué par UPF pour les donn��es utilisateur entrantes
- **TEID de contrôle SGSN** : Fournit par SGSN pour les messages de contrôle sortants
- **TEID de données SGSN** : Fournit par SGSN pour les données utilisateur sortantes

```
Flux de message :
  SGSN → GGSN : TEID de destination = TEID de contrôle du GGSN
  GGSN → SGSN : TEID de destination = TEID de contrôle du SGSN
```

Support CUPS (Séparation du plan de contrôle et du plan utilisateur)

OmniPGW prend en charge l'architecture CUPS pour 2G/3G :

- **Adresse GSN pour la signalisation** : IP du plan de contrôle d'OmniPGW (à partir de `gn.local_ipv4_address`)
- **Adresse GSN pour le trafic utilisateur** : Adresse IP de l'UPF (à partir de la réponse d'établissement de session PFCP)

Cela permet au GGSN-C et au PGW-U/UPF de fonctionner sur des nœuds séparés avec des adresses IP différentes, ce qui est standard pour les déploiements CUPS.

NSAPI (Identifiant de point d'accès au service réseau)

Le NSAPI identifie les contextes PDP individuels au sein d'une session d'abonné :

- **Plage** : 5-15 (0-4 réservés)
- **Utilisation** : Plusieurs contextes PDP par abonné possibles
- **Équivalent** : Semblable à l'EBI dans LTE/EPC

Configuration

Configuration de base

```
# config/runtime.exs
config :pgw_c,
  # L'interface Gn/Gp est activée automatiquement lorsque la
configuration gn est présente
  gn: %{
    # Adresse IPv4 locale pour l'interface Gn
    local_ipv4_address: "10.0.0.20",

    # Optionnel : Adresse IPv6 locale
    local_ipv6_address: nil,

    # Port pour GTP-C v1 (partagé avec S5/S8)
    local_port: 2123
  },

  # Serveurs DNS retournés dans PCO (partagés avec PGW)
  dns: %{
    primary_ipv4: {8, 8, 8, 8},
    secondary_ipv4: {8, 8, 4, 4}
  }
```

Paramètres de l'interface Gn

Paramètre	Type	Requis	Par défaut	Description
`local_ipv4_address`	String	Oui	-	Adresse IPv4 à lier pour l'interface Gn. C'est l'adresse GGSN annoncée aux SGSN.
`local_ipv6_address`	String	Non	`nil`	Adresse IPv6 optionnelle. La plupart des réseaux 2G/3G sont uniquement IPv4.
`local_port`	Integer	Non	2123	Port UDP pour GTP-C. Port standard selon 3GPP TS 29.060. Partagé avec S5/S8.

Paramètres DNS

Paramètre	Type	Requis	Par défaut	Description
`primary_ipv4`	Tuple	Non	`{8, 8, 8, 8}`	Adresse IPv4 du serveur DNS principal retournée dans PCO.
`secondary_ipv4`	Tuple	Non	`{8, 8, 4, 4}`	Adresse IPv4 du serveur DNS secondaire retournée dans PCO.
`primary_ipv6`	Tuple	Non	`nil`	Adresse IPv6 du serveur DNS principal.
`secondary_ipv6`	Tuple	Non	`nil`	Adresse IPv6 du serveur DNS secondaire.

Exigences réseau

Règles de pare-feu :

```
# Autoriser GTP-C depuis le réseau SGSN
iptables -A INPUT -p udp --dport 2123 -s <sgsn_network>/24 -j
ACCEPT

# Autoriser GTP-C sortant vers SGSN
iptables -A OUTPUT -p udp --dport 2123 -d <sgsn_network>/24 -j
ACCEPT
```

Types de messages

Gestion du contexte PDP

Demande de création de contexte PDP

Direction : SGSN → GGSN

But : Établir un nouveau contexte PDP pour la connectivité des données

Éléments d'information clés (IE) :

Nom de l'IE	Type	Description
IMSI	Identité	Identité de l'abonné mobile international
NSAPI	Entier	Identifiant de point d'accès au service réseau (5-15)
TEID Données I	Entier	Point de terminaison de tunnel du plan utilisateur SGSN
TEID Contrôle	Entier	Point de terminaison de tunnel du plan de contrôle SGSN
Adresse de l'utilisateur final	Adresse	Type de PDN demandé et IP optionnelle
Nom de point d'accès	String	APN pour la connexion
Adresse GSN (Signalisation)	IP	Adresse du plan de signalisation SGSN
Adresse GSN (Utilisateur)	IP	Adresse du plan utilisateur SGSN
Profil QoS	Binaire	Paramètres QoS demandés
MSISDN	Identité	Numéro de téléphone mobile (optionnel)
Mode de sélection	Enum	Mode de sélection APN
Options de configuration de protocole	Options	Options de protocole supplémentaires

Exemple :

```
Demande de création de contexte PDP
├── IMSI: 310260123456789
├── NSAPI: 5
├── TEID Données I: 0x12345678
├── TEID Contrôle: 0x87654321
├── Adresse de l'utilisateur final: IPv4 (dynamique)
├── APN: internet
├── Adresse GSN (Signalisation): 10.1.1.100
├── Adresse GSN (Utilisateur): 10.1.1.100
├── Profil QoS: [Paramètres QoS R99]
└── PCO: [Demande DNS, etc.]
```

Réponse à la création de contexte PDP

Direction : GGSN → SGSN

But : Accuser réception de la création du contexte PDP avec les ressources allouées

Éléments d'information clés :

Nom de l'IE	Type	Description
Cause	Résultat	Code de succès ou d'erreur
Réorganisation requise	Booléen	Indique si une réorganisation est nécessaire
Récupération	Compteur	Compteur de redémarrage du GGSN
TEID Données I	Entier	Point de terminaison de tunnel du plan utilisateur GGSN
TEID Contrôle	Entier	Point de terminaison de tunnel du plan de contrôle GGSN
NSAPI	Entier	NSAPI confirmé
ID de facturation	Binaire	ID de corrélation de facturation
Adresse de l'utilisateur final	Adresse	Adresse IP UE allouée
Adresse GSN (Signalisation)	IP	Adresse du plan de signalisation GGSN
Adresse GSN (Utilisateur)	IP	Adresse du plan utilisateur GGSN
Profil QoS	Binaire	Paramètres QoS négociés
Options de configuration de protocole	Options	Serveurs DNS, etc.

Réponse de succès :

```
Réponse à la création de contexte PDP
├── Cause: Demande acceptée (128)
├── TEID Données I: 0xAABBCCDD
├── TEID Contrôle: 0xDDCCBBAA
├── NSAPI: 5
├── ID de facturation: 0x11223344
├── Adresse de l'utilisateur final
│   └── IPv4: 100.64.1.42
├── Adresse GSN (Signalisation): 10.0.0.20
├── Adresse GSN (Utilisateur): 10.0.0.20
├── Profil QoS: [QoS négocié]
└── PCO
    ├── DNS principal: 8.8.8.8
    └── DNS secondaire: 8.8.4.4
```

Demande de mise à jour de contexte PDP

Direction : SGSN → GGSN

But : Modifier un contexte PDP existant (par exemple, relocalisation SGSN, changement de QoS)

Scénarios courants :

- Transfert inter-SGSN (mise à jour de la zone de routage)
- Renégociation de QoS
- Changement d'adresse SGSN

Éléments d'information clés :

Nom de l'IE	Description
NSAPI	Identifiant de contexte PDP
TEID Données I	Nouveau TEID utilisateur SGSN
Adresse GSN	Nouvelles adresses SGSN
Profil QoS	QoS mise à jour (optionnel)

Réponse à la mise à jour de contexte PDP

Direction : GGSN → SGSN

But : Accuser réception de la mise à jour du contexte PDP

Éléments d'information clés :

Nom de l'IE	Description
Cause	Code de succès ou d'erreur
TEID Données I	TEID utilisateur GGSN
Adresse GSN	Adresses GGSN
Profil QoS	QoS négociée

Demande de suppression de contexte PDP

Direction : SGSN → GGSN

But : Terminer un contexte PDP

Éléments d'information clés :

Nom de l'IE	Description
NSAPI	Contexte PDP à supprimer
Indication de suppression	Supprimer tous les contextes pour cet abonné

Réponse à la suppression de contexte PDP

Direction : GGSN → SGSN

But : Accuser réception de la suppression du contexte PDP

Éléments d'information clés :

Nom de l'IE	Description
Cause	Code de succès ou d'erreur

Gestion des chemins

Demande / Réponse Echo

Direction : Bidirectionnelle

But : Maintien de chemin et détection de redémarrage

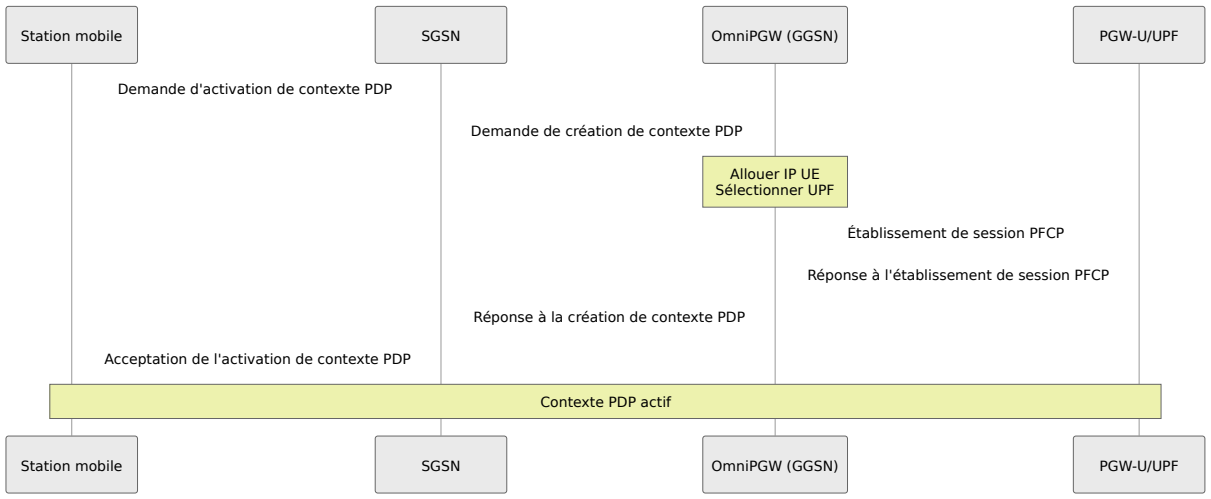
Éléments d'information clés :

Nom de l'IE	Description
Récupération	Compteur de redémarrage pour détecter les redémarrages de nœud

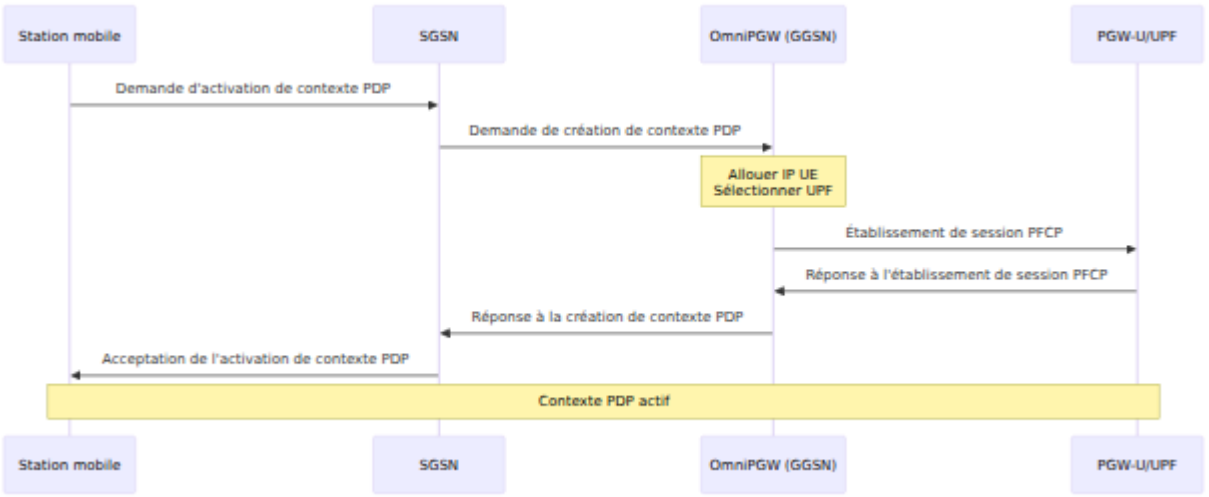
---

Flux de messages

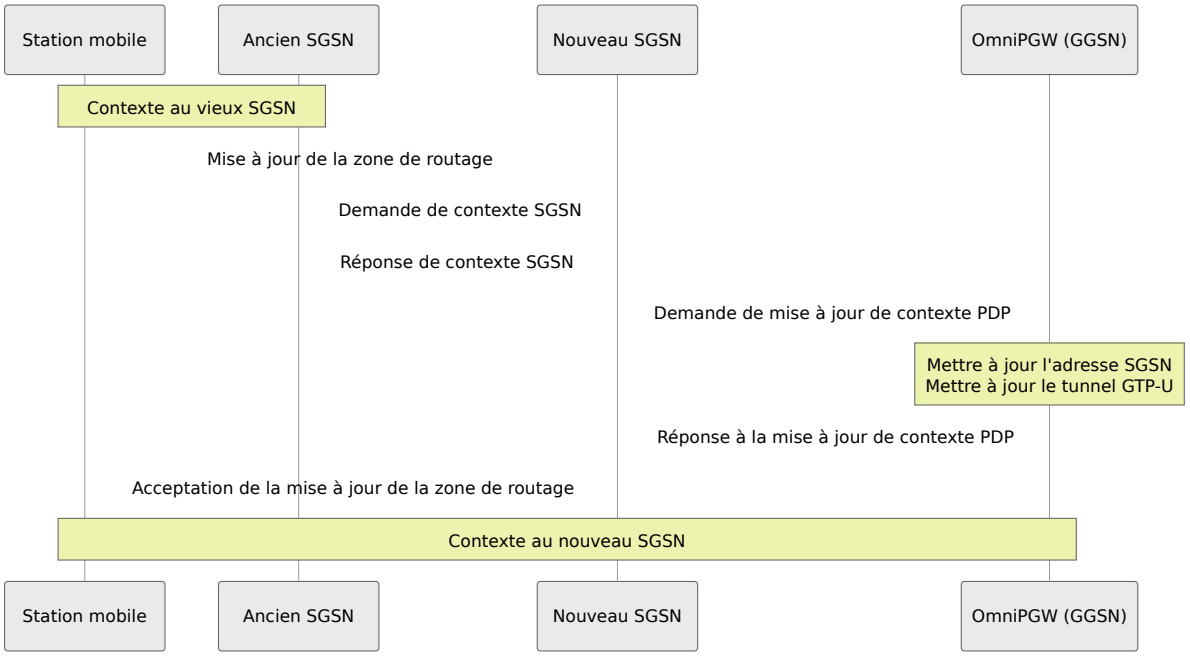
Activation du contexte PDP



Désactivation du contexte PDP



Transfert inter-SGSN



Infrastructure partagée

La fonctionnalité GGSN d'OmniPGW partage l'infrastructure avec le mode PGW :

Allocation d'adresse IP UE

Utilise le même AddressRegistry et la configuration de sous-réseau que PGW. Voir Allocation IP UE pour les détails de configuration.

Gestion de session UPF/PFCP

Utilise le même établissement de session PFCP que PGW :

- Règles PDR/FAR/QER créées par contexte PDP
- Sélection UPF via la même configuration de sélection upf
- Voir Interface PFCP pour les détails

Facturation en ligne (Gy)

Intégration Diameter Gy optionnelle disponible :

- Contrôle de crédit pour les abonnés prépayés
- Même configuration Gy que le mode PGW
- Voir Interface Diameter Gy pour la configuration

Génération de CDR

Utilise la même génération de CDR que PGW. Voir Format de CDR de données pour les détails.

Codes de cause

Succès

Code	Nom	Description
128	Demande acceptée	Opération réussie

Causes de rejet

Code	Nom	Quand utilisé
192	Inexistant	Contexte PDP non trouvé
193	Format de message invalide	Message GTP mal formé
194	IMSI inconnu	Abonné non trouvé
195	MS est détaché GPRS	MS non attaché
196	MS ne répond pas au GPRS	MS injoignable
197	MS refuse	MS a rejeté la demande
198	Version non supportée	Incompatibilité de version GTP
199	Pas de ressources disponibles	Épuisement des ressources
200	Service non supporté	Service non pris en charge
201	IE obligatoire incorrect	Échec de la validation de l'IE
202	IE obligatoire manquant	IE requis absent
203	IE optionnel incorrect	IE optionnel invalide
204	Échec système	Erreur interne
205	Restriction de roaming	Roaming non autorisé
206	Mismatch de signature P-TMSI	Échec de sécurité
207	Connexion GPRS suspendue	Connexion suspendue

Code	Nom	Quand utilisé
208	Échec d'authentification	Auth échouée
209	Échec d'authentification utilisateur	Échec d'auth utilisateur

---

Surveillance

Métriques de l'interface Gn

```
# Compteurs de messages
gn_inbound_messages_total{message_type="create_pdp_context_request"}
gn_inbound_messages_total{message_type="update_pdp_context_request"}
gn_inbound_messages_total{message_type="delete_pdp_context_request"}

gn_outbound_messages_total{message_type="create_pdp_context_response"
gn_outbound_messages_total{message_type="update_pdp_context_response"
gn_outbound_messages_total{message_type="delete_pdp_context_response"

# Latence des transactions
gn_outbound_transaction_duration_bucket
```

Requêtes utiles

Taux de création de contexte PDP :

```
rate(gn_inbound_messages_total{message_type="create_pdp_context_reque
[5m])
```

Taux de succès :

```
sum(rate(gn_outbound_messages_total{message_type="create_pdp_context_
[5m]))
  /
sum(rate(gn_inbound_messages_total{message_type="create_pdp_context_r
[5m]))
```

Contextes PDP actifs :

```
gn_inbound_messages_total{message_type="create_pdp_context_request"}
  -
gn_inbound_messages_total{message_type="delete_pdp_context_request"}
```

Dépannage

Problème : Pas de réponse à la demande de création de contexte PDP

Symptômes :

- SGSN envoie une demande de création de contexte PDP
- Pas de réponse reçue
- Délai d'attente au SGSN

Causes possibles :

1. Problème de connectivité réseau
2. OmniPGW ne répond pas à l'IP configurée
3. Pare-feu bloquant UDP 2123
4. Échec de l'établissement UPF/PFCP

Débogage :

```
# Vérifiez qu'OmniPGW écoute
netstat -ulnp | grep 2123

# Vérifiez les paquets entrants
tcpdump -i any -n port 2123

# Vérifiez la configuration
grep "local_ipv4_address" config/runtime.exs

# Vérifiez la connectivité PFCP
curl http://pgw:9090/metrics | grep pfcp
```

Problème : Création de contexte PDP rejetée

Symptômes :

- Réponse à la création de contexte PDP avec cause d'erreur
- Contexte PDP non établi

Causes courantes :

```
Cause 199 (Pas de ressources disponibles):
  → Pool IP épuisé
  → Vérifiez : curl http://pgw:9090/metrics | grep
address_registry_count
  → Élargir le pool IP dans ue.subnet_map

Cause 202 (IE obligatoire manquant):
  → SGSN n'envoie pas les IEs requis
  → Vérifiez la configuration SGSN
  → Examinez le message dans tcpdump

Cause 204 (Échec système):
  → Échec UPF/PFCP
  → Vérifiez l'état d'association PFCP
  → Examinez les journaux d'OmniPGW
```

Problème : Échec de la mise à jour du contexte PDP

Symptômes :

- Réponse à la mise à jour de contexte PDP avec cause d'erreur
- Échec du transfert

Causes possibles :

- Contexte PDP non trouvé (TEID différent)
- Mismatch de NSAPI
- Session déjà supprimée

Débogage :

```
# Vérifiez les sessions actives
curl http://pgw:9090/metrics | grep session_count

# Examinez les journaux pour la recherche de contexte
journalctl -u omnipgw | grep "PDP Context"
```

Meilleures pratiques

Conception du réseau

1. **Port partagé avec S5/S8**
    - Gn et S5/S8 partagent le port UDP 2123
    - OmniPGW détecte automatiquement la version GTP à partir du message
    - Une seule IP peut servir à la fois 2G/3G et 4G
2. **Connectivité SGSN**
    - Assurez-vous que tous les SGSN peuvent atteindre l'IP du GGSN

- Envisagez un VLAN dédié pour le trafic Gn
- Planifiez le roaming (Gp) si applicable

3. **Planification des adresses IP**

   - Les mêmes pools IP servent à la fois les modes PGW et GGSN
   - Envisagez des pools basés sur APN séparés pour 2G/3G

Performance

1. **Capacité de session**

   - Surveillez la métrique `session_count`
   - Planifiez en fonction du nombre d'abonnés 2G/3G prévu
   - Considérez que 2G/3G a généralement moins de sessions simultanées

2. **Tailles de tampon UDP**

   - Les mêmes recommandations que S5/S8
   - Typique : 4-8 Mo par socket

Considérations de migration

Lors de la migration d'un GGSN autonome vers OmniPGW :

1. **Configuration APN**

   - Assurez-vous que les APN correspondent à la configuration GGSN existante
   - Vérifiez les paramètres du serveur DNS dans PCO

2. **Migration du pool IP**

   - Planifiez soigneusement le basculement du pool IP
   - Envisagez une migration progressive par APN

3. **Mapping QoS**

   - Profils QoS R99 acceptés tels quels

- Pas de traduction QCI nécessaire pour 2G/3G

Documentation connexe

Fonctions principales

- **Guide de configuration** - Référence complète de configuration
- **Gestion de session** - Détails du cycle de vie de la session
- **Allocation IP UE** - Gestion des adresses IP
- **Configuration PCO** - Options de configuration de protocole

Interfaces connexes

- **Interface S5/S8** - Interface équivalente 4G/LTE
- **Interface PFCP** - Coordination du plan utilisateur
- **Interface Diameter Gy** - Facturation en ligne

Opérations

- **Guide de surveillance** - Métriques et alertes
- **Dépannage** - Problèmes courants
- **Format de CDR de données** - Génération de CDR

Références

- 3GPP TS 29.060 - Protocole de tunneling GPRS (GTP) à travers l'interface Gn et Gp
- 3GPP TS 23.060 - Service de description du service de GPRS (General Packet Radio Service)
- 3GPP TS 24.008 - Spécification de l'interface radio mobile couche 3 (profils QoS)

Retour à l'index de documentation

Interface Gn/Gp d'OmniPGW - *par Omnitouch Network Services*

Support PDN IPv6 / Dual-Stack (IPv4v6)

Guide complet pour l'exploitation des connexions PDN compatibles IPv6 à travers l'OmniPGW (plan de contrôle) et l'OmniUPF (plan utilisateur). Cette page couvre comment le type de PDN est négocié et signalé, la configuration requise sur les deux nœuds, et comment la connectivité IPv6 de l'UE est atteinte avec le routage dynamique.

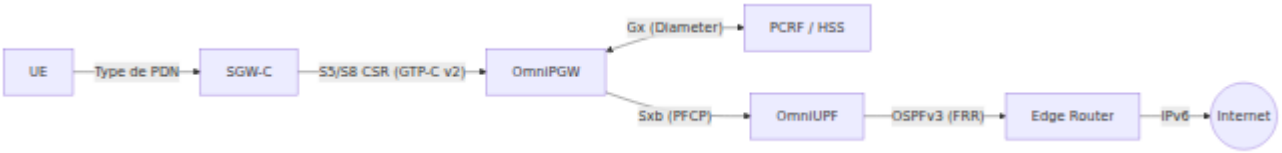
Pour le modèle détaillé de pool d'adresses UE (correspondance APN, allocation, gestion des collisions), voir Allocation d'IP UE. Pour les interfaces PFCP et Gx en général, voir PFCP/Sxb et Diameter Gx.

Table des matières

- Aperçu
- Types de PDN
- Flux de bout en bout
- Configuration PGW-C
- Plan utilisateur (OmniUPF)
- Tables de référence
- Dépannage

Aperçu

Une connexion PDN peut être **IPv4**, **IPv6** ou **IPv4v6** (dual-stack). Le type de PDN est demandé par l'UE/SGW dans la demande de création de session GTPv2 et confirmé par le PGW-C dans la réponse. Fournir un PDN IPv6 de bout en bout nécessite un comportement coordonné à travers trois interfaces et les deux nœuds :



Nœud / Interface	Responsabilité pour IPv6
OmniPGW — S5/S8	Négocier le type de PDN ; allouer l'adresse(s) de l'UE ; retourner l'allocation d'adresse PDN (PAA)
OmniPGW — Gx	Transmettre l'adresse de l'UE au PCRF : `Framed-IP-Address` (IPv4) pour les PDN IPv4/IPv4v6, `Framed-IPv6-Prefix` pour les PDN uniquement IPv6
OmniPGW — Sxb	Signaler l'adresse IPv6 de l'UE et le type de PDN à l'UPF via PFCP
OmniUPF — plan de données	Programmer la recherche de liaison descendante IPv6 ; transférer/encapsuler le trafic utilisateur IPv6
OmniUPF — FRR	Annoncer chaque route hôte IPv6 `/128` de l'UE au routeur de périphérie via OSPFv3

Références : 3GPP TS 29.274 (GTPv2-C), 3GPP TS 29.212 (Gx), 3GPP TS 29.244 (PFCP).

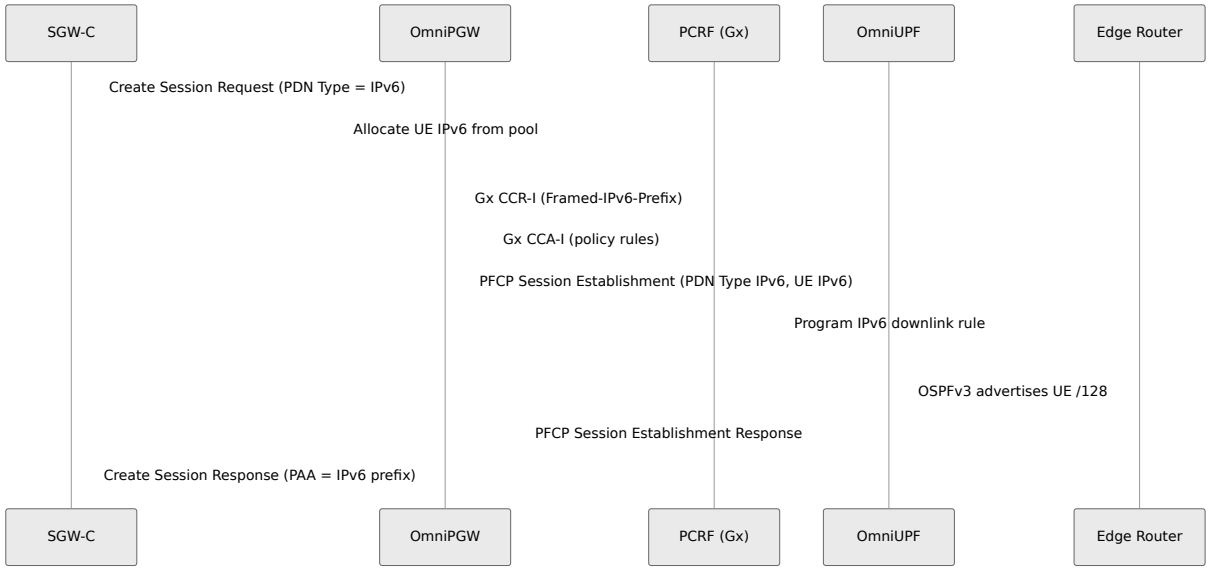
Types de PDN

Valeur	Type de PDN	L'UE reçoit	Contenu du PAA (TS 29.274 §8.14)
1	IPv4	Adresse IPv4	Adresse IPv4 de 4 octets
2	IPv6	Préfixe IPv6 (/64)	octet de longueur de préfixe + adresse de 16 octets
3	IPv4v6	Adresse IPv4 + Préfixe IPv6	octet de longueur de préfixe + 16 octets IPv6 + 4 octets IPv4

Le PGW-C dérive le type de PDN effectif de l'**IE Type de PDN**, revenant au type de l'**IE PAA** si l'IE Type de PDN est absent. Le même type négocié pilote le PAA retourné au SGW, l'IE Type de PDN PFCP envoyé à l'UPF, et quelle famille d'adresses est allouée.

> **IPv4v6 sans pool IPv6.** Si un PDN IPv4v6 est demandé mais que l'APN correspondant n'a pas de plage IPv6 configurée, le PGW-C ne plante **pas** ; il dégrade soit le PDN à IPv4 uniquement, soit le rejette, selon que l'APN est un APN d'urgence (SOS). Voir Demande IPv4v6 lorsque l'APN n'a pas de pool IPv6.

Flux de bout en bout



Une demande de création de session retransmise (même numéro de séquence) est traitée de manière idempotente et produit la même réponse.

Configuration PGW-C

Pool d'adresses IPv6 de l'UE

Les adresses UE sont tirées des pools par APN sous la clé `ue.subnet_map` dans `runtime.exs`. Ajoutez un CIDR IPv6 à côté du CIDR IPv4 pour tout APN qui devrait offrir IPv6 ou dual-stack. Le pool est analysé par famille d'adresses automatiquement.

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # Nom de l'APN => liste de chaînes CIDR (IPv4 et/ou IPv6)
      default: [
        "100.64.16.0/24",
        "2001:db8:6f85:70::/64"
      ]
    }
  }
```

Paramètre	Type	Requis	Par défaut	Description
`ue.subnet_map`	Map	Oui	-	Mappe un nom d'APN (ou `:default`) à une liste de chaînes CIDR. `:default` est le fallback lorsque aucune entrée spécifique à l'APN ne correspond.
(entrées de liste)	Chaîne (CIDR)	Oui	-	Un préfixe IPv4 (par exemple, `100.64.16.0/24`) ou IPv6 (par exemple, `2001:db8:6f85:70::/64`). La famille d'adresses est détectée à partir de la chaîne. Les adresses UE IPv6 sont tirées de l'intérieur du préfixe configuré et un `/64` est signalé à l'UE.

> Pour la correspondance regex d'APN, plusieurs sous-réseaux par APN, adresses statiques, et l'algorithme d'allocation, voir Allocation d'IP UE.

Un APN avec **à la fois** un CIDR IPv4 et un CIDR IPv6 prend en charge les PDN IPv4, IPv6 et IPv4v6 ; le PGW-C alloue à partir de la ou des familles requises par le type de PDN négocié (pour IPv4v6, une adresse de chaque).

Gx (Framed-IPv6-Prefix)

Aucune configuration supplémentaire n'est requise pour Gx. Le PGW-C définit l'AVP d'adresse encadrée Gx à partir de l'adresse de l'UE par famille :

- **PDN uniquement IPv6** — le préfixe IPv6 de l'UE est transmis en utilisant l'AVP **`Framed-IPv6-Prefix`** (RFC 3162 §2.3), pas `Framed-IP-Address` (qui

ne transporte que l'IPv4).

- **PDNs IPv4 et IPv4v6** — l'adresse IPv4 de l'UE est transmise dans `Framed-IP-Address`. Pour IPv4v6, le préfixe IPv6 **n'est pas** transporté sur Gx (les AVPs d'adresse encadrée de la CCR sont dérivés de l'adresse IPv4) ; le préfixe IPv6 est toujours signalé à l'UPF via PFCP et à l'UE dans le PAA.

Pour un abonné uniquement IPv6, le PCRF doit être provisionné et capable de traiter `Framed-IPv6-Prefix` ; sinon, la demande de contrôle de crédit est abandonnée et la session échoue — voir Dépannage.

Plan utilisateur (OmniUPF)

Pour qu'un PDN IPv6 transporte du trafic, l'OmniUPF doit être activé pour IPv6, programmer la recherche de liaison descendante IPv6 à partir de la session PFCP que le PGW-C envoie, et annoncer chaque route hôte IPv6 `/128` au routeur de périphérie (via OSPFv3) afin que le trafic descendant soit attiré vers l'UPF.

Ce sont des préoccupations de plan utilisateur configurées sur l'OmniUPF, pas sur le PGW-C. Voir le **guide IPv6 / Dual-Stack d'OmniUPF** pour les paramètres du noyau, le pool UE de l'UPF, et l'annonce de route OSPFv3. Du point de vue du PGW-C, la seule exigence est que l'UPF sélectionné prenne en charge IPv6 — le PGW-C signale l'adresse IPv6 de l'UE et le type de PDN via PFCP (voir PFCP/Sxb).

Tables de référence

Allocation d'adresse PDN (PAA) — TS 29.274 §8.14

Type de PDN	Disposition du champ d'adresse
IPv4	4 octets : adresse IPv4
IPv6	1 octet longueur de préfixe + 16 octets préfixe/adresse IPv6
IPv4v6	1 octet longueur de préfixe + 16 octets IPv6 + 4 octets IPv4

AVPs d'adresse encadrée Gx

AVP	Code	Famille	Format	Référence
`Framed-IP-Address`	8	IPv4	adresse de 4 octets	RFC 4005
`Framed-IPv6-Prefix`	97	IPv6	octet réservé + longueur de préfixe + préfixe	RFC 3162 §2.3

IE Type de PDN PFCP — TS 29.244

Valeur	Signification
1	IPv4
2	IPv6
3	IPv4v6

Dépannage

Gx CCR expire pour les sessions IPv6

Symptômes : Les sessions IPv4 pour un abonné réussissent, mais les sessions IPv6/IPv4v6 retournent la cause `remote_peer_not_responding` ; la demande de contrôle de crédit Gx expire.

Causes possibles :

- Le PCRF ne peut pas traiter l'AVP `Framed-IPv6-Prefix` et abandonne silencieusement la demande.
- L'abonné n'est pas provisionné pour IPv6 sur le PCRF/HSS.

Résolution :

1. Confirmer que l'abonné est provisionné et activé sur le PCRF/HSS.
2. Vérifier que le PCRF accepte `Framed-IPv6-Prefix` pour Gx CCR-I.
3. Comparer avec une session IPv4 pour le même abonné afin d'isoler le comportement spécifique à IPv6.

La session IPv6 est acceptée mais le trafic descendant n'atteint pas l'UE

Symptômes : La réponse de création de session contient un PAA IPv6, l'ascendant fonctionne, mais le trafic du réseau vers l'UE est abandonné.

C'est une préoccupation de plan utilisateur (OmniUPF) — le PGW-C a déjà alloué l'adresse et l'a signalée via PFCP. La cause habituelle est que la route hôte `/128` de l'UE n'atteint pas le routeur de périphérie (OSPFv3 ne fonctionne pas/adjacent sur l'UPF, ou délai de convergence). Voir le guide IPv6 / Dual-Stack d'OmniUPF pour la configuration de l'annonce de route et le dépannage.

Aucune adresse IPv6 n'est allouée

Symptômes : Une demande de PDN IPv6 est rejetée, ou aucun PAA IPv6 n'est retourné.

Causes possibles :

- L'APN correspondant (ou `:default`) n'a pas de CIDR IPv6 dans `ue.subnet_map`.
- Le pool IPv6 configuré est épuisé.

Résolution :

1. Ajouter un CIDR IPv6 à l'entrée de l'APN dans `ue.subnet_map` (voir Configuration PGW-C).
2. Confirmer que le pool a des adresses libres.

Demande IPv4v6 lorsque l'APN n'a pas de pool IPv6

Symptômes : Un UE demande un PDN **IPv4v6** contre un APN qui n'a qu'un CIDR IPv4 configuré dans `ue.subnet_map` (pas de plage IPv6). Selon l'APN, la session soit se lève **IPv4 uniquement** soit est **rejetée**.

Comportement : L'allocateur IPv6 signale qu'il n'y a pas de pool IPv6 lorsque l'APN correspondant a une liste IPv6 vide, plutôt que de faire planter le

processus de session. La procédure de session choisit ensuite par type d'APN (selon TS 29.274 / TS 23.401 gestion de porteur à adresse unique / gestion de dégradation de type PDN) :

Type d'APN	Résultat	Ce que l'UE / PCRF / UPF voient
APN d'urgence (SOS)	**Dégrader à IPv4 uniquement** — attribuer l'adresse IPv4, effacer l'adresse IPv6, définir le type de PDN négocié sur `IPv4`.	PAA, Type de PDN PFCP et IP UE PFCP sont uniquement IPv4 ; Gx transporte `Framed-IP-Address`. Le **P-CSCF est toujours retourné dans le PCO** afin que l'enregistrement IMS d'urgence puisse se poursuivre. Un `warning` est enregistré.
APN régulier	**Rejeter** — la session n'est pas silencieusement dégradée ; `{:error, :no_ipv6_pool}` est affiché afin que la mauvaise configuration (ou demande réellement non prise en charge) soit visible. Une `error` est enregistrée.	La création de session échoue avec une cause non réussie.

Un APN d'urgence est identifié par son nom : l'APN contient `"sos"` (par exemple, `sos`, `sos.mnc410.mcc310.gprs`). Les demandes uniquement IPv6 contre un APN sans pool IPv6 sont également rejetées.

> En pratique, le MME dégrade normalement la demande IPv4v6 d'un abonné régulier à son IPv4 souscrit avant que la demande n'atteigne jamais le PGW-C, donc le chemin de rejet est un filet de sécurité pour une demande IPv4v6 qui arrive réellement sans pool IPv6. Le chemin de dégradation SOS

> permet de maintenir les appels d'urgence (par exemple, 911/112) connectés sur un déploiement IPv4 uniquement.

Résolution :

1. Pour un APN régulier qui devrait offrir un dual-stack, ajouter un CIDR IPv6 à l'entrée de l'APN dans `ue.subnet_map` (voir Configuration PGW-C).
2. Pour un APN d'urgence (SOS) sur un déploiement IPv4 uniquement, aucune action n'est requise — la dégradation à IPv4 uniquement est intentionnelle. Ajouter un CIDR IPv6 uniquement si vous souhaitez que les PDNs SOS soient en dual-stack.

Guide de Surveillance et de Métriques d'OmniPGW

Intégration Prometheus et Surveillance Opérationnelle

par Omnitouch Network Services

Table des Matières

1. Aperçu
2. Point de terminaison des métriques
3. Métriques Disponibles
4. Configuration de Prometheus
5. Tableaux de Bord Grafana
6. Alertes
7. Surveillance de la Performance
8. Dépannage des Métriques

Aperçu

OmniPGW fournit deux approches de surveillance complémentaires :

1. API REST OAM en Temps Réel (couverte brièvement ici, détaillée dans les documents d'interface respectifs)

- Visualiseur de session en direct (`GET /api/sessions`)
- Statut des pairs PFCP (`GET /api/upf`)
- Connectivité des pairs Diameter (`GET /api/diameter`)

- Inspection de session individuelle (`GET /api/sessions/<imsi>`)

2. Métriques Prometheus (principal objectif de ce document)

- Tendances historiques et analyses
- Alertes et notifications
- Métriques de performance
- Planification de capacité

Ce document se concentre sur **les métriques Prometheus**. Pour l'API REST OAM (servie sur HTTPS sur le port `8443`, sous le préfixe `/api`), voir :

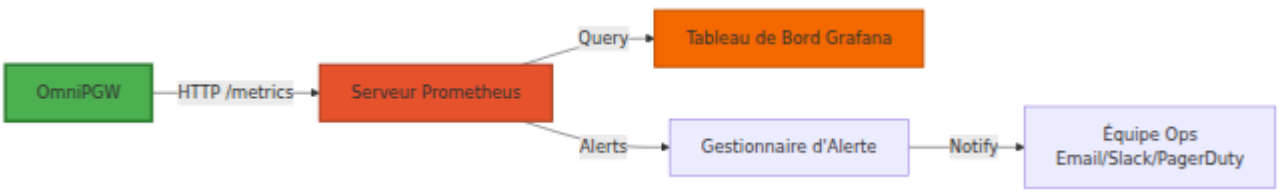
- Gestion des Sessions - API des Sessions
- Interface PFCP - API de Surveillance
- Diameter Gx - API de Surveillance des Pairs

Aperçu des Métriques Prometheus

OmniPGW expose des **métriques compatibles avec Prometheus** pour une surveillance complète de la santé, de la performance et de la capacité du système. Cela permet aux équipes opérationnelles de :

- **Surveiller la Santé du Système** - Suivre les sessions actives, les allocations et les erreurs
- **Planification de Capacité** - Comprendre les tendances d'utilisation des ressources
- **Analyse de Performance** - Mesurer la latence de traitement des messages
- **Alertes** - Notification proactive des problèmes
- **Débogage** - Identifier les causes profondes des problèmes

Architecture de Surveillance



Point de terminaison des métriques

Configuration

Activez les métriques dans `config/runtime.exs` :

```
config :pgw_c,
  metrics: %{
    enabled: true,
    ip_address: "127.0.0.42",       # Adresse de liaison par
défaut (utilisez "0.0.0.0" pour exposer sur toutes les interfaces)
    port: 9090,                      # Port HTTP par défaut
    registry_poll_period_ms: 10_000  # Intervalle de sondage pour
les jauges sondées (par défaut 10s)
  }
```

> **Note :** Les valeurs par défaut expédient le point de terminaison des métriques à `127.0.0.42:9090`. Pour le récupérer depuis un Prometheus distant, définissez `ip_address` sur une adresse accessible (par exemple, `"0.0.0.0"`).

Accéder aux Métriques

Point de terminaison HTTP :

```
http://<omnipgw_ip>:<port>/metrics
```

Exemple :

```
curl http://10.0.0.20:9090/metrics
```

Format de Sortie

Les métriques sont exposées au format **texte Prometheus** :

```
# HELP teid_registry_count Le nombre de TEID enregistrés pour les
sessions
# TYPE teid_registry_count gauge
teid_registry_count 150

# HELP address_registry_count Le nombre d'adresses enregistrées
pour les sessions
# TYPE address_registry_count gauge
address_registry_count 150

# HELP s5s8_inbound_messages_total Le nombre total de messages
reçus des pairs S5/S8
# TYPE s5s8_inbound_messages_total counter
s5s8_inbound_messages_total{message_type="create_session_request"}
1523
s5s8_inbound_messages_total{message_type="delete_session_request"}
1487
```

Métriques Disponibles

OmniPGW expose les catégories de métriques suivantes :

Métriques de Session

Comptes de Sessions Actives :

Nom de la Métrique	Type	Description
`teid_registry_count`	Gauge	Sessions S5/S8 actives (compte TEID)
`seid_registry_count`	Gauge	Sessions PFCP actives (compte SEID)
`gx_session_id_registry_count`	Gauge	Sessions Gx actives (compte Diameter Session-ID)
`gy_session_id_registry_count`	Gauge	Sessions Gy actives (compte Diameter Session-ID)
`session_registry_count`	Gauge	Sessions actives (paires IMSI, EBI)
`address_registry_count`	Gauge	Adresses IP UE allouées
`charging_id_registry_count`	Gauge	IDs de facturation actifs (voir Format CDR de Données pour les enregistrements de facturation CDR)
`sxb_sequence_number_registry_count`	Gauge	Réponses PFCP en attente (attente de réponse)
`s5s8_sequence_number_registry_count`	Gauge	Réponses S5/S8 en attente (attente de réponse)

Nom de la Métrique	Type	Description
`sxb_peer_registry_count`	Gauge	Nombre de processus de pairs PFCP enregistrés

Utilisation :

```
# Sessions actives actuelles
teid_registry_count

# Taux de création de session (par seconde)
rate(teid_registry_count[5m])

# Sessions maximales au cours de la dernière heure
max_over_time(teid_registry_count[1h])
```

Cycle de Vie de Session & de Bearer :

Compteurs incrémentés à mesure que les sessions PDN et les bearers sont créés, modifiés et supprimés.

Nom de la Métrique	Type	Labels	Description
`pgw_session_create_total`	Counter	`result`, `cause`, `pdn_type`, `rat_type`	Créations de sessions PDN, par résultat (succès/échec), cause GTP, type PDN et type RAT
`pgw_session_delete_total`	Counter	`pdn_type`	Suppressions de sessions PDN, par type PDN
`pgw_session_modify_total`	Counter	-	Requêtes de modification de bearer traitées avec succès
`pgw_bearer_create_total`	Counter	`type`, `qci`	Bearers créés, par type (par défaut/dédié) et QCI
`pgw_bearer_delete_total`	Counter	`type`	Bearers supprimés, par type (par d��faut/dédié)

Utilisation :

```
# Taux de succès de la configuration de session (dernières 5m)
sum(rate(pgw_session_create_total{result="success"}[5m])) /
sum(rate(pgw_session_create_total[5m]))

# Échecs de configuration de session par cause
sum(rate(pgw_session_create_total{result="failure"}[5m])) by
(cause)
```

Pool d'Adresses IP UE

Les jauges d'utilisation du pool sont sondées périodiquement ; les échecs d'allocation sont comptés au fur et à mesure qu'ils se produisent. Voir Allocation IP UE.

Nom de la Métrique	Type	Labels	Descrip
`ue_pool_addresses_allocated`	Gauge	`pool`, `ip_version`	Adresse UE actuelle allouées partir d' pool
`ue_pool_addresses_total`	Gauge	`pool`, `ip_version`	Total des adresse UE disponib dans un
`ue_pool_addresses_utilization_ratio`	Gauge	`pool`, `ip_version`	Fraction (0.0–1.0 d'un poo actuelle allouée
`ue_ip_allocation_failures_total`	Counter	`ip_version`, `reason`	Échecs d'allocat IP UE (p exemple épuisem du pool, collision statique

Utilisation :

```
# Pools au-dessus de 80% d'utilisation
ue_pool_addresses_utilization_ratio > 0.8

# Taux d'échec d'allocation par raison
sum(rate(ue_ip_allocation_failures_total[5m])) by (reason)
```

Compteurs de Messages

Messages S5/S8 (GTP-C) :

Nom de la Métrique	Type	Labels	Description
`s5s8_inbound_messages_total`	Counter	`message_type`	Total des messages S5/S8 entrants
`s5s8_outbound_messages_total`	Counter	`message_type`	Total des messages S5/S8 sortants
`s5s8_inbound_errors_total`	Counter	`message_type`	Erreurs de traitement des messages S5/S8 entrants
`s5s8_outbound_errors_total`	Counter	`message_type`	Erreurs de traitement des messages S5/S8 sortants

Types de Messages :

- `create_session_request`
- `create_session_response`
- `delete_session_request`
- `delete_session_response`
- `create_bearer_request`
- `delete_bearer_request`

Messages Sxb (PFCP) :

Nom de la Métrique	Type	Labels	Descript
`sxb_inbound_messages_total`	Counter	`message_type`	Total des messages PFCP entrants
`sxb_outbound_messages_total`	Counter	`message_type`	Total des messages PFCP sortants
`sxb_inbound_errors_total`	Counter	`message_type`	Erreurs d traitemer des messages PFCP entrants
`sxb_outbound_errors_total`	Counter	`message_type`	Erreurs d traitemer des messages PFCP sortants
`sxb_outbound_responses_total`	Counter	`message_type`, `cause_code_class`, `peer_ip`	Réponses PFCP envoyées classées classe de code de cause PF et pair UP

Types de Messages :

- `association_setup_request`
- `association_setup_response`
- `heartbeat_request`
- `heartbeat_response`
- `session_establishment_request`
- `session_establishment_response`
- `session_modification_request`
- `session_deletion_request`

Messages Gx (Diameter) :

Nom de la Métrique	Type	Labels	Descript
`gx_inbound_messages_total`	Counter	`message_type`	Total des messages Diameter entrants
`gx_outbound_messages_total`	Counter	`message_type`	Total des messages Diameter sortants
`gx_inbound_errors_total`	Counter	`message_type`	Erreurs d traitemer des messages Diameter entrants
`gx_outbound_errors_total`	Counter	`message_type`	Erreurs d traitemer des messages Diameter sortants
`gx_outbound_responses_total`	Counter	`message_type`, `result_code_class`, `diameter_host`	Réponses Diameter envoyées classées classe de code de résultat e hôte pair

Types de Messages :

- `gx_CCA` (Credit-Control-Answer)
- `gx_CCR` (Credit-Control-Request)
- `gx_RAA` (Re-Auth-Answer)
- `gx_RAR` (Re-Auth-Request)

Classes de Codes de Résultat (pour `gx_outbound_responses_total`) :

- `2xxx` - Réponses de succès (par exemple, 2001 DIAMETER_SUCCESS)
- `3xxx` - Erreurs de protocole (par exemple, 3001 DIAMETER_COMMAND_UNSUPPORTED)
- `4xxx` - Échecs transitoires (par exemple, 4001 DIAMETER_AUTHENTICATION_REJECTED)
- `5xxx` - Échecs permanents (par exemple, 5012 DIAMETER_UNABLE_TO_COMPLY)

Exemples d'Utilisation :

```
# Surveiller le taux de succès des réponses Gx
sum(rate(gx_outbound_responses_total{result_code_class="2xxx"}[5m]))
sum(rate(gx_outbound_responses_total[5m])) * 100

# Suivre les échecs par hôte PCRF
rate(gx_outbound_responses_total{result_code_class!="2xxx"}[5m]) by (

# Compter le nombre total de messages Re-Auth-Answer réussis
gx_outbound_responses_total{message_type="gx_RAA",result_code_class="

# Alerter sur un taux d'échec élevé vers un PCRF spécifique
rate(gx_outbound_responses_total{result_code_class=~"4xxx|5xxx",diame
[5m]) > 0.1
```

Messages Gy (Diameter) :

Compteurs de messages de facturation en ligne Gy (OCS). Voir Diameter Gy pour l'interface.

Nom de la Métrique	Type	Labels	Descript
`gy_inbound_messages_total`	Counter	`message_type`	Total des messages Gy entran
`gy_outbound_messages_total`	Counter	`message_type`	Total des messages Gy sortan
`gy_inbound_errors_total`	Counter	`message_type`	Erreurs d traitemer des messages Gy entran
`gy_outbound_errors_total`	Counter	`message_type`	Erreurs d traitemer des messages Gy sortan
`gy_inbound_responses_total`	Counter	`cc_request_type`, `result_code_class`, `diameter_host`	Credit-Control-Answers reçus de l'OCS, pa requête à laquelle i répondent classe de code de résultat e hôte OCS
`gy_outbound_responses_total`	Counter	`message_type`, `result_code_class`,	Réponses envoyées

Nom de la Métrique	Type	Labels	Descript
		`diameter_host`	l'OCS (Re Auth-Ans à un RAR OCS), par classe de code de résultat e hôte OCS
`gy_usage_clamped_total`	Counter	-	CCR-Ts do l'Used-Service-U a été limi à zéro pa que l'utilisatio rapportée était supérieur l'utilisatio mesurée. Une valeu non nulle signifie q l'OCS est sous-fact

Types de Messages (`message_type`) :

- `ro_CCR` (Credit-Control-Request)
- `ro_CCA` (Credit-Control-Answer)
- `ro_RAR` (Re-Auth-Request)
- `ro_RAA` (Re-Auth-Answer)

Types de Requêtes CC (`cc_request_type`, pour `gy_inbound_responses_total`) :

- `initial` - réponse à un CCR-I (début de session, première attribution de quota)
- `update` - réponse à un CCR-U (rapport d'utilisation, prochaine attribution de quota)
- `termination` - réponse à un CCR-T (fin de session, utilisation finale)
- `event` - réponse à un CCR basé sur un événement

Classes de Codes de Résultat (pour `gy_inbound_responses_total` et `gy_outbound_responses_total`) : les mêmes classes que pour Gx (`2xxx`, `3xxx`, `4xxx`, `5xxx`, `other`). Sur Gy, `4012` DIAMETER_CREDIT_LIMIT_REACHED est un `4xxx` et `5030` DIAMETER_USER_UNKNOWN est un `5xxx`.

La `result_code_class` est le code de résultat de niveau supérieur du CCA. Un OCS peut répondre `2001` au niveau supérieur et toujours refuser ou rediriger un service unique dans son Multiple-Services-Credit-Control.

Exemples d'Utilisation :

```
# Taux de succès des CCA par type de requête
sum by (cc_request_type) (rate(gy_inbound_responses_total{result_code
[5m]))
  / sum by (cc_request_type) (rate(gy_inbound_responses_total[5m])) *

# Échecs de CCR-Is (nouvelles sessions que l'OCS a refusées), par hôt
sum by (diameter_host)
(rate(gy_inbound_responses_total{cc_request_type="initial",result_coc
[5m]))

# CCRs envoyés sans CCA de retour (timeouts OCS ou réponses perdues),
sum by (instance) (rate(gy_outbound_messages_total{message_type="ro_C
  - sum by (instance) (rate(gy_inbound_responses_total[5m]))

# Re-Auth-Answers envoyés à l'OCS qui n'ont pas été réussis
rate(gy_outbound_responses_total{message_type="ro_RAA",result_code_cl
[5m])
```

Messages Gn/Gp (GTPv1-C) :

Compteurs de messages de contrôle Gn/Gp GGSN (interopérabilité SGSN 2G/3G). Voir Interface Gn.

Nom de la Métrique	Type	Labels	Description
`gn_inbound_messages_total`	Counter	`message_type`	Total des messages Gn/Gp entrants (provenant de SGSN)
`gn_outbound_messages_total`	Counter	`message_type`	Total des messages Gn/Gp sortants (vers SGSN)
`gn_inbound_errors_total`	Counter	`message_type`	Erreurs de traitement des messages Gn/Gp entrants
`gn_outbound_errors_total`	Counter	`message_type`	Réponses d'erreur Gn/Gp envoyées au SGSN

Gestion des Erreurs :

Nom de la Métrique	Type	Labels	Description
`rescues_total`	Counter	`module`, `function`	Total des blocs de secours atteints (gestion des exceptions)

Métriques de Latence

Durée de Traitement des Messages Entrants :

Nom de la Métrique	Type	Labels	D
`s5s8_inbound_handling_duration`	Histogram	`request_message_type`	T t c r S c e e r
`gn_inbound_handling_duration`	Histogram	`request_message_type`	T t c r C ((c e e r
`sxb_inbound_handling_duration`	Histogram	`request_message_type`	T t c r P c e e r
`gx_inbound_handling_duration`	Histogram	`request_message_type`	T t c

Nom de la Métrique	Type	Labels	D
			r C ((c e e r
`gy_inbound_handling_duration`	Histogram	`request_message_type`	T t c r C ((c e e r

Durée des Transactions Sortantes :

Nom de la Métrique	Type	Labels
`s5s8_outbound_transaction_duration`	Histogram	`request_message_type`
`gn_outbound_transaction_duration`	Histogram	`request_message_type`
`sxb_outbound_transaction_duration`	Histogram	`request_message_type`
`gx_outbound_transaction_duration`	Histogram	`request_message_type`
`gy_outbound_transaction_duration`	Histogram	`request_message_type`

Seaux (secondes) :

- Valeurs : 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0
- (100µs, 500µs, 1ms, 5ms, 10ms, 50ms, 100ms, 500ms, 1s, 5s)

Utilisation :

```
# Latence S5/S8 au 95e percentile
histogram_quantile(0.95,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)

# Latence moyenne PFCP
rate(sxb_inbound_handling_duration_sum[5m]) /
rate(sxb_inbound_handling_duration_count[5m])
```

Surveillance de la Santé du UPF

Métriques des Pairs UPF :

Nom de la Métrique	Type	Labels	Description
`upf_peers_total`	Gauge	-	Nombre total de pairs UPF enregistrés
`upf_peers_healthy`	Gauge	-	Nombre de pairs UPF sains (associés + cœurs OK)
`upf_peers_unhealthy`	Gauge	-	Nombre de pairs UPF non sains
`upf_peers_associated`	Gauge	-	Nombre de pairs UPF avec association PFCP active
`upf_peers_unassociated`	Gauge	-	Nombre de pairs UPF sans association PFCP
`upf_peer_associated`	Gauge	`peer_ip`	Indique si un UPF spécifique a une association PFCP active (1/0)
`upf_peer_healthy`	Gauge	`peer_ip`	État de santé d'un UPF spécifique (1=sain, 0=non sain)
`upf_peer_missed_heartbeats`	Gauge	`peer_ip`	Cœurs consécutifs

Nom de la Métrique	Type	Labels	Description
			manqués pour un UPF spécifique
`upf_heartbeat_rtt`	Histogram	`peer_ip`	Temps aller-retour de la requête/réponse de cœur PFCP (secondes)

Utilisation :

```
# Surveiller la santé du pool UPF
upf_peers_healthy / upf_peers_total

# Alerter sur les UPF non sains
upf_peers_unhealthy > 0

# Suivre la santé d'un UPF spécifique
upf_peer_healthy{peer_ip="10.98.0.20"}

# Identifier les UPF avec des problèmes de cœur
upf_peer_missed_heartbeats > 2
```

Exemples d'Alerte :

```yaml
# Alerte lorsque le UPF tombe
- alert: UPF_Peer_Down
  expr: upf_peer_healthy == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "Le UPF {{ $labels.peer_ip }} est hors ligne"
    description: "Le pair UPF ne répond pas aux cœurs PFCP"

# Alerte lorsque plusieurs UPF sont hors ligne
- alert: UPF_Pool_Degraded
  expr: (upf_peers_healthy / upf_peers_total) < 0.5
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "Pool UPF dégradé"
    description: "Seulement {{ $value | humanizePercentage }} des
UPF sont sains"

# Avertissement sur les cœurs manqués
- alert: UPF_Heartbeat_Issues
  expr: upf_peer_missed_heartbeats > 2
  for: 30s
  labels:
    severity: warning
  annotations:
    summary: "UPF {{ $labels.peer_ip }} problèmes de cœur"
    description: "{{ $value }} cœurs consécutifs manqués"
```

Surveillance de la Santé du P-CSCF

Métriques du Serveur P-CSCF :

Nom de la Métrique	Type	Labels	Description
`pcscf_fqdns_total`	Gauge	-	Total des FQDN P-CSCF surveillés
`pcscf_fqdns_resolved`	Gauge	-	FQDN P-CSCF résolus avec succès via DNS
`pcscf_fqdns_failed`	Gauge	-	FQDN P-CSCF ayant échoué à la résolution DNS
`pcscf_servers_total`	Gauge	-	Total des serveurs P-CSCF découverts
`pcscf_servers_healthy`	Gauge	`fqdn`	Serveurs P-CSCF sains par FQDN
`pcscf_servers_unhealthy`	Gauge	`fqdn`	Serveurs P-CSCF non sains par FQDN

Voir : Guide de Surveillance P-CSCF pour un suivi détaillé de la santé IMS.

Connectivité des Pairs Diameter

Sondé périodiquement depuis la pile Diameter.

Nom de la Métrique	Type	Labels	Description
`diameter_peers_connected`	Gauge	`application`	Nombre de pairs Diameter actuellement connectés, par application (`gx`/`gy`/`all`)

Utilisation :

```
# Alerte lorsque aucun pair Gx (PCRF) n'est connecté
diameter_peers_connected{application="gx"} == 0
```

Métriques de Licence

État de la Licence :

Nom de la Métrique	Type	Description
`license_status`	Gauge	État actuel de la licence (1 = valide, 0 = invalide)

Utilisation :

```
# Vérifier si la licence est valide
license_status == 1

# Alerte sur une licence invalide
license_status == 0
```

Exemple d'Alerte :

```
- alert: PGW_C_License_Invalid
  expr: license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "Licence PGW-C invalide ou expirée"
    description: "L'état de la licence est invalide - les demandes
de création de session sont bloquées"
```

Impact d'une licence invalide :

Lorsque la licence est invalide ou que le serveur de licence est inaccessible, **les demandes de création de session seront rejetées** avec le code de cause GTP-C **"Aucune ressource disponible" (73)**. Cela est visible dans les captures de paquets comme indiqué ci-dessous :

Capture Wireshark montrant la réponse de création de session avec le code de cause "Aucune ressource disponible" lorsque la licence est invalide

Remarques :

- Nom du produit enregistré avec le serveur de licence : `omnipgwc`
- L'URL du serveur de licence est configurée dans `config/runtime.exs` sous `:license_client`
- Lorsque la licence est invalide (`license_status == 0`), les demandes de création de session sont bloquées avec le code de cause GTP-C 73 (Aucune ressource disponible)
- L'interface utilisateur et la surveillance restent accessibles indépendamment de l'état de la licence
- Les pairs Diameter, GTP-C et PFCP continuent de maintenir des connexions
- Les sessions existantes ne sont pas affectées - seule la création de nouvelles sessions est bloquée

Métriques du système

Métriques de la VM Erlang :

Nom de la métrique	Type	Description
`vm_memory_total`	Gauge	Mémoire totale de la VM (octets)
`vm_memory_processes`	Gauge	Mémoire utilisée par les processus
`vm_memory_system`	Gauge	Mémoire utilisée par le système
`vm_system_process_count`	Gauge	Total des processus Erlang
`vm_system_port_count`	Gauge	Total des ports ouverts

Configuration de Prometheus

Configuration de Scrape

Ajoutez OmniPGW au `prometheus.yml` de Prometheus :

```yaml
# prometheus.yml
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'omnipgw'
    static_configs:
      - targets: ['10.0.0.20:9090']
        labels:
          instance: 'omnipgw-01'
          environment: 'production'
          site: 'datacenter-1'
```

Instances multiples d'OmniPGW

```yaml
scrape_configs:
  - job_name: 'omnipgw'
    static_configs:
      - targets:
          - '10.0.0.20:9090'
          - '10.0.0.21:9090'
          - '10.0.0.22:9090'
        labels:
          environment: 'production'
```

Découverte de services

Kubernetes :

```yaml
scrape_configs:
  - job_name: 'omnipgw'
    kubernetes_sd_configs:
      - role: pod
    relabel_configs:
      - source_labels: [__meta_kubernetes_pod_label_app]
        action: keep
        regex: omnipgw
      - source_labels: [__meta_kubernetes_pod_ip]
        target_label: __address__
        replacement: '${1}:9090'
```

Vérification

Test de scrape :

```
# Vérifiez les cibles de Prometheus
curl http://prometheus:9090/api/v1/targets

# Interrogez une métrique
curl 'http://prometheus:9090/api/v1/query?
query=teid_registry_count'
```

Tableaux de bord Grafana

Configuration du tableau de bord

1. Ajouter la source de données Prometheus :

```
Configuration → Sources de données → Ajouter une source de données
→ Prometheus
URL : http://prometheus:9090
```

2. Importer le tableau de bord :

Créez un nouveau tableau de bord ou importez à partir de JSON.

Panneaux clés

Panneau 1 : Sessions actives

```
# Requête
teid_registry_count

# Type de panneau : Gauge
# Seuils :
#   Vert : < 5000
#   Jaune : 5000-8000
#   Rouge : > 8000
```

Panneau 2 : Taux de session

```
# Requête
rate(s5s8_inbound_messages_total{message_type="create_session_request
[5m])

# Type de panneau : Graphique
# Unité : demandes/sec
```

Panneau 3 : Utilisation du pool IP

```
# Requête (pour un sous-réseau /24 avec 254 IPs)
(address_registry_count / 254) * 100

# Type de panneau : Gauge
# Unité : pourcentage (0-100)
# Seuils :
#   Vert : < 70%
#   Jaune : 70-85%
#   Rouge : > 85%
```

Panneau 4 : Latence des messages (95e percentile)

```
# Requête
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea
[5m])
)

# Type de panneau : Graphique
# Unité : millisecondes
```

Panneau 5 : Taux d'erreur

```
# Requête
rate(s5s8_inbound_errors_total[5m])

# Type de panneau : Graphique
# Unité : erreurs/sec
# Seuil d'alerte : > 0.1
```

Panneau 6 : Taux de succès des réponses Gx

```
# Requête : Calculer le pourcentage de réponses Gx réussies
sum(rate(gx_outbound_responses_total{result_code_class="2xxx"}
[5m])) /
sum(rate(gx_outbound_responses_total[5m])) * 100

# Type de panneau : Gauge
# Unité : pourcentage (0-100)
# Seuils :
#   Vert : > 95%
#   Jaune : 90-95%
#   Rouge : < 90%
```

Alternative - Répartition par classe de code de résultat :

```
# Requête : Montrer les comptes de réponses par classe de code de
résultat
sum(rate(gx_outbound_responses_total[5m])) by (result_code_class)

# Type de panneau : Graphique en secteurs ou graphique à barres
# Légende : {{ result_code_class }}
```

Alternative - Par statut de réponse PCRF :

```
# Requête : Montrer les réponses par hôte PCRF
sum(rate(gx_outbound_responses_total[5m])) by (diameter_host,
result_code_class)

# Type de panneau : Graphique à barres empilées
# Légende : {{ diameter_host }} - {{ result_code_class }}
```

Panneau 7 : État de santé du UPF

```
# Requête : Pourcentage global de santé du pool
(upf_peers_healthy / upf_peers_total) * 100

# Type de panneau : Gauge
# Unité : pourcentage (0-100)
# Seuils :
#   Vert : 100%
#   Jaune : 50-99%
#   Rouge : < 50%
```

Alternative - Par statut de chaque UPF :

```
# Requête : Santé individuelle du UPF
upf_peer_healthy

# Type de panneau : Stat
# Mappings :
#   1 = "UP" (Vert)
#   0 = "DOWN" (Rouge)
```

Exemple complet de tableau de bord

```
{
  "dashboard": {
    "title": "OmniPGW - Tableau de bord des opérations",
    "panels": [
      {
        "title": "Sessions actives",
        "targets": [
          {
            "expr": "teid_registry_count",
            "legendFormat": "Sessions actives"
          }
        ],
        "type": "graph"
      },
      {
        "title": "Taux de création de session",
        "targets": [
          {
            "expr":
"rate(s5s8_inbound_messages_total{message_type=\"create_session_reque
[5m])",
            "legendFormat": "Sessions/sec"
          }
        ],
        "type": "graph"
      },
      {
        "title": "Utilisation du pool IP",
        "targets": [
          {
            "expr": "(address_registry_count / 254) * 100",
            "legendFormat": "Utilisation du pool %"
          }
        ],
        "type": "gauge"
      },
      {
        "title": "Latence des messages (p95)",
        "targets": [
          {
            "expr": "histogram_quantile(0.95,
```

```
rate(s5s8_inbound_handling_duration_bucket[5m]))",
            "legendFormat": "S5/S8 p95"
          },
          {
            "expr": "histogram_quantile(0.95,
rate(sxb_inbound_handling_duration_bucket[5m]))",
            "legendFormat": "PFCP p95"
          }
        ],
        "type": "graph"
      }
    ]
  }
}
```

Alertes

Règles d'alerte

Créez `omnipgw_alerts.yml` :

```yaml
groups:
  - name: omnipgw
    interval: 30s
    rules:
      # Alertes de compte de session
      - alert: OmniPGW_HighSessionCount
        expr: teid_registry_count > 8000
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Nombre de sessions élevé dans OmniPGW"
          description: "{{ $value }} sessions actives (seuil :
8000)"

      - alert: OmniPGW_SessionCountCritical
        expr: teid_registry_count > 9500
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: "Nombre de sessions critique dans OmniPGW"
          description: "{{ $value }} sessions actives approchant
la capacité"

      # Alertes de pool IP
      - alert: OmniPGW_IPPoolUtilizationHigh
        expr: (address_registry_count / 254) * 100 > 80
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "Utilisation élevée du pool IP dans OmniPGW"
          description: "Pool IP {{ $value }}% utilisé"

      - alert: OmniPGW_IPPoolExhausted
        expr: address_registry_count >= 254
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "Pool IP épuisé dans OmniPGW"
          description: "Aucune IP disponible pour allocation"
```

```
      # Alertes de taux d'erreur
      - alert: OmniPGW_HighErrorRate
        expr: rate(s5s8_inbound_errors_total[5m]) > 0.1
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Taux d'erreur élevé dans OmniPGW"
          description: "{{ $value }} erreurs/sec sur l'interface
S5/S8"

      - alert: OmniPGW_GxErrorRate
        expr: rate(gx_inbound_errors_total[5m]) > 0.05
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Erreurs Gx dans OmniPGW"
          description: "{{ $value }} erreurs Diameter/sec"

      # Alertes de réponse Gx
      - alert: OmniPGW_GxResponseFailureRate
        expr: |

sum(rate(gx_outbound_responses_total{result_code_class!="2xxx"}
[5m])) /
          sum(rate(gx_outbound_responses_total[5m])) > 0.1
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Taux d'échec élevé des réponses Gx dans
OmniPGW"
          description: "{{ $value | humanizePercentage }} des
réponses Gx sont des échecs (codes de résultat non-2xxx)"

      - alert: OmniPGW_GxPCRFFailures
        expr:
rate(gx_outbound_responses_total{result_code_class=~"4xxx|5xxx"}
[5m]) by (diameter_host) > 0.05
        for: 3m
        labels:
          severity: warning
```

```
        annotations:
          summary: "PCRF {{ $labels.diameter_host }} recevant des
réponses d'échec"
          description: "{{ $value }} réponses d'échec/sec au PCRF
{{ $labels.diameter_host }}"

      # Alertes de santé UPF
      - alert: OmniPGW_UPF_PeerDown
        expr: upf_peer_healthy == 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "Pair UPF {{ $labels.peer_ip }} hors ligne"
          description: "UPF ne répond pas aux cœurs PFCP"

      - alert: OmniPGW_UPF_PoolDegraded
        expr: (upf_peers_healthy / upf_peers_total) < 0.5
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: "Pool UPF dégradé"
          description: "{{ $value | humanizePercentage }} des UPF
sont sains (< 50%)"

      - alert: OmniPGW_UPF_HeartbeatFailures
        expr: upf_peer_missed_heartbeats > 2
        for: 30s
        labels:
          severity: warning
        annotations:
          summary: "Échecs de cœur UPF {{ $labels.peer_ip }}"
          description: "{{ $value }} cœurs manqués consécutifs"

      - alert: OmniPGW_UPF_AllDown
        expr: upf_peers_healthy == 0 and upf_peers_total > 0
        for: 30s
        labels:
          severity: critical
        annotations:
          summary: "Tous les pairs UPF hors ligne"
          description: "Aucun UPF sain disponible pour la création
de session"
```

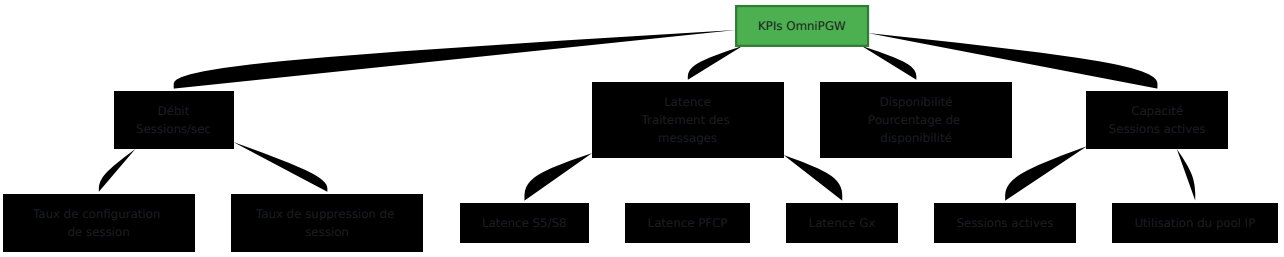
```
      # Alertes de latence
      - alert: OmniPGW_HighLatency
        expr: |
          histogram_quantile(0.95,
            rate(s5s8_inbound_handling_duration_bucket[5m])
          ) > 100000
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Latence élevée des messages dans OmniPGW"
          description: "Latence p95 {{ $value }}µs (> 100ms)"

      # Alertes système
      - alert: OmniPGW_HighMemoryUsage
        expr: vm_memory_total > 2000000000
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "Utilisation élevée de la mémoire dans OmniPGW"
          description: "VM utilisant {{ $value | humanize }}B de
mémoire"

      - alert: OmniPGW_HighProcessCount
        expr: vm_system_process_count > 100000
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "Nombre élevé de processus dans OmniPGW"
          description: "{{ $value }} processus Erlang (fuite
potentielle)"
```

Configuration d'AlertManager

```yaml
# alertmanager.yml
global:
  resolve_timeout: 5m

route:
  receiver: 'ops-team'
  group_by: ['alertname', 'instance']
  group_wait: 10s
  group_interval: 10s
  repeat_interval: 12h

  routes:
    - match:
        severity: critical
      receiver: 'pagerduty'

    - match:
        severity: warning
      receiver: 'slack'

receivers:
  - name: 'ops-team'
    email_configs:
      - to: 'ops@example.com'

  - name: 'slack'
    slack_configs:
      - api_url:
'https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK'
        channel: '#omnipgw-alerts'
        title: 'Alerte OmniPGW : {{ .GroupLabels.alertname }}'
        text: '{{ range .Alerts }}{{ .Annotations.description }}{{
end }}'

  - name: 'pagerduty'
    pagerduty_configs:
      - service_key: 'YOUR_PAGERDUTY_KEY'
```

Surveillance des performances

Indicateurs clés de performance (KPI)



Requêtes de débit

Taux de configuration de session :

```
rate(s5s8_inbound_messages_total{message_type="create_session_request
[5m])
```

Taux de suppression de session :

```
rate(s5s8_inbound_messages_total{message_type="delete_session_request
[5m])
```

Croissance nette des sessions :

```
rate(s5s8_inbound_messages_total{message_type="create_session_request
[5m]) -
rate(s5s8_inbound_messages_total{message_type="delete_session_request
[5m])
```

Analyse de la latence

Latence de traitement des messages (percentiles) :

```
# p50 (Médiane)
histogram_quantile(0.50,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)

# p95
histogram_quantile(0.95,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)

# p99
histogram_quantile(0.99,
  rate(s5s8_inbound_handling_duration_bucket[5m])
)
```

Répartition de la latence par type de message :

```
histogram_quantile(0.95,
  rate(s5s8_inbound_handling_duration_bucket[5m])
) by (request_message_type)
```

Tendance de capacité

Tendance de croissance des sessions (24h) :

```
teid_registry_count -
teid_registry_count offset 24h
```

Capacité restante :

```
# Pour une capacité maximale de 10 000 sessions
10000 - teid_registry_count
```

Temps jusqu'à l'épuisement de la capacité :

```
# Jours jusqu'à épuisement de la capacité (basé sur le taux de
croissance de 1h)
(10000 - teid_registry_count) /
(rate(teid_registry_count[1h]) * 86400)
```

Dépannage des métriques

Identification des problèmes

Problème : Taux de rejet de session élevé

Requête :

```
rate(s5s8_inbound_errors_total[5m]) by (message_type)
```

Action :

- Vérifiez les journaux d'erreurs
- Vérifiez la connectivité PCRF (erreurs Gx)
- Vérifiez l'épuisement du pool IP

Problème : Configuration de session lente

Requête :

```
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea
[5m])
)
```

Action :

- Vérifiez la latence Gx (temps de réponse PCRF)
- Vérifiez la latence PFCP (temps de réponse PGW-U)

- Examinez l'utilisation des ressources système

Problème : Échecs de politique PCRF

Requêtes :

```
# Taux d'échec global des réponses Gx
sum(rate(gx_outbound_responses_total{result_code_class!="2xxx"}
[5m])) /
sum(rate(gx_outbound_responses_total[5m])) * 100

# Répartition par hôte PCRF
sum(rate(gx_outbound_responses_total[5m])) by (diameter_host,
result_code_class)

# Classes de codes de résultat spécifiques
rate(gx_outbound_responses_total{result_code_class="5xxx"}[5m]) by
(diameter_host)
```

Action :

- Vérifiez la connectivité et la santé du PCRF
- Examinez les profils d'abonnés dans le PCRF (les erreurs 5xxx indiquent souvent des problèmes de politique)
- Vérifiez la configuration des pairs Diameter
- Vérifiez les journaux du PCRF pour les erreurs correspondantes
- Pour 5012 (DIAMETER_UNABLE_TO_COMPLY), examinez le traitement de la demande de ré-authentification

Problème : Fuite de mémoire suspectée

Requêtes :

```
# Tendance de la mémoire totale
rate(vm_memory_total[1h])

# Tendance de la mémoire des processus
rate(vm_memory_processes[1h])

# Tendance du nombre de processus
rate(vm_system_process_count[1h])
```

Action :

- Vérifiez les sessions obsolètes
- Examinez les comptes de registre
- Redémarrez si la fuite est confirmée

Requêtes de débogage

Trouver le moment de pic des sessions :

```
max_over_time(teid_registry_count[24h])
```

Comparer actuel vs historique :

```
teid_registry_count /
avg_over_time(teid_registry_count[7d])
```

Identifier les anomalies :

```
abs(
  teid_registry_count -
  avg_over_time(teid_registry_count[1h])
) > 100
```

Meilleures pratiques

Collecte de métriques

1. **Intervalle de scrape :** 15-30 secondes (équilibrer granularité vs charge)
2. **Rétention :** 15+ jours pour analyse historique
3. **Étiquettes :** Utilisez un étiquetage cohérent (instance, environnement, site)

Conception de tableau de bord

1. **Tableau de bord de vue d'ensemble** - KPI de haut niveau pour le NOC
2. **Tableaux de bord détaillés** - Analyse approfondie par interface
3. **Tableau de bord de dépannage** - Métriques d'erreur et journaux

Conception d'alerte

1. **Éviter la fatigue d'alerte** - Alerter uniquement sur des problèmes exploitables
2. **Escalade** - Avertissement → Critique avec sévérité croissante
3. **Contexte** - Inclure des liens vers des runbooks dans les descriptions d'alerte

---

Documentation connexe

Configuration et installation

- **Guide de configuration** - Configuration des métriques Prometheus, configuration de l'API REST OAM
- **Guide de dépannage** - Utilisation des métriques pour le débogage

Métriques d'interface

- **Interface PFCP** - Métriques de session PFCP, surveillance de la santé UPF
- **Interface Diameter Gx** - Métriques de politique Gx, suivi de l'interaction PCRF
- **Interface Diameter Gy** - Métriques de facturation Gy, suivi des quotas, délais OCS
- **Interface S5/S8** - Métriques de message GTP-C, communication SGW-C

Surveillance spécialisée

- **Surveillance P-CSCF** - Métriques de découverte P-CSCF, santé IMS
- **Gestion des sessions** - Sessions actives, métriques du cycle de vie des sessions
- **Attribution IP UE** - Métriques d'utilisation du pool IP

Retour au guide des opérations

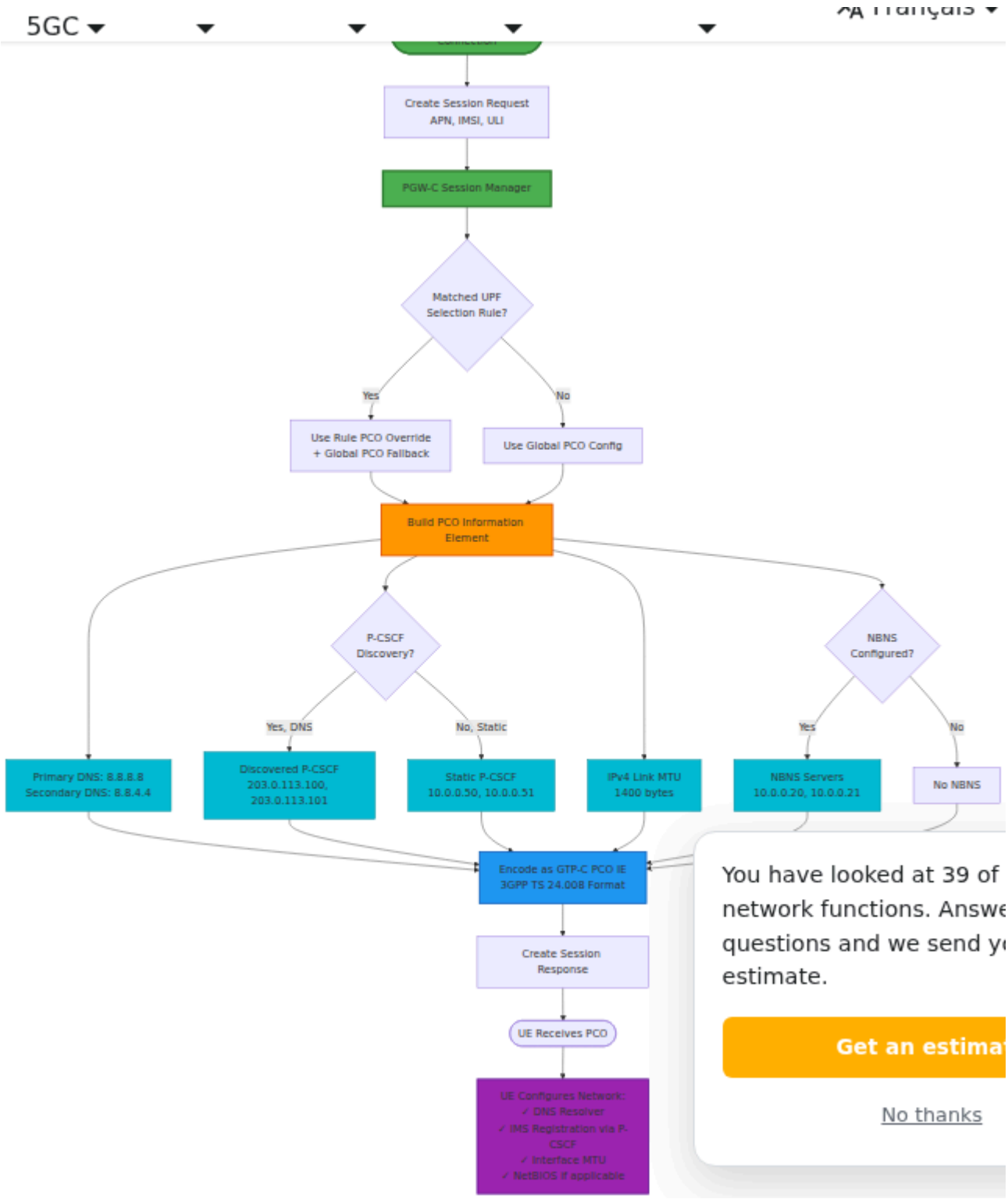
Options de Configuration de Protocole (PCO)

Paramètres Réseau Fournis au UE

OmniPGW par Omnitouch Network Services

Vue d'ensemble

PCO (Options de Configuration de Protocole) sont des paramètres réseau envoyés au UE (dispositif mobile) lors de l'établissement de la connexion PDN. Ces paramètres permettent au UE d'accéder à des services réseau tels que DNS, IMS, et de configurer les paramètres réseau.



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

Conteneurs PCO émis par OmniPGW (réseau → MS) :

Nom IE	ID de Conteneur	Description	Requis
Adresse IPv4 du Serveur DNS	0x000D	DNS Principal	Oui
Adresse IPv4 du Serveur DNS	0x000D	DNS Secondaire	Optionnel
Adresse IPv6 du Serveur DNS	0x0003	DNS Principal/Secondaire (IPv6)	Optionnel
Adresse IPv4 du P-CSCF	0x000C	P-CSCF pour IMS (un conteneur par adresse)	Optionnel (IMS)
Adresse IPv6 du P-CSCF	0x0001	P-CSCF pour IMS (un conteneur par adresse)	Optionnel (IMS)
MTU de Lien IPv4	0x0010	Unité de transmission maximale	Recommandé

> **DNS/NBNS via IPCP.** Lorsque le UE demande des serveurs DNS (et NBNS/WINS) en utilisant un **IPCP** (Internet Protocol Control Protocol, conteneur `0x8021`) Configure-Request plutôt qu'un conteneur DNS nu, OmniPGW répond à l'intérieur d'un IPCP **Configure-Nak** portant les adresses configurées comme options IPCP `ms_dns1`/`ms_dns2` et `ms_wins1`/`ms_wins2` (RFC 1877). NBNS/WINS est donc livré **uniquement** par ce chemin IPCP — il n'y a pas de conteneur PCO NBNS autonome (`0x0011`) émis. Un Configure-Nak (pas un Ack) est utilisé parce que le PGW fournit des adresses qui diffèrent des espaces réservés `0.0.0.0` du UE, et la réponse fait écho à l'Identifiant IPCP de la demande (RFC 1661 §5).

> **P-CSCF IPv4 et IPv6 / DNS ; pas de E-CSCF.** OmniPGW répond aux demandes d'adresses P-CSCF IPv4 et IPv6, acheminant chaque adresse

> découverte ou configurée vers le conteneur correspondant à sa famille : les adresses IPv4 vont dans `0x000C`, les adresses IPv6 vont dans `0x0001`. Une demande P-CSCF IPv6 (`p_cscf_ipv6_address_request`) est servie à partir de `p_cscf_ipv6_address_list` (ou les adresses IPv6 retournées par DNS/AAAA découverte), et une demande DNS IPv6 (`dns_server_ipv6_address_request`) est servie dans le conteneur `0x0003` à partir de `primary_dns_server_ipv6_address` / `secondary_dns_server_ipv6_address`. Les familles d'adresses ne sont jamais mélangées entre les conteneurs. Il n'y a **pas** de gestion de conteneur E-CSCF. Tout ce qui précède s'applique également à APCO (PCO Additionnel).

Configuration

Configuration de Base

```
# config/runtime.exs
config :pgw_c,
  pco: %{
    # Serveurs DNS (requis)
    primary_dns_server_address: "8.8.8.8",
    secondary_dns_server_address: "8.8.4.4",

    # Serveurs DNS IPv6 (optionnels, conteneur 0x0003)
    primary_dns_server_ipv6_address: nil,
    secondary_dns_server_ipv6_address: nil,

    # Serveurs NBNS (optionnels, pour les dispositifs Windows)
    primary_nbns_server_address: nil,
    secondary_nbns_server_address: nil,

    # Adresses P-CSCF pour IMS/VoLTE (optionnelles)
    p_cscf_ipv4_address_list: [],   # conteneur 0x000C
    p_cscf_ipv6_address_list: [],   # conteneur 0x0001

    # Découverte Dynamique P-CSCF (optionnelle ; résout les
enregistrements A et AAAA)
    p_cscf_discovery_enabled: false,
    p_cscf_discovery_dns_server: nil,
    p_cscf_discovery_timeout_ms: 5000,

    # Taille MTU IPv4 (octets)
    ipv4_link_mtu_size: 1400
  }
```

Paramètres PCO

Adresses des Serveurs DNS

DNS Principal et Secondaire :

```
pco: %{
  primary_dns_server_address: "8.8.8.8",
  secondary_dns_server_address: "8.8.4.4"
}
```

Fournisseurs DNS Communs :

Fournisseur	Principal	Secondaire
Google	8.8.8.8	8.8.4.4
Cloudflare	1.1.1.1	1.0.0.1
Quad9	9.9.9.9	149.112.112.112
OpenDNS	208.67.222.222	208.67.220.220

DNS Privé :

```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11"
}
```

Adresses P-CSCF (IMS)

Pour les Services IMS/VoLTE :

```
pco: %{
  p_cscf_ipv4_address_list: [
    "10.0.0.50",  # P-CSCF Principal (IPv4, conteneur 0x000C)
    "10.0.0.51"   # P-CSCF Secondaire
  ],
  # Adresses P-CSCF IPv6 (conteneur 0x0001) pour les PDN IMS
IPv6/IPv4v6
  p_cscf_ipv6_address_list: [
    "2001:db8::50",
    "2001:db8::51"
  ]
}
```

Chaque adresse est émise uniquement dans le conteneur correspondant à sa famille, et uniquement lorsque le UE demande cette famille — une demande P-CSCF IPv4 reçoit les conteneurs `0x000C`, une demande P-CSCF IPv6 reçoit les conteneurs `0x0001`. Les PDN IMS à double pile demandent généralement les deux.

P-CSCF (Fonction de Contrôle de Session d'Appel Proxy) :

- Point d'entrée pour la signalisation IMS
- Requis pour VoLTE, VoWiFi, RCS
- UE utilise SIP via ce serveur

Découverte Dynamique P-CSCF

Découverte P-CSCF Basée sur DNS :

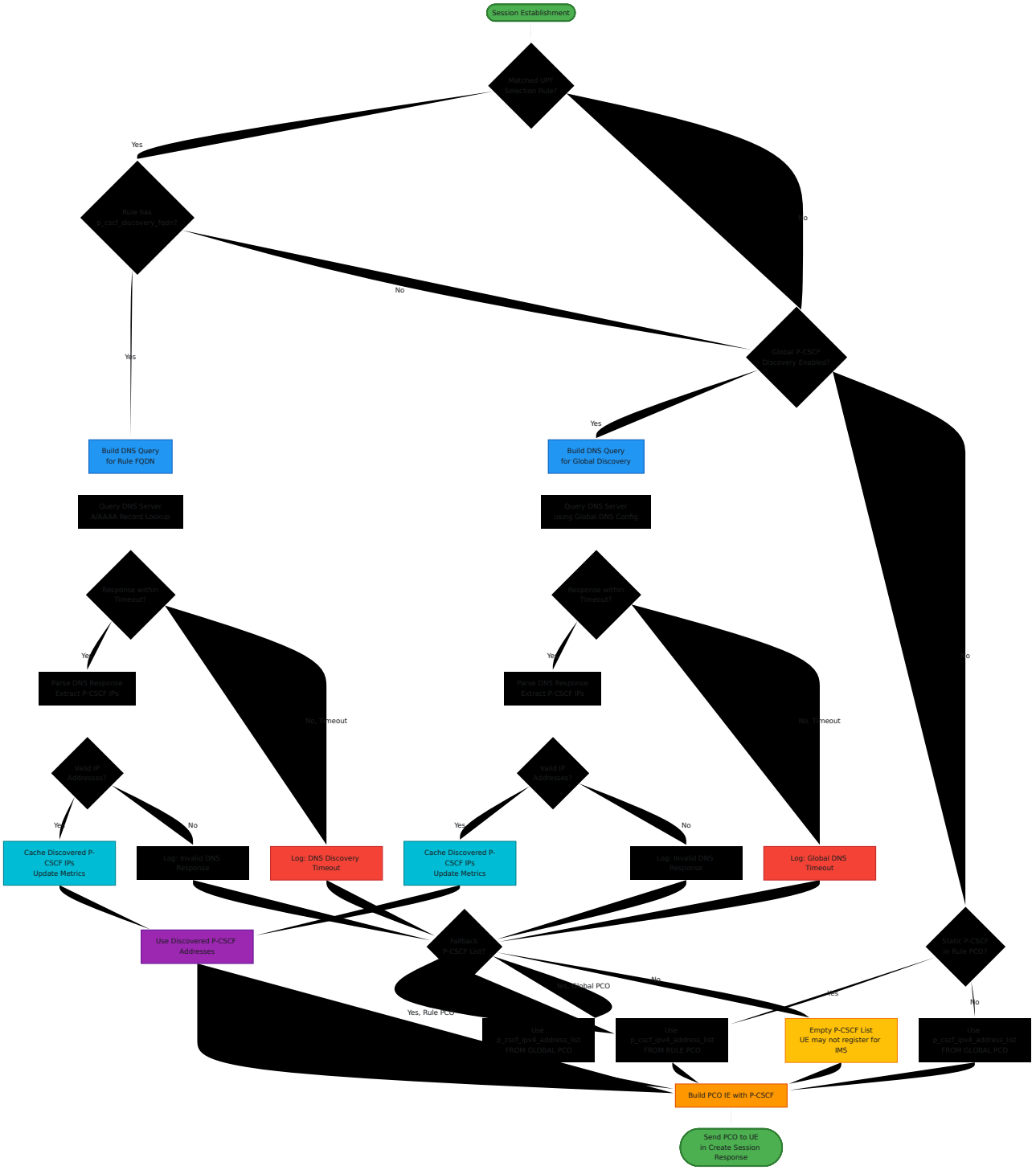
OmniPGW prend en charge la découverte dynamique de P-CSCF via des requêtes DNS comme défini dans 3GPP TS 23.003 et TS 24.229. Lorsqu'elle est activée, PGW-C peut interroger DNS pour les adresses P-CSCF au lieu d'utiliser une configuration statique.

```
pco: %{
  # Activer la découverte dynamique P-CSCF
  p_cscf_discovery_enabled: true,

  # Serveur DNS pour les requêtes P-CSCF (sous forme de tuple)
  p_cscf_discovery_dns_server: {10, 179, 2, 177},

  # Délai d'attente pour les requêtes DNS (millisecondes)
  p_cscf_discovery_timeout_ms: 5000,

  # Liste statique de P-CSCF (utilisée comme secours si DNS
échoue)
  p_cscf_ipv4_address_list: ["10.0.0.50"]
}
```

Comment ça fonctionne :

1. Lorsque `p_cscf_discovery_enabled: true`, PGW-C effectue des requêtes DNS pour les adresses P-CSCF
2. La requête DNS est envoyée au `p_cscf_discovery_dns_server` configuré
3. Si la requête DNS réussit, les adresses P-CSCF découvertes sont envoyées au UE via PCO
4. Si la requête DNS échoue ou expire, elle revient à la liste statique `p_cscf_ipv4_address_list`
5. Voir Surveillance P-CSCF pour un suivi détaillé et des métriques

Flux de Découverte P-CSCF



Priorité de Découverte :

1. **Découverte FQDN par Règle** (Priorité la Plus Élevée) - `p_cscf_discovery_fqdn` dans la règle de sélection UPF
2. **Découverte DNS Globale** - `p_cscf_discovery_enabled: true` dans la configuration PCO globale

3. **Liste Statique PCO par Règle** - `p_cscf_ipv4_address_list` dans la substitution PCO de règle
4. **Liste Statique PCO Globale** (Secours) - `p_cscf_ipv4_address_list` dans la configuration PCO globale

Surveillance :

Tous les essais de découverte P-CSCF sont enregistrés et suivis avec des métriques :

- Taux de succès/échec des requêtes DNS
- Latence de découverte
- Statistiques d'utilisation des secours
- Métriques de découverte par règle et globale

Voir Surveillance P-CSCF pour des détails complets sur la surveillance.

Options de Configuration :

Paramètre	Type	Par Défaut	Description
`p_cscf_discovery_enabled`	Booléen	`false`	Activer la découverte dynamique P-CSCF basée sur DNS
`p_cscf_discovery_dns_server`	Tuple (IP)	`nil`	Adresse IP du serveur DNS sous forme de tuple 4 (ex. : `{10, 179, 2, 177}`)
`p_cscf_discovery_timeout_ms`	Entier	`5000`	Délai d'attente pour les requêtes DNS en millisecondes

Cas d'Utilisation :

- **Déploiements IMS Dynamiques** - Les adresses P-CSCF changent en fonction de la configuration DNS
- **Équilibrage de Charge Géographique** - DNS retourne les serveurs P-CSCF les plus proches
- **Haute Disponibilité** - DNS retourne automatiquement les serveurs P-CSCF disponibles
- **Environnements Multi-Tenants** - Différents abonnés obtiennent différents serveurs P-CSCF

Exemple : Production IMS avec Découverte DNS

```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11",

  # Activer la découverte dynamique P-CSCF
  p_cscf_discovery_enabled: true,
  p_cscf_discovery_dns_server: {10, 179, 2, 177},  # Serveur DNS
IMS
  p_cscf_discovery_timeout_ms: 3000,

  # Adresses P-CSCF de secours (si DNS échoue)
  p_cscf_ipv4_address_list: [
    "10.0.0.50",  # Secours principal
    "10.0.0.51"   # Secours secondaire
  ],

  ipv4_link_mtu_size: 1400
}
```

Découverte P-CSCF par Règle :

La découverte P-CSCF peut également être configurée par règle de sélection UPF. Cela permet à différents APN d'utiliser différents serveurs DNS pour la découverte P-CSCF :

```
# Dans la configuration de sélection upf
rules: [
  %{
    name: "IMS Traffic",
    priority: 20,
    match_field: :apn,
    match_regex: "^ims",
    upf_pool: [...],

    # Découverte P-CSCF par règle
    p_cscf_discovery_fqdn: "pcscf.mnc001.mcc001.3gppnetwork.org"
  }
]
```

Voir Configuration de Sélection UPF pour des détails sur la découverte P-CSCF par règle.

Voir aussi : Surveillance P-CSCF pour surveiller la découverte et la santé P-CSCF

Serveurs NBNS (NetBIOS)

Pour la Compatibilité des Dispositifs Windows :

```
pco: %{
  primary_nbns_server_address: "10.0.0.20",
  secondary_nbns_server_address: "10.0.0.21"
}
```

Quand Utiliser :

- Réseaux d'entreprise avec des dispositifs Windows
- Support d'applications héritées
- Résolution de noms NetBIOS requise

Taille MTU de Lien

Unité de Transmission Maximale :

```
pco: %{
  ipv4_link_mtu_size: 1400  # octets
}
```

Valeurs MTU Communes :

MTU	Cas d'Utilisation
1500	Ethernet standard (sans tunneling)
1400	Surcharge de tunneling GTP prise en compte
1420	Surcharge réduite
1280	MTU minimum IPv6
1360	Environnements VPN/tunnel

Recommandation : Utilisez **1400** pour LTE afin de tenir compte de la surcharge GTP-U.

Exemples de Configuration

APN Internet

```
pco: %{
  primary_dns_server_address: "8.8.8.8",
  secondary_dns_server_address: "8.8.4.4",
  ipv4_link_mtu_size: 1400
}
```

APN IMS

```
pco: %{
  primary_dns_server_address: "10.0.0.10",
  secondary_dns_server_address: "10.0.0.11",
  p_cscf_ipv4_address_list: [
    "10.0.0.50",
    "10.0.0.51"
  ],
  ipv4_link_mtu_size: 1400
}
```

Voir : Surveillance P-CSCF pour surveiller les taux de succès d'enregistrement IMS et la santé P-CSCF

APN Entreprise

```
pco: %{
  primary_dns_server_address: "10.100.0.10",
  secondary_dns_server_address: "10.100.0.11",
  primary_nbns_server_address: "10.100.0.20",
  secondary_nbns_server_address: "10.100.0.21",
  ipv4_link_mtu_size: 1400
}
```

PCO dans les Messages GTP-C

Réponse de Création de Session

OmniPGW inclut PCO dans le message **Réponse de Création de Session** :

```
Create Session Response
├── Cause: Request accepted
├── UE IP Address: 100.64.1.42
├── PCO (Protocol Configuration Options)
│   ├── DNS Server IPv4 Address: 8.8.8.8
│   ├── DNS Server IPv4 Address: 8.8.4.4
│   ├── P-CSCF IPv4 Address: 10.0.0.50
│   ├── P-CSCF IPv4 Address: 10.0.0.51
│   └── IPv4 Link MTU: 1400
```

Traitement UE

Le UE reçoit PCO et :

1. Configure le résolveur DNS avec les serveurs fournis
2. S'enregistre avec P-CSCF pour les services IMS
3. Définit la MTU de l'interface à la valeur spécifiée

Dépannage

Problème : UE Ne Peut Pas Résoudre DNS

Symptômes :

- UE a une adresse IP mais ne peut pas accéder à Internet
- Les recherches DNS échouent

Causes Possibles :

1. Adresses de serveur DNS incorrectes dans la configuration PCO
2. Serveurs DNS non accessibles depuis le pool IP du UE
3. Pare-feu bloquant le trafic DNS

Résolution :

```
# Tester la connectivité du serveur DNS
ping 8.8.8.8

# Tester la résolution DNS depuis le réseau UE
nslookup google.com 8.8.8.8

# Vérifier la configuration PCO
grep "primary_dns_server_address" config/runtime.exs
```

Problème : Échec de l'Enregistrement IMS

Symptômes :

- Les appels VoLTE échouent
- UE affiche "Pas d'enregistrement IMS"

Causes Possibles :

1. Configuration P-CSCF manquante
2. Adresses IP P-CSCF incorrectes
3. P-CSCF non accessible

Résolution :

```
# Vérifier la configuration P-CSCF
pco: %{
  p_cscf_ipv4_address_list: ["10.0.0.50"]  # Assurez-vous que ce
n'est pas vide
}
```

Problème : Problèmes de MTU

Symptômes :

- Certains sites Web se chargent, d'autres non
- Les transferts de fichiers volumineux échouent
- Problèmes de fragmentation

Causes Possibles :

- MTU trop grande pour la surcharge de tunneling
- MTU trop petite causant une fragmentation excessive

Résolution :

```
# Recommandé : 1400 pour le tunneling GTP
pco: %{
  ipv4_link_mtu_size: 1400
}

# Si vous avez toujours des problèmes, essayez plus bas
pco: %{
  ipv4_link_mtu_size: 1360
}
```

Meilleures Pratiques

Configuration DNS

1. **Utilisez des Serveurs DNS Fiables**
    - Public : Google (8.8.8.8), Cloudflare (1.1.1.1)
    - Privé : DNS interne pour l'entreprise
2. **Configurez Toujours un Secondaire**
    - Fournit de la redondance
    - Améliore la fiabilité
3. **Considérez la Sécurité DNS**
    - Résolveurs capables de DNSSEC
    - Filtrage DNS pour la sécurité

Configuration IMS

1. **Fournissez Plusieurs P-CSCF**
    - Au moins 2 pour la redondance
    - Distribution géographique si possible
2. **Assurez la Connectivité**
    - P-CSCF doit être accessible depuis le pool IP du UE
    - Tester la connectivité SIP

Optimisation de la MTU

1. **Tenez Compte de la Surcharge**
    - GTP-U : 36 octets (IPv4)
    - IPsec : Variable (50-100 octets)
2. **MTU Standard LTE**
    - Recommandé : **1400 octets**
    - Équilibre le débit et la compatibilité
3. **Testez de Bout en Bout**
    - Découverte MTU de chemin
    - Tester avec de gros paquets

Documentation Connexe

Guides de Configuration

- **Guide de Configuration** - Référence complète de runtime.exs, sélection UPF avec substitutions PCO
- **Allocation IP UE** - Gestion des pools IP, allocation basée sur APN

- **Surveillance P-CSCF** - Surveillance de la découverte P-CSCF, suivi de la santé, métriques

Gestion des Sessions et Interfaces

- **Gestion des Sessions** - Cycle de vie de la session PDN, établissement de porteurs
- **Interface S5/S8** - Protocole GTP-C, encodage et livraison PCO
- **Interface PFCP** - Établissement de session de plan utilisateur

IMS et VoLTE

- **Interface Diameter Gx** - Contrôle de politique pour les porteurs IMS
- **Guide de Surveillance** - Métriques et tableaux de bord liés au PCO

Retour au Guide des Opérations

Découverte et Surveillance P-CSCF

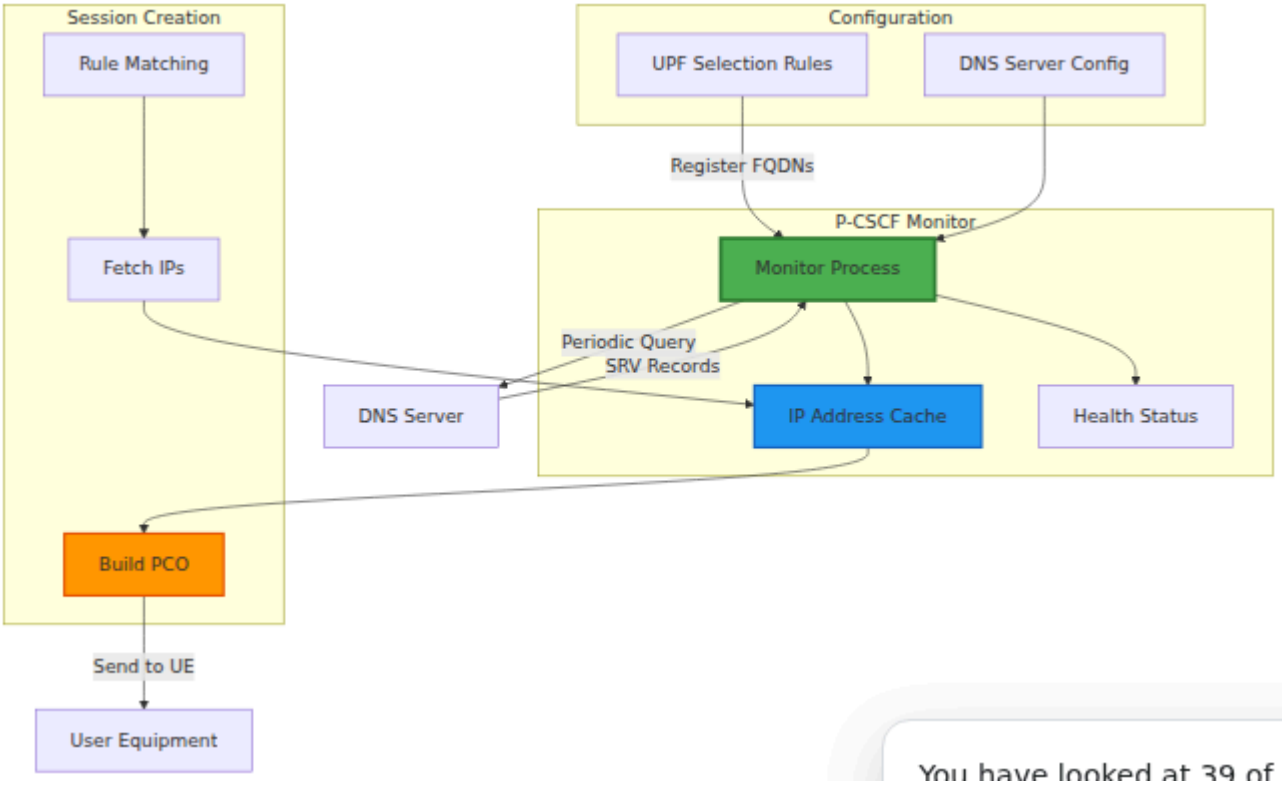
Découverte Dynamique du Serveur P-CSCF avec Surveillance en Temps Réel

OmniPGW par Omnitouch Network Services

Vue d'ensemble

La Découverte et Surveillance P-CSCF (Proxy Call Session Control Function) fournit une découverte dynamique des serveurs IMS P-CSCF en utilisant des requêtes DNS SRV avec vérification de santé SIP OPTIONS en temps réel. Cette fonctionnalité permet :

- **Découverte P-CSCF par Règle** : Différents serveurs P-CSCF pour différents types de trafic
- **Surveillance Automatique** : Processus en arrière-plan surveillant en continu la résolution DNS (toutes les 60 secondes)
- **Vérifications de Santé SIP OPTIONS** : Vérifie que les serveurs P-CSCF sont actifs via des pings SIP OPTIONS
  - **TCP d'abord** : Tente SIP OPTIONS via TCP (préféré pour la fiabilité)
  - **Repli UDP** : Repli sur UDP si TCP échoue
  - **Suivi de Statut** : Marque chaque serveur comme :up ou :down en fonction de la réponse
- **Suivi de Santé en Temps Réel** : L'API OAM expose l'état de résolution, les IP découvertes et l'état de santé
- **Repli Élégant** : Stratégie de repli à trois niveaux pour une fiabilité maximale
- **Métriques Prometheus** : Observabilité complète via des métriques Prometheus



You have looked at 39 of

Table des Matières

1. Démarrage Rapide
2. Configuration
3. Comment Ça Marche
4. Surveillance de l'API OAM
5. Métriques et Observabilité
6. Stratégie de Repli
7. Configuration DNS
8. Dépannage
9. Meilleures Pratiques

Démarrage Rapide

Configuration de Base

```
# config/runtime.exs

# Configuration globale PCO (serveur DNS pour la découverte P-
CSCF)
config :pgw_c,
  pco: %{
    p_cscf_discovery_dns_server: "192.0.2.177",
    p_cscf_discovery_enabled: true,
    p_cscf_discovery_timeout_ms: 5000
  },

  upf_selection: %{
    rules: [
      # Trafic IMS - Découverte dynamique P-CSCF
      %{
        name: "Trafic IMS",
        priority: 20,
        match_field: :apn,
        match_regex: "^ims",
        upf_pool: [
          %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80}
        ],
        # FQDN de découverte P-CSCF (voir le Guide de
Configuration pour plus de règles de sélection UPF)
        p_cscf_discovery_fqdn:
"pcscf.mnc001.mcc001.3gppnetwork.org",
        # Repli statique (voir le Guide de Configuration PCO)
        pco: %{
          p_cscf_ipv4_address_list: ["203.0.113.100",
"203.0.113.101"]
        }
      }
    ]
  }
```

Voir Guide de Configuration pour la configuration complète des règles de sélection UPF et Configuration PCO pour les options de repli P-CSCF statiques.

Surveillance d'Accès

1. Démarrer OmniPGW
2. Interroger l'API OAM : `curl -k https://localhost:8443/api/pcscf_monitor`
3. Inspecter l'état de résolution retourné et les IP découvertes (réexécuter la requête pour l'état actuel)

Configuration

Paramètres Globaux de Découverte P-CSCF

Configurer le serveur DNS utilisé pour la découverte P-CSCF dans la section PCO :

```
pco: %{
  # Serveur DNS pour la découverte P-CSCF (séparé du DNS donné à
l'UE)
  p_cscf_discovery_dns_server: "192.0.2.177",

  # Activer la fonctionnalité de découverte DNS P-CSCF
  p_cscf_discovery_enabled: true,

  # Délai d'expiration pour les requêtes DNS SRV (millisecondes)
  p_cscf_discovery_timeout_ms: 5000,

  # Adresses P-CSCF statiques (repli global)
  p_cscf_ipv4_address_list: ["203.0.113.146"]
}
```

FQDN P-CSCF par Règle

Chaque règle de sélection UPF peut spécifier son propre FQDN de découverte P-CSCF :

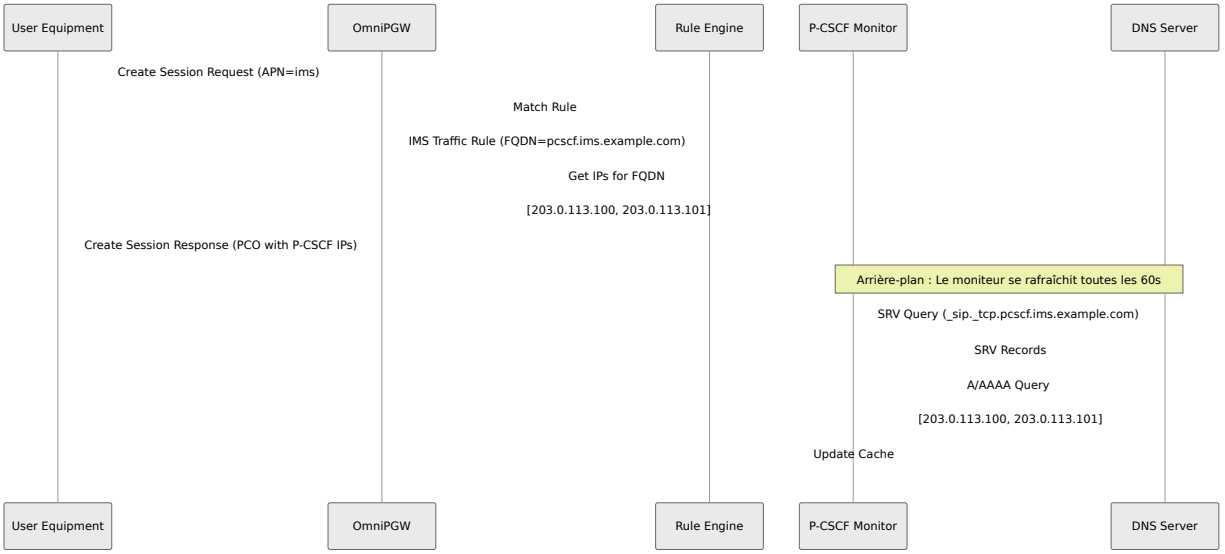
```
upf_selection: %{
  rules: [
    # Trafic IMS - P-CSCF spécifique à IMS
    %{
      name: "Trafic IMS",
      match_field: :apn,
      match_regex: "^ims",
      upf_pool: [...],
      p_cscf_discovery_fqdn:
"pcscf.ims.mnc001.mcc001.3gppnetwork.org",
      pco: %{
        p_cscf_ipv4_address_list: ["203.0.113.100"]  # Repli
      }
    },

    # Entreprise - P-CSCF spécifique à l'entreprise
    %{
      name: "Trafic Entreprise",
      match_field: :apn,
      match_regex: "^enterprise",
      upf_pool: [...],
      p_cscf_discovery_fqdn: "pcscf.enterprise.example.com",
      pco: %{
        p_cscf_ipv4_address_list: ["192.168.1.50"]  # Repli
      }
    },

    # Internet - Pas de découverte P-CSCF (utilise la
configuration globale)
    %{
      name: "Trafic Internet",
      match_field: :apn,
      match_regex: "^internet",
      upf_pool: [...]
      # Pas de p_cscf_discovery_fqdn - utilise la configuration
PCO globale
    }
  ]
}
```

Comment Ça Marche

Processus de Démarrage

1. **L'application démarre**
    - Le GenServer de surveillance P-CSCF s'initialise
    - Le parseur de configuration extrait tous les FQDN P-CSCF uniques des règles de sélection UPF
2. **Enregistrement des FQDN**
    - Chaque FQDN unique est enregistré auprès du moniteur
    - Le moniteur effectue une requête DNS SRV initiale pour chaque FQDN
    - **Vérification de Santé SIP OPTIONS** (en parallèle pour tous les serveurs découverts) :
        - Essayer TCP d'abord (`SIP/2.0/TCP` sur le port 5060)
        - Si TCP échoue, repli sur UDP (`SIP/2.0/UDP` sur le port 5060)
        - Marquer chaque serveur comme `:up` (répond) ou `:down` (pas de réponse/délai d'attente)
    - Les résultats (IPs, état de santé ou erreurs) sont mis en cache avec des horodatages
3. **Surveillance Périodique** (Toutes les 60 secondes)
    - Le moniteur rafraîchit tous les FQDN
    - Les requêtes DNS s'exécutent en arrière-plan sans blocage
    - Pour chaque serveur découvert :
        - Envoyer SIP OPTIONS via TCP (délai d'attente : 5 secondes)
        - Si TCP échoue, essayer UDP (délai d'attente : 5 secondes)
        - Mettre à jour l'état de santé en fonction de la réponse
    - Le cache est mis à jour avec les derniers résultats DNS et l'état de santé

Flux de Création de Session



Processus de Requête DNS

Le moniteur utilise **les enregistrements DNS SRV** pour la découverte directe P-CSCF :

1. **Requête SRV** : Requête des enregistrements SRV à `_sip._tcp.{fqdn}`
2. **Tri par Priorité** : Trier par priorité et poids
3. **Extraction de Cible** : Extraire les noms d'hôtes cibles des enregistrements SRV
4. **Résolution de Nom d'Hôte** : Résoudre les noms d'hôtes cibles en adresses IP (enregistrements A/AAAA)
5. **Mise en Cache** : Mettre en cache les IP résolues avec statut et horodatage

Précédence de Sélection d'Adresse P-CSCF

Lorsque le FQDN et le PCO statique sont configurés sur une règle, le FQDN prend la priorité :

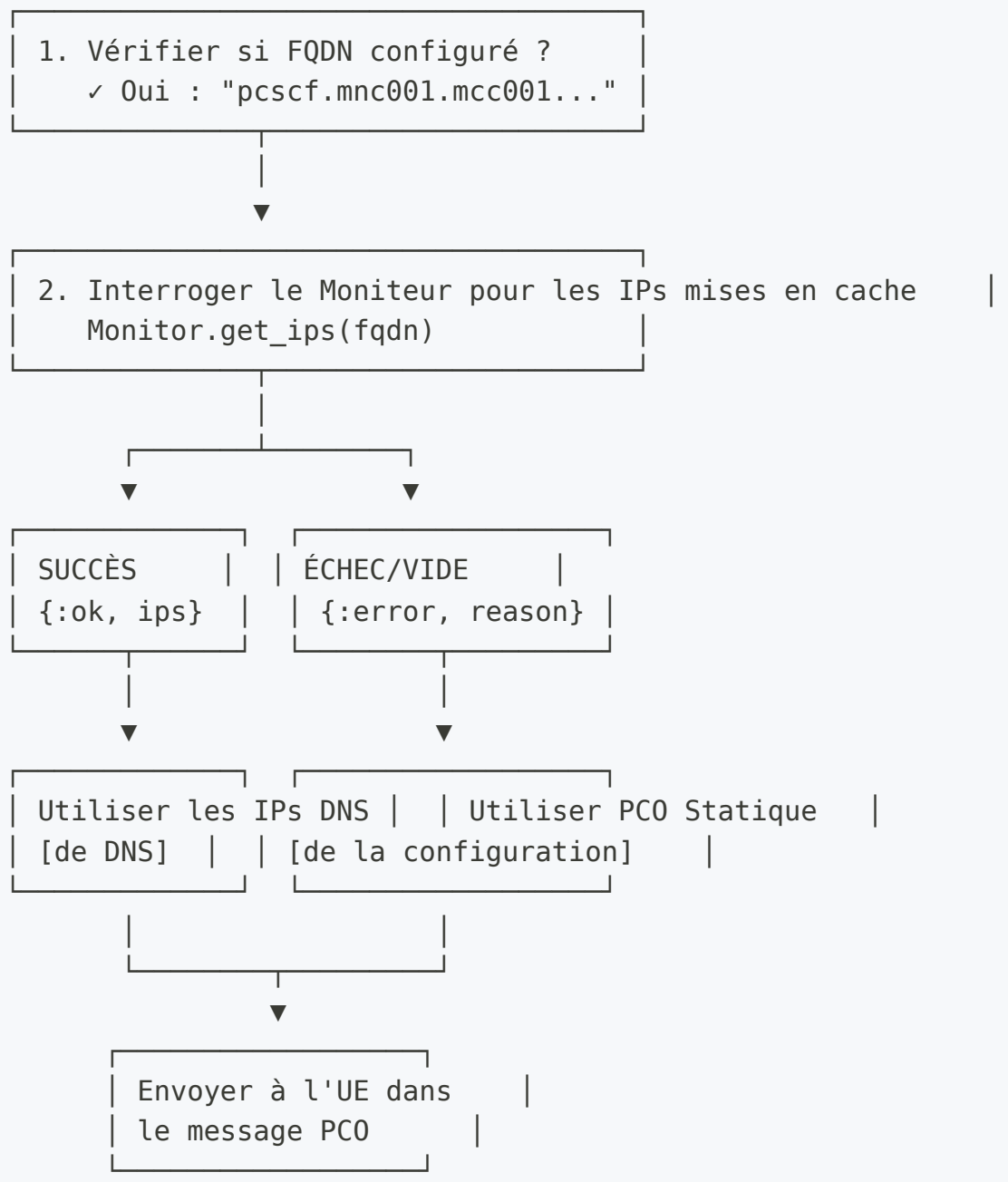
```
%{
  name: "Trafic IMS",
  p_cscf_discovery_fqdn: "pcscf.mnc001.mcc001.3gppnetwork.org",  #
← Essayer D'ABORD
  pco: %{
    p_cscf_ipv4_address_list: ["203.0.113.100", "203.0.113.101"]
# ← Repli
  }
}
```

Logique de Sélection :

Condition	Source P-CSCF	IPs Utilisées	Message
Le FQDN se résout avec succès	Découverte DNS (Moniteur)	IPs découvertes via DNS	`"Utilisati adresses P FQDN pcscf.exam`
Le FQDN échoue à se résoudre	Surcharge PCO de Règle	IPs statiques de `pco.p_cscf_ipv4_address_list`	`"Échec d'c des IPs P- FQDN..., r la configu statique"`
Le FQDN retourne une liste vide	Surcharge PCO de Règle	IPs statiques de `pco.p_cscf_ipv4_address_list`	Repli décler
Moniteur indisponible	Surcharge PCO de Règle	IPs statiques de `pco.p_cscf_ipv4_address_list`	L'erreur déc le repli
Aucun FQDN configuré	Surcharge PCO de Règle ou Globale	IPs statiques de la règle ou configuration globale	Utilise direc la configura statique

Flux d'Exemple :

Création de Session pour la Règle de Trafic IMS :



Scénarios du Monde Réel :

Scénario 1 : La Découverte DNS Fonctionne □

```
Configuration:
  p_cscf_discovery_fqdn: "pcscf.ims.example.com"
  pco.p_cscf_ipv4_address_list: ["203.0.113.100"]

Résultat DNS: [203.0.113.150, 203.0.113.151]
L'UE reçoit: [203.0.113.150, 203.0.113.151] ← De DNS
Remarque : Le PCO statique est ignoré lorsque DNS réussit
```

Scénario 2 : DNS Échoue, Repli Élégant ⚠

```
Configuration:
  p_cscf_discovery_fqdn: "pcscf.ims.example.com"
  pco.p_cscf_ipv4_address_list: ["203.0.113.100"]

Résultat DNS: ERREUR :no_naptr_records
L'UE reçoit: [203.0.113.100] ← Du PCO statique
Remarque : La session réussit malgré l'échec de DNS
```

Scénario 3 : Aucun FQDN Configuré

```
Configuration:
  # Pas de p_cscf_discovery_fqdn
  pco.p_cscf_ipv4_address_list: ["192.168.1.50"]

L'UE reçoit: [192.168.1.50] ← Du PCO statique
Remarque : La découverte DNS n'est pas tentée
```

Pourquoi ce Design ?

1. **Préférer Dynamique** : DNS offre flexibilité, équilibrage de charge et routage conscient de la localisation
2. **Assurer la Fiabilité** : Le repli statique garantit que les sessions ne échouent jamais en raison de problèmes DNS
3. **Aucune Intervention Manuelle** : Basculement automatique sans intervention de l'opérateur
4. **Sécurisé pour la Production** : Le meilleur des deux mondes - agilité avec stabilité

Recommandation : Toujours configurer à la fois le FQDN et le PCO statique pour les déploiements en production :

```
# ✓ RECOMMANDÉ : Dynamique avec repli
%{
  p_cscf_discovery_fqdn: "pcscf.ims.example.com",  # Préféré
  pco: %{
    p_cscf_ipv4_address_list: ["203.0.113.100"]     # Filet de
sécurité
  }
}

# ⚠ RISQUÉ : Dynamique uniquement (repli sur PCO global)
%{
  p_cscf_discovery_fqdn: "pcscf.ims.example.com"
  # Pas de repli spécifique à la règle !
}

# ✓ VALIDE : Statique uniquement (pas de surcharge DNS)
%{
  pco: %{
    p_cscf_ipv4_address_list: ["192.168.1.50"]
  }
}
```

Surveillance de l'API OAM

Point de Terminaison du Moniteur P-CSCF

La surveillance est exposée via l'API REST OAM (HTTPS/TLS sur le port `8443`). Tous les itinéraires sont servis sous le préfixe `/api`. Interrogez le moniteur de découverte P-CSCF avec :

```
GET /api/pcscf_monitor
```

```
curl -k https://localhost:8443/api/pcscf_monitor
```

Le point de terminaison ne prend aucun paramètre de chemin ou de requête et retourne l'état actuel de la découverte à chaque appel. La documentation interactive de l'API (Swagger UI) est disponible à `https://localhost:8443/api/docs`.

Contenu de la réponse :

- **Statistiques Générales**
  - Total des FQDN surveillés
  - FQDN résolus avec succès
  - Échecs de résolutions
  - Total des IP P-CSCF découvertes
- **Détails par FQDN**
  - FQDN surveillé
  - État de résolution (Résolu / Échoué / En Attente)
  - Nombre d'IP découvertes
  - Liste des adresses IP résolues (avec détails par serveur)
  - Horodatage de la dernière mise à jour
  - **État de Santé** : Chaque serveur découvert rapporte :
    - Adresse IP et port
    - Nom d'hôte (à partir de la cible DNS SRV)
    - Indicateur de santé (En ligne / Hors ligne)
- **Métriques de Surveillance**
  - **Total des FQDN** : Nombre de FQDN uniques enregistrés pour la surveillance
  - **Résolus avec Succès** : FQDN qui se sont résolus avec succès via DNS
  - **Échecs de Résolutions DNS** : FQDN qui n'ont pas réussi à se résoudre

- **Total des Serveurs P-CSCF** : Nombre total de serveurs découverts à travers tous les FQDN
- **Sains (SIP OPTIONS EN LIGNE)** : Serveurs répondant aux vérifications de santé SIP OPTIONS
- **Malades (SIP OPTIONS HORS LIGNE)** : Serveurs ne répondant pas aux SIP OPTIONS
- **Taux de Réussite DNS** : Pourcentage de résolutions DNS réussies
- **Intervalle de Vérification de Santé** : Fréquence des vérifications de santé SIP OPTIONS (60s, délai d'attente de 5s)

La réponse fournit une visibilité actuelle sur la santé de la résolution DNS et la disponibilité des serveurs P-CSCF via SIP OPTIONS. Pour rafraîchir, il suffit de réexécuter `GET /api/pcscf_monitor` — chaque appel retourne l'état le plus récent.

État de Sélection UPF

Le point de terminaison de l'état de sélection UPF (`GET /api/upf_selection`) rapporte l'état de découverte P-CSCF pour chaque règle :

```
 Trafic IMS (Priorité 20)
   Correspondance : Correspondance APN ^ims
   Pool : UPF-IMS-Primary (10.100.2.21:8805)

    Découverte P-CSCF
      FQDN : pcscf.mnc001.mcc001.3gppnetwork.org
      État : ✓ Résolu (2 IPs)
      IPs Résolues : 203.0.113.100, 203.0.113.101

   ⚙ Surcharges PCO
      DNS Principal : 198.51.100.195
      P-CSCF (repli statique) : 203.0.113.100, 203.0.113.101
```

Métriques et Observabilité

Métriques Prometheus

Le système de surveillance P-CSCF expose des métriques via Prometheus (port 42069 par défaut) :

Métriques de Type Gauge

```
# Métriques au niveau FQDN
pcscf_fqdns_total                       # Nombre total de FQDN
surveillés
pcscf_fqdns_resolved                    # FQDN résolus avec succès
(DNS réussi)
pcscf_fqdns_failed                      # Échecs de résolutions
FQDN (DNS échoué)

# Métriques au niveau serveur (agrégat)
pcscf_servers_total                     # Nombre total de serveurs
P-CSCF découverts via DNS SRV
pcscf_servers_healthy                   # Serveurs répondant aux
SIP OPTIONS (agrégat)
pcscf_servers_unhealthy                 # Serveurs ne répondant pas
aux SIP OPTIONS (agrégat)

# Métriques au niveau serveur (par FQDN avec étiquette)
pcscf_servers_healthy{fqdn="..."}       # Serveurs sains pour un
FQDN spécifique
pcscf_servers_unhealthy{fqdn="..."}     # Serveurs malades pour un
FQDN spécifique
```

Détails de Vérification de Santé :

- `healthy`: Le serveur a répondu au ping SIP OPTIONS (TCP ou UDP)
- `unhealthy`: Le serveur n'a pas répondu aux SIP OPTIONS (délai d'attente de 5s par transport)

Exemples de Métriques

Métriques de Résolution DNS :

```
# Requête des FQDN résolus avec succès
pcscf_fqdns_resolved

# Calculer le taux de réussite DNS
(pcscf_fqdns_resolved / pcscf_fqdns_total) * 100

# Total des serveurs découverts
pcscf_servers_total
```

Métriques de Santé SIP OPTIONS :

```
# Total des serveurs sains à travers tous les FQDN
pcscf_servers_healthy

# Total des serveurs malades
pcscf_servers_unhealthy

# Calculer le taux de réussite des vérifications de santé
(pcscf_servers_healthy / pcscf_servers_total) * 100

# Serveurs sains pour un FQDN spécifique
pcscf_servers_healthy{fqdn="pcscf.mnc001.mcc001.3gppnetwork.org"}

# Alerte sur tous les serveurs hors ligne
pcscf_servers_healthy == 0 AND pcscf_servers_total > 0
```

Exemples d'Alerte Prometheus :

```yaml
# Alerte lorsque tous les serveurs P-CSCF sont hors ligne
- alert: AllPCSCFServersDown
  expr: pcscf_servers_healthy == 0 AND pcscf_servers_total > 0
  for: 5m
  labels:
    severity: critical
  annotations:
    summary: "Tous les serveurs P-CSCF sont malades"
    description: "{{ $value }} serveurs sains (0) - tous les
contrôles SIP OPTIONS échoués"

# Alerte lorsque plus de 50% des serveurs sont hors ligne
- alert: MajorityPCSCFServersDown
  expr: (pcscf_servers_healthy / pcscf_servers_total) < 0.5
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "La majorité des serveurs P-CSCF sont malades"
    description: "Seulement {{ $value }}% des serveurs répondent
aux SIP OPTIONS"

# Alerte sur les échecs de résolution DNS
- alert: PCSCFDNSResolutionFailed
  expr: pcscf_fqdns_failed > 0
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Échecs de résolution DNS P-CSCF"
    description: "{{ $value }} FQDN(s) échouent à se résoudre"
```

Journalisation

Le moniteur enregistre des événements clés :

```
[info] Moniteur P-CSCF démarré
[info] Enregistrement de 2 FQDN P-CSCF uniques pour la
surveillance : ["pcscf.ims.example.com",
"pcscf.enterprise.example.com"]
[info] Moniteur P-CSCF : Enregistrement du FQDN
pcscf.ims.example.com
[debug] Moniteur P-CSCF : Résolution réussie de
pcscf.ims.example.com vers 2 IPs
[warning] Moniteur P-CSCF : Échec de la résolution de
pcscf.enterprise.example.com : :nxdomain
[debug] Utilisation des adresses P-CSCF du FQDN
pcscf.ims.example.com : [{10, 101, 2, 100}, {10, 101, 2, 101}]
```

Stratégie de Repli

Le système utilise une **stratégie de repli à trois niveaux** pour une fiabilité maximale :

Niveau 1 : Découverte DNS (Préférée)

```
p_cscf_discovery_fqdn: "pcscf.ims.example.com"
```

- Le moniteur interroge DNS et met en cache les IP résolues
- La session utilise les IP mises en cache si disponibles
- **Avantage** : Dynamique, équilibré, conscient de la localisation

Niveau 2 : PCO Statique Spécifique à la Règle (Repli)

```
pco: %{
  p_cscf_ipv4_address_list: ["203.0.113.100", "203.0.113.101"]
}
```

- Utilisé si la découverte DNS échoue ou ne retourne aucune IP

- Configuration statique spécifique à la règle
- **Avantage** : Repli spécifique à la r��gle, prévisible

Niveau 3 : Configuration PCO Globale (Dernier Recours)

```
# Configuration PCO globale
pco: %{
  p_cscf_ipv4_address_list: ["203.0.113.146"]
}
```

- Utilisé si aucune configuration spécifique à la règle et que DNS échoue
- Adresses P-CSCF par défaut globales
- **Avantage** : Toujours disponible, empêche l'échec de session

Exemple de Logique de Repli

```
La session correspond à la règle "Trafic IMS" :

1. Essayer la découverte DNS pour "pcscf.ims.example.com"
   ├─ Succès → Utiliser [203.0.113.100, 203.0.113.101] ✓
   └─ Échec → Essayer le niveau suivant

2. Essayer la surcharge PCO de la règle
   ├─ Configuré → Utiliser [203.0.113.100, 203.0.113.101] ✓
   └─ Non configuré → Essayer le niveau suivant

3. Utiliser la configuration PCO globale
   └─ Utiliser [203.0.113.146] ✓ (Réussit toujours)
```

Configuration DNS

Configuration du Serveur DNS

Configurer le serveur DNS avec des enregistrements SRV et A/AAAA pour la découverte P-CSCF :

```
; Enregistrements SRV pour P-CSCF (_sip._tcp préfixe est interrogé
automatiquement)
_sip._tcp.pcscf.mnc001.mcc001.3gppnetwork.org. IN SRV 10 50 5060
pcscf1.example.com.
_sip._tcp.pcscf.mnc001.mcc001.3gppnetwork.org. IN SRV 20 50 5060
pcscf2.example.com.

; Enregistrements A
pcscf1.example.com. IN A 203.0.113.100
pcscf2.example.com. IN A 203.0.113.101
```

Important : OmniPGW préfixe automatiquement `_sip._tcp.` au FQDN configuré. Si vous configurez `p_cscf_discovery_fqdn: "pcscf.mnc001.mcc001.3gppnetwork.org"`, le système interrogera `_sip._tcp.pcscf.mnc001.mcc001.3gppnetwork.org`.

Format d'Enregistrement SRV

Les enregistrements SRV suivent ce format :

```
_service._proto.domain. IN SRV priority weight port target.
```

- **Priorité** : Les valeurs plus basses ont une priorité plus élevée (10 avant 20)
- **Poids** : Pour l'équilibrage de charge parmi la même priorité (plus élevé = plus de trafic)
- **Port** : Port SIP (généralement 5060 pour TCP, 5060 pour UDP)
- **Cible** : Nom d'hôte à résoudre en adresse IP

Tester la Configuration DNS

```
# Interroger les enregistrements SRV (notez le préfixe _sip._tcp)
dig SRV _sip._tcp.pcscf.mnc001.mcc001.3gppnetwork.org @192.0.2.177

# Sortie attendue :
# _sip._tcp.pcscf.mnc001.mcc001.3gppnetwork.org. 300 IN SRV 10 50
5060 pcscf1.example.com.

# Résoudre le nom d'hôte P-CSCF en IP
dig A pcscf1.example.com @192.0.2.177

# Sortie attendue :
# pcscf1.example.com. 300 IN A 203.0.113.100
```

Dépannage

Problème : FQDN Affiche un Statut "Échoué"

Symptômes :

- `GET /api/pcscf_monitor` rapporte un statut Échoué
- Erreur : `:nxdomain`, `:timeout`, ou `:no_naptr_records`

Causes Possibles :

1. Serveur DNS non accessible
2. FQDN n'existe pas dans DNS
3. Aucun enregistrement NAPTR configuré
4. Délai d'attente du serveur DNS

Résolution :

```
# 1. Tester la connectivité du serveur DNS
ping 192.0.2.177

# 2. Tester la requête NAPTR manuellement
dig NAPTR pcscf.mnc001.mcc001.3gppnetwork.org @192.0.2.177

# 3. Vérifier les journaux d'OmniPGW
grep "P-CSCF" /var/log/pgw_c.log

# 4. Vérifier la configuration
grep "p_cscf_discovery_dns_server" config/runtime.exs

# 5. Re-interroger l'API OAM pour l'état actuel
curl -k https://localhost:8443/api/pcscf_monitor
```

Problème : Aucune IP Retournée

Symptômes :

- `GET /api/pcscf_monitor` rapporte "0 IPs"
- Le statut peut être Résolu ou Échoué

Causes Possibles :

1. Les enregistrements NAPTR existent mais les FQDN de remplacement ne se résolvent pas
2. Le champ de service ne correspond pas au modèle IMS/SIP
3. Les enregistrements A/AAAA manquent

Résolution :

```
# Vérifier le champ de service de l'enregistrement NAPTR
dig NAPTR pcscf.example.com @192.0.2.177

# Assurez-vous que le service contient "SIP" ou "IMS" :
# CORRECT : "SIP+D2U", "x-3gpp-ims:sip"
# FAUX : "HTTP", "FTP"

# Vérifier que les enregistrements A/AAAA existent
dig pcscf1.example.com A @192.0.2.177
```

Problème : Les Sessions Utilisent le Mauvais P-CSCF

Symptômes :

- L'UE reçoit des adresses P-CSCF inattendues
- Repli statique utilisé au lieu des IP découvertes

Causes Possibles :

1. La découverte DNS a échoué mais le repli fonctionne
2. Correspondance de règle incorrecte
3. FQDN non enregistré

Résolution :

```
# 1. Vérifier la page du Moniteur P-CSCF
# Vérifier que le FQDN est enregistré et résolu

# 2. Vérifier les journaux de session
grep "Utilisation des adresses P-CSCF du FQDN" /var/log/pgw_c.log

# 3. Vérifier la page de Sélection UPF
# Vérifier que la règle montre le bon FQDN et statut

# 4. Tester la correspondance de règle
# Créer une session avec un APN spécifique et vérifier quelle
règle correspond
```

Problème : Latence Élevée des Requêtes DNS

Symptômes :

- Création de session lente
- Les métriques montrent une haute `pcscf_discovery_query_duration_seconds`

Causes Possibles :

1. Problèmes de performance du serveur DNS
2. Latence réseau vers le serveur DNS
3. Délai d'attente trop élevé

Résolution :

```
# Réduire le délai d'expiration de la requête
pco: %{
  p_cscf_discovery_timeout_ms: 2000  # Réduire de 5000ms
}

# Envisager d'utiliser un serveur DNS plus proche
pco: %{
  p_cscf_discovery_dns_server: "10.0.0.10"  # DNS local
}
```

Meilleures Pratiques

1. Sélection du Serveur DNS

Utiliser un Serveur DNS Dédié

```
pco: %{
  # DNS dédié pour la découverte P-CSCF (pas le même que le DNS de
l'UE)
  p_cscf_discovery_dns_server: "192.0.2.177",

  # Serveurs DNS de l'UE (donnés aux appareils mobiles)
  primary_dns_server_address: "8.8.8.8",
  secondary_dns_server_address: "8.8.4.4"
}
```

Pourquoi ?

- Séparer les préoccupations : DNS de l'UE vs. DNS IMS interne
- Différentes politiques d'accès et de sécurité
- Mise à l'échelle et fiabilité indépendantes

2. Toujours Configurer un Repli Statique

```
%{
  p_cscf_discovery_fqdn: "pcscf.ims.example.com",  # Préféré
  pco: %{
    p_cscf_ipv4_address_list: ["203.0.113.100"]  # Repli requis
  }
}
```

Pourquoi ?

- Garantit que les sessions réussissent même si DNS échoue
- Dégradation élégante
- Respecte les exigences SLA

3. Utiliser des FQDN Spécifiques par Type de Trafic

```
rules: [
  # IMS
  %{
    name: "IMS",
    match_regex: "^ims",
    p_cscf_discovery_fqdn:
"pcscf.ims.mnc001.mcc001.3gppnetwork.org"
  },

  # Entreprise
  %{
    name: "Entreprise",
    match_regex: "^enterprise",
    p_cscf_discovery_fqdn: "pcscf.enterprise.example.com"
  }
]
```

Pourquoi ?

- Différents pools P-CSCF par service
- Meilleure distribution de charge
- Routage spécifique au service

4. Surveiller la Performance des Requêtes DNS

```
# Alerte sur la latence élevée des requêtes P-CSCF
alert: HighPCSCFQueryLatency
expr: histogram_quantile(0.95,
pcscf_discovery_query_duration_seconds_bucket) > 2
for: 5m
labels:
  severity: warning
annotations:
  summary: "Les requêtes DNS P-CSCF sont lentes (p95 > 2s)"
```

5. Vérifications de Santé DNS Régulières

- **API OAM** : Interroger `GET /api/pcscf_monitor` quotidiennement
- **Métriques** : Surveiller la métrique `pcscf_monitor_fqdns_failed`
- **Journaux** : Surveiller les erreurs DNS
- **Tests** : Vérifier périodiquement que les enregistrements DNS existent

6. Configurer un Délai d'Expiration Approprié

```
# Production : Équilibrer fiabilité et latence
pco: %{
  p_cscf_discovery_timeout_ms: 5000  # 5 secondes
}

# Haute performance : Favoriser la vitesse, compter sur le repli
pco: %{
  p_cscf_discovery_timeout_ms: 2000  # 2 secondes
}
```

7. Utiliser la Redondance DNS

Configurer un DNS primaire et secondaire :

```
# DNS P-CSCF primaire
pcscf.mnc001.mcc001.3gppnetwork.org. IN NAPTR 10 50 "s" "SIP+D2U"
"" _sip._udp.pcscf1.example.com.

# DNS P-CSCF secondaire
pcscf.mnc001.mcc001.3gppnetwork.org. IN NAPTR 20 50 "s" "SIP+D2U"
"" _sip._udp.pcscf2.example.com.
```

Documentation Connexe

- **Configuration PCO** - Options de Configuration de Protocole, paramètres DNS et P-CSCF

- **Guide de Configuration** - Référence complète de configuration OmniPGW
- **Surveillance** - Métriques, journalisation et observabilité
- **Gestion de Session** - Cycle de vie de la session et livraison PCO
- **Interface PFCP** - Communication de la Fonction de Plan Utilisateur

---

Retour à la Documentation Principale

Documentation de l'interface PFCP/Sxb

Protocole de contrôle de transfert de paquets - Communication entre PGW-C et PGW-U

Table des matières

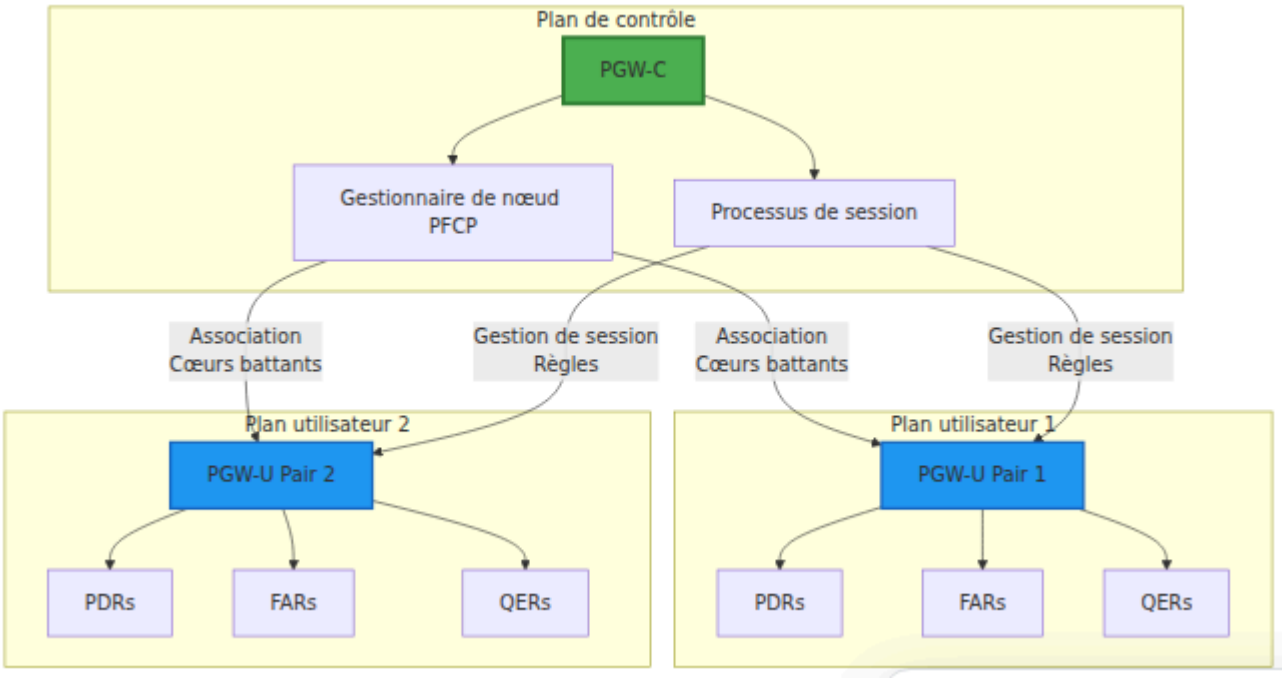
1. Aperçu
2. Principes de base du protocole
3. Gestion des associations PFCP
4. Gestion des sessions PFCP
5. Règles de traitement des paquets
6. Configuration
7. Sélection de l'UPF basée sur DNS
8. Flux de messages
9. Dépannage
10. Surveillance PFCP
11. Documentation connexe

Aperçu

L'**interface Sxb** utilise le **PFCP (Protocole de contrôle de transfert de paquets)** pour la communication entre le PGW-C (plan de contrôle) et le PGW-U (plan utilisateur). Cette séparation permet :

- **Plan de contrôle (PGW-C)** - Gère la signalisation, la gestion des sessions, les décisions de politique
- **Plan utilisateur (PGW-U)** - Gère le transfert de paquets réel à grande vitesse

Architecture PFCP



Principes de base du protocole

Version PFCP

PGW-C implémente **PFCP Version 1** (3GPP TS 29.244).

Transport

- **Protocole :** UDP
- **Port par défaut :** 8805
- **Format de message :** Encodé binaire selon la spécification PFCP

Types d'ID de nœud

Les pairs PFCP sont identifiés par l'ID de nœud, qui peut être :

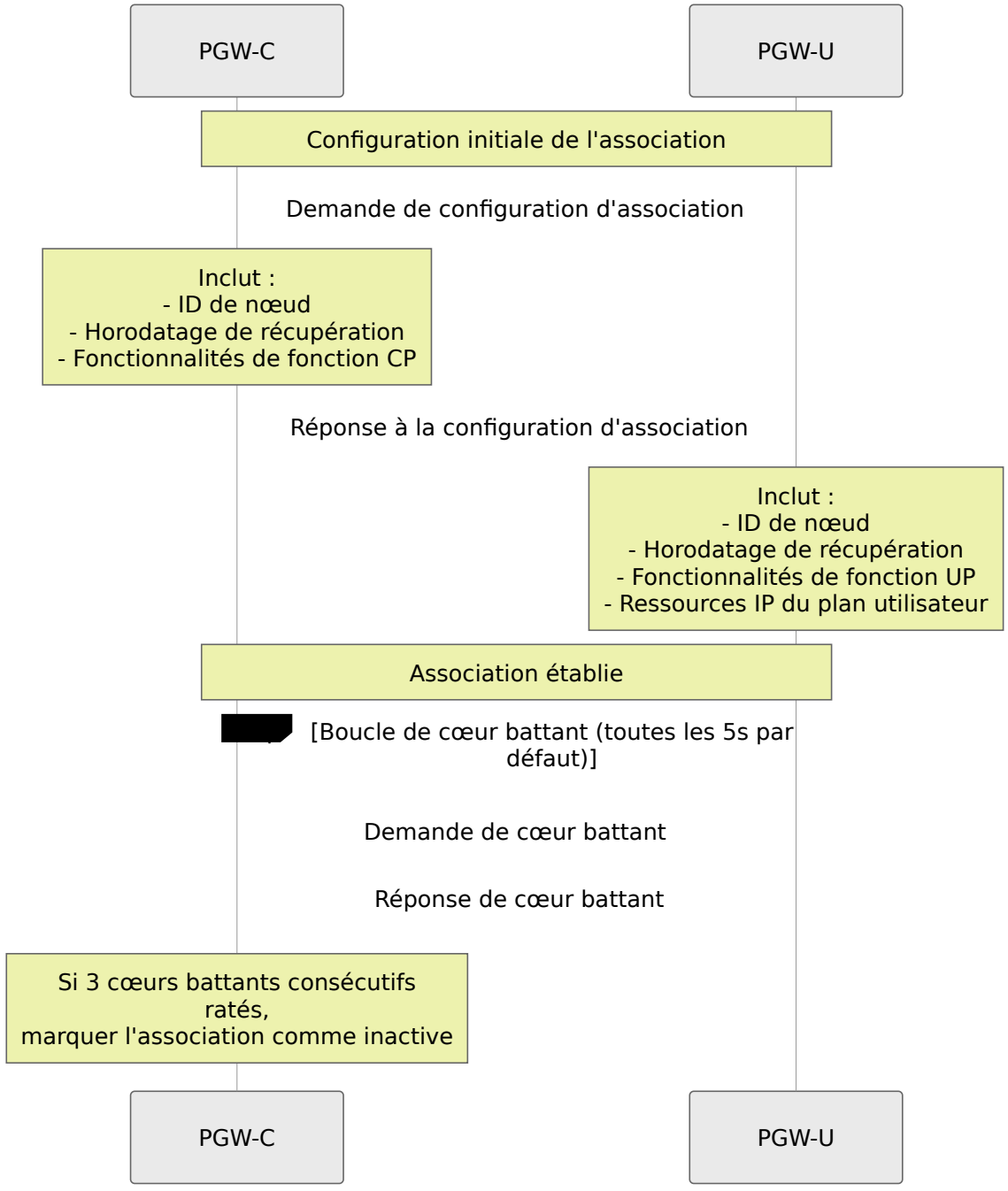
- **Adresse IPv4** - La plus courante
- **Adresse IPv6**

- **FQDN** (Nom de domaine entièrement qualifié)

Gestion des associations PFCP

Avant que la gestion des sessions puisse avoir lieu, une **association** PFCP doit être établie entre PGW-C et PGW-U.

Flux de configuration de l'association



Gestion de l'état des pairs

Chaque pair PFCP maintient un état :

Champ	Description
`is_associated`	Booléen indiquant l'état de l'association
`remote_node_id`	ID de nœud du pair (IP ou FQDN)
`remote_ip_address`	Adresse IP pour la communication
`remote_port`	Port UDP (par défaut 8805)
`heartbeat_period_ms`	Intervalle de cœur battant en millisecondes
`missed_heartbeats_consecutive`	Compte des cœurs battants manqués
`up_function_features`	Fonctionnalités du plan utilisateur prises en charge
`up_recovery_time_stamp`	Horodatage de récupération du pair

Mécanisme de cœur battant

But : Détecter les pannes des pairs et maintenir la vivacité de l'association

Configuration :

```
# Dans config/runtime.exs
sxb: %{
  local_ip_address: "10.0.0.20"
},
upf_selection: %{
  fallback_pool: [
    %{remote_ip_address: "10.0.0.21", remote_port: 8805, weight:
100}
  ]
}
# Tous les UPFs sont automatiquement enregistrés avec des cœurs
battants de 5 secondes
```

Détection de panne :

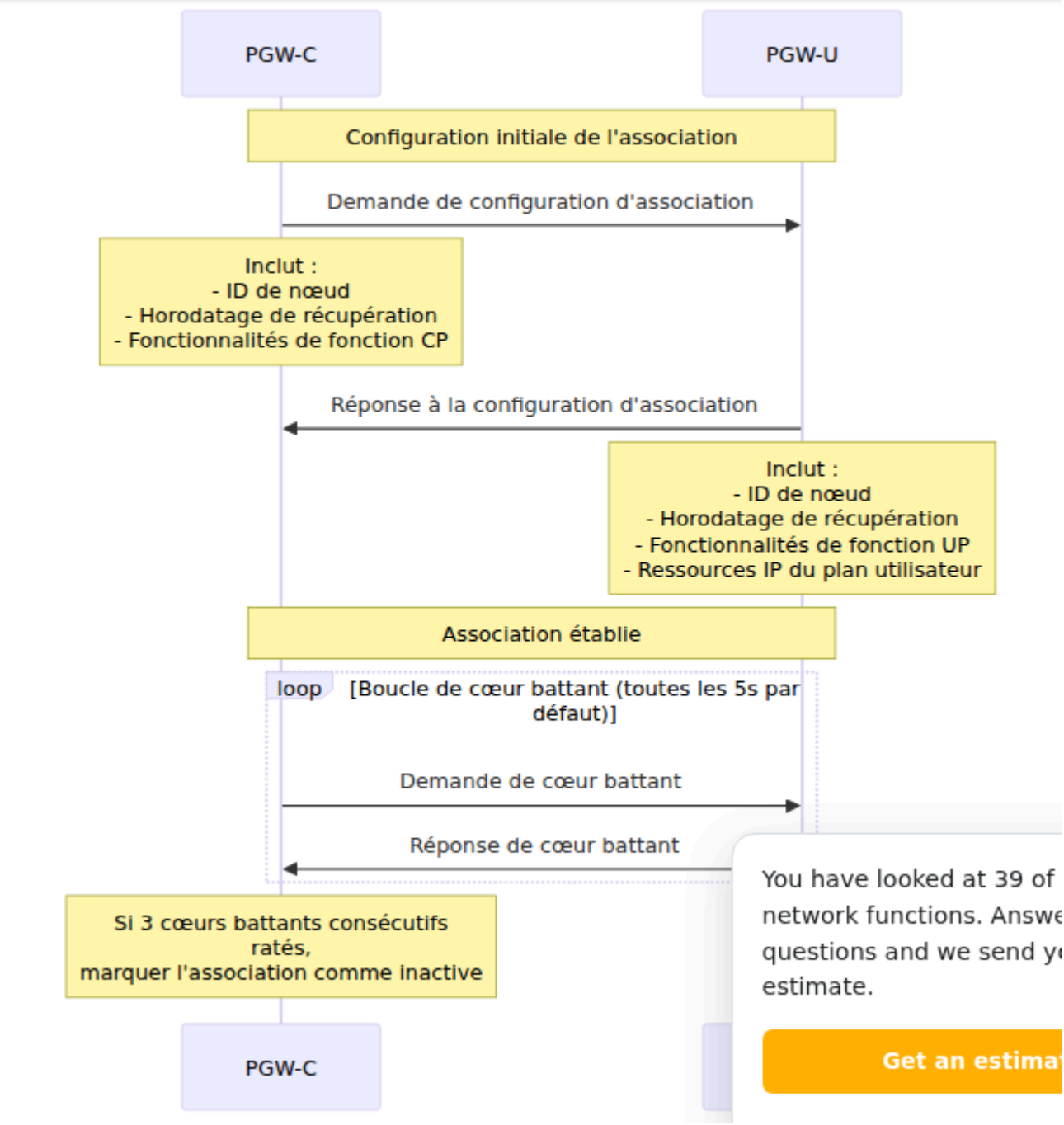
- Chaque cœur battant manqué incrémente `missed_heartbeats_consecutive`
- Typiquement configuré pour échouer après 3 manques consécutifs
- L'association échouée empêche de nouvelles sessions vers ce pair

---

Gestion des sessions PFCP

Les sessions PFCP sont créées pour chaque connexion PDN UE afin de programmer des règles de transfert dans le plan utilisateur.

Cycle de vie de la session



Établissement de session

Quand : L'UE se connecte et crée une connexion PDN

PGW-C envoie à PGW-U :

Demande d'établissement de session contenant :

- **SEID** (ID de point de terminaison de session) - Identifiant de session unique
- **ID de nœud** - ID de nœud de PGW-C
- **F-SEID** - SEID entièrement qualifié (comprend IP + SEID)
- **PDRs** - Règles de détection de paquets (typiquement 2 : liaison montante + liaison descendante)
- **FARs** - Règles d'action de transfert (typiquement 2 : liaison montante + liaison descendante)
- **QERs** - Règles d'application de QoS (limites de débit)
- **BAR** - Règle d'action de mise en mémoire tampon (pour la mise en mémoire tampon en liaison descendante)

PGW-U répond :

Réponse d'établissement de session contenant :

- **Cause** - Raison de succès ou d'échec
- **F-SEID** - Point de terminaison de session de PGW-U
- **PDRs créés** - Accusé de réception des règles créées
- **F-TEID** - TEID entièrement qualifié pour l'interface S5/S8

Modification de session

Quand : Des changements de QoS, des mises à jour de politique ou des modifications de bearer se produisent

La modification peut inclure :

- Ajout de nouveaux PDRs, FARs, QERs
- Suppression de règles existantes
- Mise à jour des paramètres de règle

Suppression de session

Quand : L'UE se déconnecte ou la connexion PDN est terminée

Processus :

1. PGW-C envoie une demande de suppression de session avec SEID
2. PGW-U supprime toutes les règles et libère les ressources
3. PGW-U répond avec une réponse de suppression de session

Allocation de F-TEID

F-TEID (Identifiant de point de terminaison de tunnel entièrement qualifié) identifie les points de terminaison de tunnel GTP-U pour le trafic du plan utilisateur. Lors de l'établissement d'une session PFCP, quelqu'un doit allouer le F-TEID qui identifie où l'UPF doit envoyer le trafic de liaison montante. Il existe deux approches :

Comprendre l'allocation de F-TEID

Ce qui est alloué : Le F-TEID se compose de :

- **TEID** (Identifiant de point de terminaison de tunnel) - Numéro de 32 bits identifiant le tunnel
- **Adresse IP** - Où envoyer les paquets GTP-U (l'adresse IP de l'UPF)

La question : Qui alloue la valeur TEID ?

Option 1 : Allocation par l'UPF (Recommandé par défaut)

- PGW-C dit "veuillez allouer un TEID pour moi" (drapeau CHOOSE)
- L'UPF choisit un TEID dans son pool local et répond avec la valeur

Option 2 : Allocation par PGW-C (Mode de compatibilité)

- PGW-C choisit un TEID et dit à l'UPF "utilisez ce TEID spécifique"
- L'UPF utilise le TEID fourni sans allocation

Allocation par l'UPF (Défaut - Recommandé)

Configuration :

```
sxb: %{
  allocate_uplink_f_teid: false  # Par défaut
}
```

Comment cela fonctionne :

1. PGW-C construit une demande d'établissement de session PFCP avec le drapeau F-TEID CHOOSE
2. L'UPF reçoit la demande, alloue le TEID de son pool interne
3. L'UPF répond avec le F-TEID alloué (TEID + adresse IP)
4. PGW-C stocke le F-TEID alloué pour la durée de la session

Pourquoi c'est mieux (généralement) :

□ Séparation des préoccupations

- L'UPF possède le plan utilisateur = L'UPF gère les identifiants du plan utilisateur
- Pas besoin pour PGW-C de suivre quels TEIDs l'UPF a disponibles
- Chaque composant gère son propre pool de ressources

□ Scalabilité Multi-PGW-C

- Plusieurs instances de PGW-C peuvent communiquer avec le même UPF sans coordination
- Aucun risque de collisions de TEID entre différentes instances de PGW-C
- L'UPF assure l'unicité entre tous les pairs du plan de contrôle

□ Comportement standard 3GPP

- Le drapeau CHOOSE est défini dans 3GPP TS 29.244 à cet effet
- Les implémentations modernes de l'UPF le prennent en charge
- Suit le principe "laisser le propriétaire allouer"

□ Basculage plus simple

- Si PGW-C redémarre, l'UPF possède toujours l'espace de noms TEID
- Pas besoin de synchroniser l'état d'allocation de TEID

- L'UPF peut continuer à utiliser les TEIDs existants

Quand l'utiliser :

- □ Déploiements de production avec des UPFs modernes (par défaut)
- □ Déploiements Multi-PGW-C partageant des pools d'UPF
- □ Architectures cloud-native avec des plans de contrôle sans état
- □ Vous voulez un comportement PFCP standard 3GPP

Problèmes potentiels :

- ⚠ Certaines implémentations d'UPF héritées ou propriétaires ne prennent pas en charge le drapeau CHOOSE
- ⚠ Si l'établissement de session échoue avec "IE obligatoire manquant" ou similaire, l'UPF peut ne pas prendre en charge CHOOSE

Allocation par PGW-C (Compatibilité héritée)

Configuration :

```
sxb: %{
  allocate_uplink_f_teid: true
}
```

Comment cela fonctionne :

1. PGW-C alloue le TEID à partir du pool local lors de la création de la session
2. PGW-C construit une demande d'établissement de session PFCP avec une valeur TEID explicite
3. L'UPF reçoit la demande, utilise le TEID fourni sans allocation
4. PGW-C et l'UPF suivent tous deux la même valeur TEID

Pourquoi vous pourriez avoir besoin de cela :

□ L'UPF ne prend pas en charge CHOOSE

- Certaines implémentations d'UPF (surtout héritées/propriétaires) ne prennent pas en charge l'allocation dynamique

- L'UPF s'attend à un TEID explicite dans la demande d'établissement de session PFCP
- Seul contournement pour la compatibilité

** Gestion centralisée des TEID**

- Si vous avez besoin que PGW-C ait une visibilité complète sur tous les TEIDs alloués
- Utile pour le débogage des problèmes du plan utilisateur (PGW-C connaît les valeurs TEID exactes)
- Peut corréler le TEID dans les captures de paquets avec l'état de la session

** Allocation déterministe**

- Si vous avez besoin de modèles d'allocation TEID prévisibles
- Certains environnements de test peuvent nécessiter des plages de TEID spécifiques

Compromis :

⚠ **Coordination requise pour Multi-PGW-C**

- Plusieurs instances de PGW-C partageant un UPF doivent éviter les collisions de TEID
- Nécessite soit :
  - Plages de TEID partitionnées par PGW-C (configuration complexe)
  - Service d'allocation de TEID partagé (infrastructure supplémentaire)
  - Accepter le risque de collision avec une allocation aléatoire (faible probabilité)

⚠ **Synchronisation de l'état**

- PGW-C doit suivre les TEIDs alloués pour éviter la réutilisation
- L'état du pool de TEID est perdu lors du redémarrage de PGW-C (doit être reconstruit à partir des sessions)
- Scénarios de basculement plus complexes

⚠ **Comportement non standard**

- Pas le modèle de conception PFCP prévu
- Peut ne pas fonctionner avec toutes les implémentations d'UPF s'attendant à CHOOSE

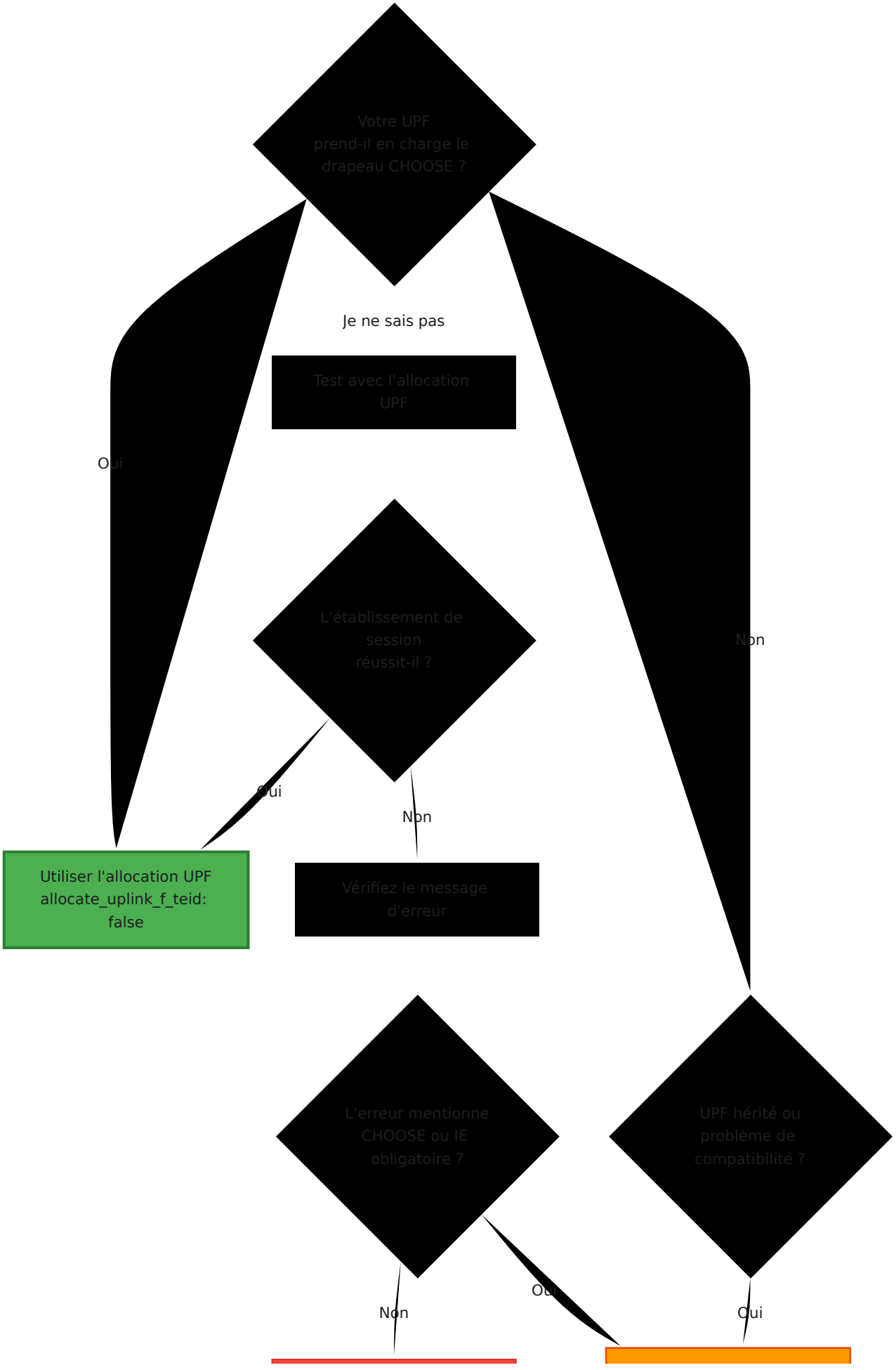
Quand l'utiliser :

- ⚠ **Uniquement lorsque l'UPF ne prend pas en charge le drapeau CHOOSE**
- ⚠ Implémentations d'UPF héritées (par exemple, certains matériels propriétaires)
- ⚠ Exigences de compatibilité spécifiques
- ⚠ Scénarios de débogage nécessitant la visibilité TEID de PGW-C

Gestion des collisions de TEID : PGW-C utilise une allocation aléatoire avec détection de collision :

- Plage TEID : 1 à 0xFFFFFFFF (4,2 milliards de valeurs)
- Probabilité de collision : ~0,023 % à 1 million de sessions
- Réessai automatique en cas de collision (transparent pour l'appelant)
- Les TEIDs sont automatiquement libérés lorsque la session se termine

Comment choisir


Votre UPF prend-il en charge le drapeau CHOOSE ?
Je ne sais pas
Test avec l'allocation UPF
Oui
Non
L'établissement de session réussit-il ?
Oui
Non
Utiliser l'allocation UPF allocate_uplink_f_teid: false
Vérifiez le message d'erreur
L'erreur mentionne CHOOSE ou IE obligatoire ?
UPF hérité ou problème de compatibilité ?
Non
Oui
Oui



Dépannage

Symptôme : L'établissement de session échoue immédiatement

Vérifiez les journaux PFCP :

```
# Recherchez les erreurs liées à CHOOSE
grep -i "choose\|mandatory.*missing" /var/log/pgw_c.log

# Vérifiez les codes de cause de la réponse d'établissement de
session PFCP
grep "Session Establishment Response" /var/log/pgw_c.log
```

Si l'UPF rejette le drapeau CHOOSE :

- L'erreur peut dire "IE obligatoire manquant" ou "IE invalide"
- L'UPF s'attend à un F-TEID explicite mais a reçu CHOOSE
- **Solution :** Définir `allocate_uplink_f_teid: true`

Si l'allocation PGW-C cause des problèmes :

- Très rare - l'espace TEID est énorme (4 milliards de valeurs)
- Vérifiez l'épuisement de TEID (improbable en dessous de millions de sessions) :

```
# Vérifiez le compte du registre
grep "registered_teid_count" /var/log/pgw_c.log
```

Changement entre les modes :

```
# Éditez config/runtime.exs
sxb: %{
  local_ip_address: "10.0.0.20",
  allocate_uplink_f_teid: false  # Changez en true si l'UPF ne
prend pas en charge CHOOSE
}
```

Puis redémarrez PGW-C :

```
systemctl restart pgw_c
```

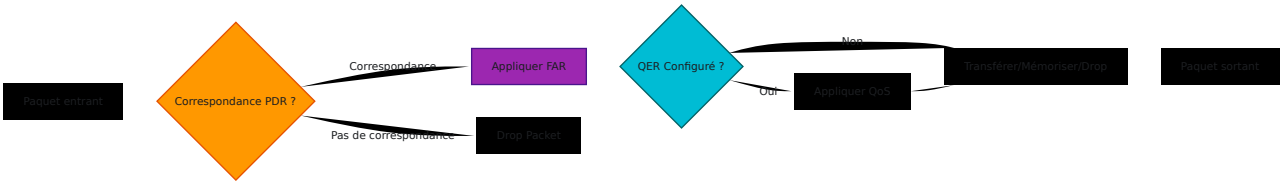
Vérification du mode actif : Vérifiez les captures de paquets PFCP :

```
# Capturez le trafic PFCP
tcpdump -i any -n port 8805 -w pfcp.pcap

# Ouvrez dans Wireshark et regardez la demande d'établissement de
session
# Si le F-TEID montre des drapeaux "CHOOSE" : mode d'allocation
UPF
# Si le F-TEID montre une valeur TEID explicite : mode
d'allocation PGW-C
```

Règles de traitement des paquets

PFCP utilise un ensemble de règles pour définir comment le plan utilisateur traite les paquets.

Architecture des règles



PDR (Règle de détection de paquets)

But : Identifier à quels paquets cette règle s'applique

Configuration typique de PGW-C :

PDR #1 - Liaison descendante :

```
ID PDR : 1
Précédence : 100
PDI (Informations de détection de paquets) :
  - Interface source : CORE (côté Internet)
  - Adresse IP de l'UE : 100.64.1.42/32
ID FAR : 1 (règle de transfert associée)
```

PDR #2 - Liaison montante :

```
ID PDR : 2
Précédence : 100
PDI (Informations de détection de paquets) :
  - Interface source : ACCESS (côté SGW)
  - F-TEID : <point de terminaison de tunnel S5/S8>
ID FAR : 2 (règle de transfert associée)
ID QER : 1 (application de QoS)
```

Champs clés de PDR :

- **ID PDR** - Identifiant de règle unique (par session)
- **Précédence** - Priorité de correspondance de règle (plus élevé = plus spécifique)
- **PDI** - Critères de correspondance (interface, IP, TEID, etc.)
- **Suppression de l'en-tête externe** - Supprimer l'en-tête GTP-U à l'entrée
- **ID FAR** - Action de transfert associée
- **ID QER** - Application de QoS associée (facultatif)

FAR (Règle d'action de transfert)

But : Définir quoi faire avec les paquets correspondants

FAR #1 - Liaison descendante (Internet → UE) :

```
ID FAR : 1
Appliquer l'action : TRANSFÉRER
Paramètres de transfert :
  - Interface de destination : ACCESS (vers SGW)
  - Création d'en-tête externe : GTP-U/UDP/IPv4
  - F-TEID distant : <point de terminaison de tunnel SGW S5/S8>
```

FAR #2 - Liaison montante (UE → Internet) :

```
ID FAR : 2
Appliquer l'action : TRANSFÉRER
Paramètres de transfert :
  - Interface de destination : CORE (vers Internet)
  - (Pas d'en-tête externe - transfert IP simple)
```

Champs clés de FAR :

- **ID FAR** - Identifiant de règle unique
- **Appliquer l'action** - TRANSFÉRER, DROP, BUFFER, NOTIFIER
- **Paramètres de transfert :**
  - Interface de destination (ACCESS/CORE)
  - Création d'en-tête externe (ajouter un tunnel GTP-U)
  - Instance réseau (VRF/table de routage)

QER (Règle d'application de QoS)

But : Appliquer des limites de débit et des paramètres de QoS. Les QER peuvent également suivre l'utilisation pour la gestion des quotas de facturation en ligne (voir Interface Diameter Gy pour le contrôle de crédit).

Exemple de QER :

```
ID QER : 1
État de la porte : OUVERT
Débit maximum :
  - Liaison montante : 100 Mbps
  - Liaison descendante : 50 Mbps
Débit garanti : (facultatif, pour les porteurs GBR)
  - Liaison montante : 10 Mbps
  - Liaison descendante : 10 Mbps
```

Champs clés de QER :

- **ID QER** - Identifiant de règle unique
- **État de la porte** - OUVERT (autoriser) ou FERMÉ (bloquer)
- **MBR** - Débit maximum (liaison montante/descendante)
- **GBR** - Débit garanti (pour les porteurs dédiés)
- **QCI** - Identifiant de classe de QoS (affecte la planification)

BAR (Règle d'action de mise en mémoire tampon)

But : Contrôler la mise en mémoire tampon des paquets en liaison descendante lorsque l'UE est inactif

Exemple de BAR :

```
ID BAR : 1
Délai de notification de données en liaison descendante : 100ms
Nombre de paquets suggérés pour la mise en mémoire tampon : 10
```

Utilisé pour : Optimisation DRX (réception discontinue) en mode inactif

---

Configuration

Configuration de base de Sxb

Éditez `config/runtime.exs` :

```
config :pgw_c,
  sxb: %{
    # Adresse IP locale pour la communication PFCP
    local_ip_address: "10.0.0.20",

    # Facultatif : Remplacer le port par défaut (8805)
    local_port: 8805,

    # Délai d'attente de la demande PFCP en millisecondes (par
défaut : 500ms)
    # Temps à attendre pour la réponse de l'UPF avant de
retransmettre
    # Doit être >= temps de traitement attendu de l'UPF pour
éviter les sessions dupliquées
    request_timeout_ms: 500,

    # Nombre de tentatives de retransmission pour les demandes
PFCP (par défaut : 3)
    # Temps d'attente maximum total = request_timeout_ms *
request_attempts
    request_attempts: 3,

    # Facultatif : Contrôler l'allocation de F-TEID pour le plan
utilisateur
    # Lorsque faux (par défaut) : l'UPF alloue F-TEID (drapeau
CHOOSE)
    # Lorsque vrai : PGW-C pré-alloue F-TEID et fournit une valeur
explicite
    # Remarque : Certains UPFs peuvent ne pas prendre en charge le
drapeau CHOOSE et nécessiter une allocation explicite
    allocate_uplink_f_teid: false
  },

  # Sélection de l'UPF - Tous les UPFs définis ici sont
automatiquement enregistrés
  upf_selection: %{
    fallback_pool: [
      %{
        # Adresse IP de PGW-U
        remote_ip_address: "10.0.0.21",

        # Port PFCP (par défaut : 8805)
        remote_port: 8805,
```

```
        # Poids pour l'équilibrage de charge (100 = normal, 0 =
veille)
        weight: 100
      }
    ]
  }
```

Configuration du délai d'attente des demandes

L'interface PFCP utilise des délais d'attente configurables pour les demandes d'établissement, de modification et de suppression de session à l'UPF.

Paramètres :

Paramètre	Type	Par défaut	Description
`request_timeout_ms`	Entier	500	Temps en millisecondes à attendre pour la réponse de l'UPF avant de retransmettre
`request_attempts`	Entier	3	Nombre maximum de tentatives de transmission avant d'échouer la demande

Temps d'attente total : `request_timeout_ms × request_attempts`

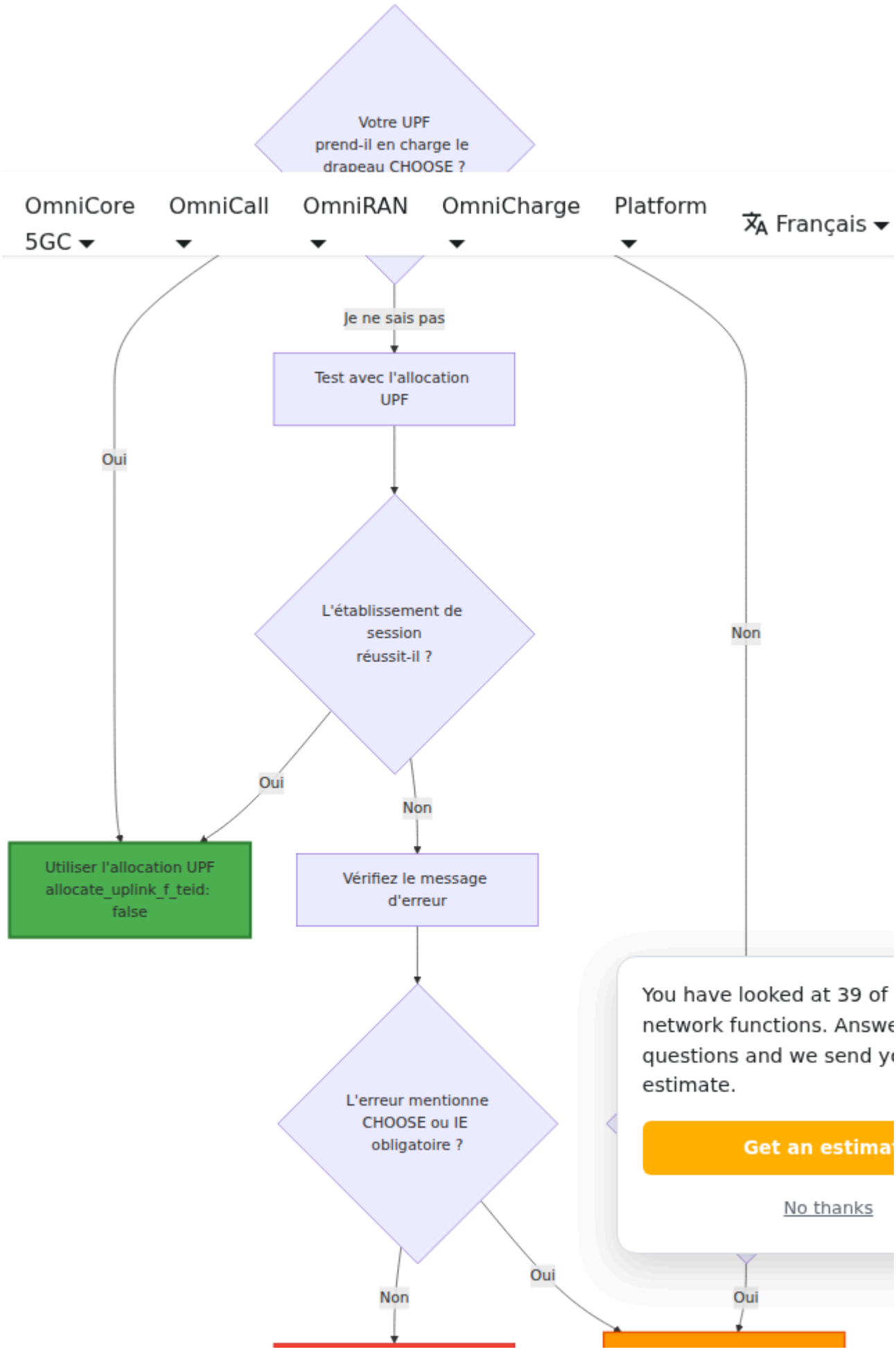
Comportement par défaut : 500ms × 3 tentatives = **1,5 secondes de temps d'attente maximum total**

Pourquoi la configuration du délai d'attente est importante

Si `request_timeout_ms` est défini trop bas par rapport au temps de traitement de l'UPF :

1. PGW-C envoie une demande d'établissement de session
2. Le délai d'attente expire avant que l'UPF ne réponde

3. PGW-C retransmet avec le même numéro de séquence
4. L'UPF traite les deux demandes et crée **des sessions PFCP dupliquées**
5. PGW-C reçoit la première réponse et stocke un ID de session
6. La deuxième session devient **orpheline** dans l'UPF


Votre UPF prend-il en charge le drapeau CHOOSE ?
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Français
Je ne sais pas
Test avec l'allocation UPF
Oui
L'établissement de session réussit-il ?
Non
Oui
Non
Utiliser l'allocation UPF allocate_uplink_f_teid: false
Vérifiez le message d'erreur
L'erreur mentionne CHOOSE ou IE obligatoire ?
Non
Oui
Oui
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Directives de réglage

Temps de réponse de l'UPF	`request_timeout_ms` recommandé	Temps d'attente total
Rapide (<100ms)	200-300ms	600-900ms (3 tentatives)
Normal (100-300ms)	500ms (par défaut)	1.5s (3 tentatives)
Lent (300-500ms)	750-1000ms	2.25-3s (3 tentatives)
Très lent (>500ms)	1500-2000ms	4.5-6s (3 tentatives)

Recommandation : Définir `request_timeout_ms` à au moins **2× le temps de réponse attendu de l'UPF** pour éviter les sessions orphelines induites par retransmission.

Exemple - UPF lent

```
sxb: %{
  local_ip_address: "10.0.0.20",
  request_timeout_ms: 1000,  # 1 seconde par tentative
  request_attempts: 3         # Total : 3 secondes max
}
```

Diagnostic des problèmes de délai d'attente

Symptômes d'un délai d'attente trop bas :

- L'UPF signale plus de sessions PFCP que prévu
- Les sessions orphelines s'accumulent dans l'UPF au fil du temps

- Messages de journal : `"Demande d'établissement de session a expiré"` suivis d'une session réussie

Comment diagnostiquer :

1. Vérifiez le compte de session de l'UPF via l'API ou l'interface de gestion de l'UPF
2. Comparez avec le compte de session actif de PGW-C
3. Si l'UPF a plus de sessions, des orphelins existent à partir de retransmissions

Résolution :

1. Augmentez `request_timeout_ms` pour dépasser le temps de réponse de l'UPF
2. Redémarrez PGW-C pour appliquer la nouvelle configuration
3. Effacez les sessions orphelines de l'UPF (nettoyage manuel ou redémarrage de l'UPF)

Plusieurs pairs PGW-U

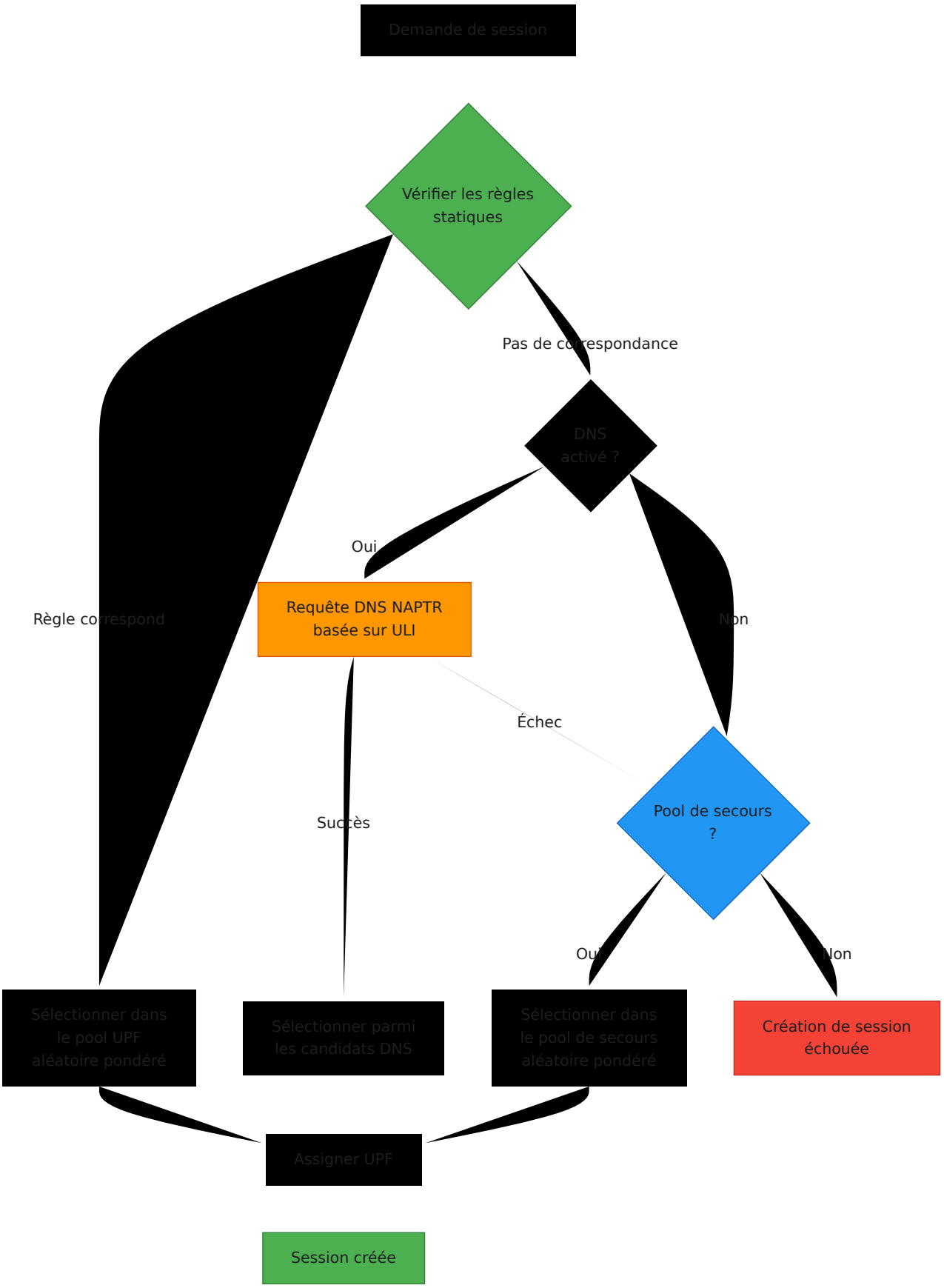
Pour l'équilibrage de charge ou la redondance :

```
config :pgw_c,
  sxb: %{
    local_ip_address: "10.0.0.20"
  },
  upf_selection: %{
    fallback_pool: [
      %{remote_ip_address: "10.0.1.21", remote_port: 8805, weight:
50},  # 50% du trafic
      %{remote_ip_address: "10.0.2.21", remote_port: 8805, weight:
50}   # 50% du trafic
    ]
  }
# Les deux UPFs sont automatiquement enregistrés avec des cœurs
battants de 5 secondes
```

Configuration de la sélection de l'UPF

PGW-C utilise un **système de sélection UPF à trois niveaux** avec des règles basées sur la priorité :

1. **Règles statiques** (priorité la plus élevée) - Correspondent aux attributs de session
2. **Sélection basée sur DNS** (priorité moyenne) - Routage conscient de la localisation via des requêtes DNS NAPTR
3. **Pool de secours** (priorité la plus basse) - Pool UPF par défaut lorsque aucune règle ne correspond



Exemple complet de sélection d'UPF

```
config :pgw_c,
  # Interface PFCP
  sxb: %{
    local_ip_address: "10.0.0.20"
  },

  # Sélection de l'UPF : Tous les UPFs définis ici sont
automatiquement enregistrés
  upf_selection: %{
    #
=========================================================================
    # Sélection basée sur DNS (Routage conscient de la localisation)
    #
=========================================================================
    # Interroge le DNS en utilisant les informations de localisation
de l'utilisateur (ULI)
    # Fournit une sélection dynamique de l'UPF en fonction de la
localisation de la cellule
    dns_enabled: false,
    dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],
    dns_suffix: "epc.3gppnetwork.org",
    dns_timeout_ms: 5000,

    #
=========================================================================
    # Règles de sélection statiques (évaluées par priorité)
    #
=========================================================================
    # Les règles sont vérifiées de la plus haute à la plus basse
priorité
    # La première règle correspondante détermine le pool UPF
    rules: [
      # Règle 1 : Trafic IMS - Priorité la plus élevée
      %{
        name: "Trafic IMS",
        priority: 20,
        match_field: :apn,
        match_regex: "^ims",
        upf_pool: [
          %{remote_ip_address: "10.100.2.21", remote_port: 8805,
weight: 80},
          %{remote_ip_address: "10.100.2.22", remote_port: 8805,
weight: 20}
```

```
        ],
        # Facultatif : Remplacements PCO pour cette règle
        pco: %{
          p_cscf_ipv4_address_list: ["203.0.113.100",
"203.0.113.101"]
        }
      },

      # Règle 2 : APN d'entreprise - Haute priorité
      %{
        name: "Trafic d'entreprise",
        priority: 15,
        match_field: :apn,
        match_regex: "^(enterprise|corporate)\.apn",
        upf_pool: [
          %{remote_ip_address: "10.100.3.21", remote_port: 8805,
weight: 100}
        ],
        pco: %{
          primary_dns_server_address: "192.168.1.10",
          secondary_dns_server_address: "192.168.1.11",
          ipv4_link_mtu_size: 1500
        }
      },

      # Règle 3 : Abonnés en itinérance - Priorité moyenne
      %{
        name: "Abonnés en itinérance",
        priority: 10,
        match_field: :serving_network_plmn_id,
        match_regex: "^(310|311|312|313)",  # Réseaux américains
        upf_pool: [
          %{remote_ip_address: "10.100.4.21", remote_port: 8805,
weight: 100}
        ]
      },

      # Règle 4 : Trafic Internet - Priorité inférieure
      %{
        name: "Trafic Internet",
        priority: 5,
        match_field: :apn,
        match_regex: "^internet",
        upf_pool: [
```

```
          %{remote_ip_address: "10.100.1.21", remote_port: 8805,
weight: 33},
          %{remote_ip_address: "10.100.1.22", remote_port: 8805,
weight: 33},
          %{remote_ip_address: "10.100.1.23", remote_port: 8805,
weight: 34}
        ]
      }
    ],

    #
=============================================================================
    # Pool de secours (dernier recours)
    #
=============================================================================
    # Utilisé lorsque aucune règle ne correspond et que la sélection
DNS échoue ou est désactivée
    fallback_pool: [
      %{remote_ip_address: "127.0.0.21", remote_port: 8805, weight:
100}
    ]
  }
```

Champs de correspondance pris en charge

Champ de correspondance	Description	Valeur exemple
`:imsi`	Identifiant international de l'abonné mobile	`"310260123456789"`
`:apn`	Nom de point d'accès	`"internet"`, `"ims"`
`:serving_network_plmn_id`	PLMN du réseau servant (MCC+MNC)	`"310260"` (réseau de test)
`:sgw_ip_address`	Adresse IP du SGW (format chaîne)	`"10.0.1.50"`
`:uli_tai_plmn_id`	ID PLMN de la zone de suivi	`"310260"`
`:uli_ecgi_plmn_id`	ID PLMN de la cellule E-UTRAN	`"310260"`

Pool UPF et équilibrage de charge

Chaque règle peut spécifier un **pool UPF** avec une sélection aléatoire pondérée :

```
upf_pool: [
  %{remote_ip_address: "10.100.1.21", remote_port: 8805, weight:
50},
  %{remote_ip_address: "10.100.1.22", remote_port: 8805, weight:
30},
  %{remote_ip_address: "10.100.1.23", remote_port: 8805, weight:
20}
]
```

Comment fonctionne la sélection pondérée :

1. Calculez le poids total : 50 + 30 + 20 = 100
2. Générez un nombre aléatoire : 0.0 à 100.0
3. Sélectionnez l'UPF en fonction des plages de poids cumulées :
    - 0-50 : UPF-1 (chance de 50 %)
    - 50-80 : UPF-2 (chance de 30 %)
    - 80-100 : UPF-3 (chance de 20 %)

Cas d'utilisation :

- **Distribution égale :** Tous les poids égaux (33, 33, 34)
- **Principal/secondaire :** Principal à poids élevé (80), secondaire à poids faible (20)
- **Basé sur la capacité :** Poids proportionnel à la capacité de l'UPF

Remplacements PCO

Les règles peuvent remplacer les valeurs PCO (Options de configuration de protocole) :

```
%{
  name: "Trafic IMS",
  match_field: :apn,
  match_regex: "^ims",
  upf_pool: [...],
  pco: %{
    # Remplacer uniquement certains champs
    p_cscf_ipv4_address_list: ["203.0.113.100", "203.0.113.101"],
    # Autres champs utilisent les valeurs par défaut de la
configuration principale de pco
  }
}
```

Champs de remplacement PCO disponibles :

- `primary_dns_server_address`
- `secondary_dns_server_address`
- `primary_nbns_server_address`
- `secondary_nbns_server_address`

- `p_cscf_ipv4_address_list`
- `ipv4_link_mtu_size`

Sélection basée sur DNS

Lorsqu'elle est activée, PGW-C effectue des requêtes DNS NAPTR basées sur les informations de localisation de l'utilisateur :

```
upf_selection: %{
  dns_enabled: true,
  dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],
  dns_suffix: "epc.3gppnetwork.org",
  dns_timeout_ms: 5000
}
```

Priorité de requête :

1. **ECGI** (Identifiant global de cellule E-UTRAN) - Routage au niveau de la cellule le plus spécifique
2. **TAI** (Identifiant de zone de suivi) - Zone de cellule
3. **RAI** (Identifiant de zone de routage) - Zone 3G/2G
4. **SAI** (Identifiant de zone de service) - Zone de service 3G
5. **CGI** (Identifiant global de cellule) - Cellule 2G

Exemple de requête DNS :

```
# Pour la requête ECGI :
eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org

# Pour la requête TAI :
tac-lb64.tac-hb00.tac.epc.mnc999.mcc999.epc.3gppnetwork.org
```

Processus de sélection DNS :

1. Essayer les requêtes dans l'ordre de priorité (ECGI d'abord, puis TAI, etc.)
2. Si le DNS renvoie des candidats, utiliser le premier résultat (enregistré dynamiquement si nécessaire)
3. Sélectionner l'UPF renvoyé

4. Si aucune correspondance DNS ou DNS désactivé, passer au pool de secours

Voir Sélection de l'UPF basée sur DNS pour des informations détaillées.

Sélection de l'UPF basée sur DNS

Aperçu

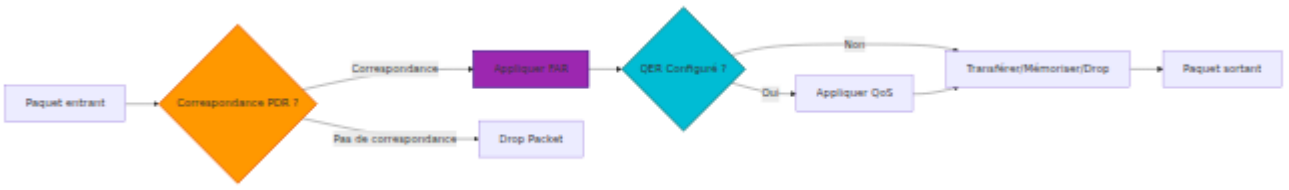
La sélection de l'UPF basée sur DNS fournit un **routage conscient de la localisation** en effectuant des requêtes DNS NAPTR en utilisant les informations de localisation de l'utilisateur (ULI) de la cellule actuelle de l'UE.

Référence 3GPP : TS 23.003 - Procédures DNS pour la découverte de l'UPF

Avantages :

- Sélection automatique de l'UPF en fonction de la localisation géographique
- Pas de configuration manuelle des règles par cellule
- Adaptation dynamique aux changements de topologie du réseau
- Réduit le backhaul en routant vers l'UPF le plus proche

Comment cela fonctionne



Configuration

```
config :pgw_c,
  upf_selection: %{
    # Activer la sélection basée sur DNS
    dns_enabled: true,

    # Priorité de requête : essayer ECGI d'abord, puis TAI, puis
RAI, etc.
    dns_query_priority: [:ecgi, :tai, :rai, :sai, :cgi],

    # Suffixe DNS pour les requêtes
    dns_suffix: "epc.3gppnetwork.org",

    # Délai d'attente de la requête DNS
    dns_timeout_ms: 5000,

    # Les règles statiques ont toujours la priorité sur DNS
    rules: [...],

    # Fallback si DNS échoue
    fallback_pool: [...]
  }
```

Formats de requête DNS

Les requêtes DNS sont construites en utilisant les informations de localisation de l'utilisateur (ULI) du message GTP-C :

1. ECGI (Identifiant global de cellule E-UTRAN)

Le plus spécifique - Routage au niveau de la cellule LTE

Format :

```
eci-<HEX-ECI>.ecgi.epc.mnc<MNC>.mcc<MCC>.<dns_suffix>
```

Exemple :

```
# ID de cellule : 0x1A2B3C (1 715 004 décimal)
# PLMN : MCC=999, MNC=999
eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org
```

Quand utilisé : Réseaux LTE (4G)

2. TAI (Identifiant de zone de suivi)

Zone de cellule - Plusieurs cellules dans la même zone de suivi

Format :

```
tac-lb<LB>.tac-hb<HB>.tac.epc.mnc<MNC>.mcc<MCC>.<dns_suffix>
```

Exemple :

```
# TAC : 0x0064 (100 décimal)
# Octet bas : 0x64, Octet haut : 0x00
tac-lb64.tac-hb00.tac.epc.mnc999.mcc999.epc.3gppnetwork.org
```

Quand utilisé : Zones de suivi LTE (4G)

3. RAI (Identifiant de zone de routage)

Zone de routage 3G/2G

Format :

```
rac<RAC>.lac-lb<LB>.lac-hb<HB>.lac.rai.mnc<MNC>.mcc<MCC>.
<dns_suffix>
```

Exemple :

```
# RAC : 0x0A (10 décimal)
# LAC : 0x1234 (4660 décimal)
rac0a.lac-lb34.lac-hb12.lac.rai.mnc999.mcc999.epc.3gppnetwork.org
```

Quand utilisé : Réseaux 3G/2G UMTS/GPRS

4. SAI (Identifiant de zone de service)

Zone de service 3G

Format :

```
sac<SAC>.lac-lb<LB>.lac-hb<HB>.lac.sai.mnc<MNC>.mcc<MCC>.
<dns_suffix>
```

Exemple :

```
# SAC : 0x0001
# LAC : 0x1234
sac0001.lac-lb34.lac-
hb12.lac.sai.mnc999.mcc999.epc.3gppnetwork.org
```

Quand utilisé : Zones de service 3G UMTS

5. CGI (Identifiant global de cellule)

Niveau de cellule 2G

Format :

```
ci<CI>.lac-lb<LB>.lac-hb<HB>.lac.cgi.mnc<MNC>.mcc<MCC>.
<dns_suffix>
```

Exemple :

```
# CI : 0x5678
# LAC : 0x1234
ci5678.lac-lb34.lac-hb12.lac.cgi.mnc999.mcc999.epc.3gppnetwork.org
```

Quand utilisé : Cellules 2G GSM

Traitement des réponses DNS

Format d'enregistrement NAPTR :

Le DNS renvoie des enregistrements NAPTR pointant vers les adresses IP de l'UPF :

```
eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org.
  IN NAPTR 10 50 "a" "x-3gpp-upf:x-s5-gtp:x-s8-gtp" ""
upf1.epc.mnc999.mcc999.3gppnetwork.org.

upf1.epc.mnc999.mcc999.3gppnetwork.org.
  IN A 10.100.1.21
```

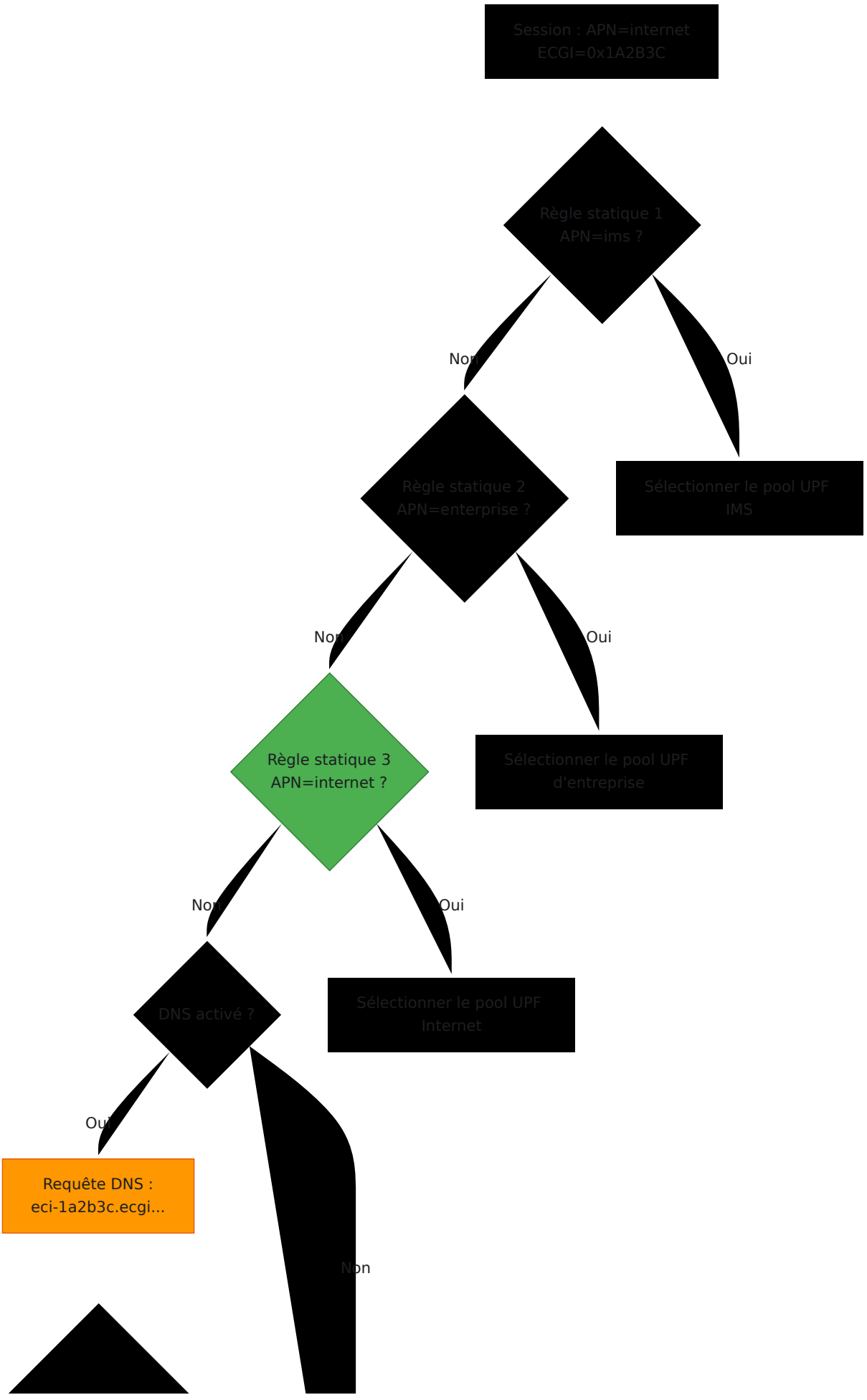
Traitement par PGW-C :

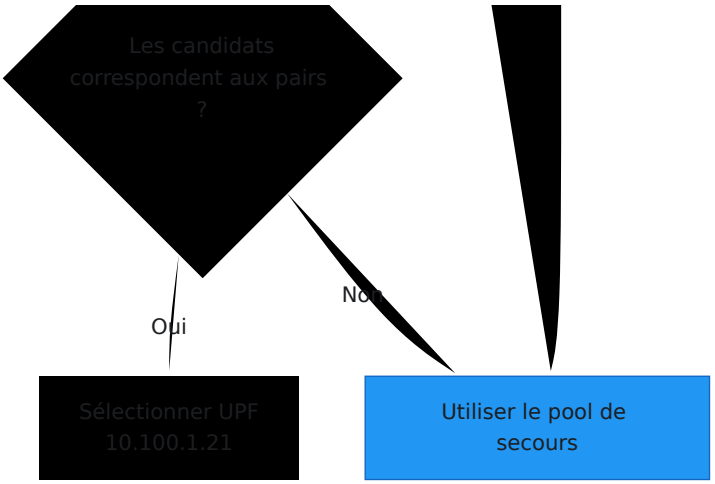
1. Analyser les enregistrements NAPTR pour extraire les adresses IP de l'UPF
2. Sélectionner le premier candidat de la réponse DNS
3. Enregistrer dynamiquement si pas déjà configuré (ou implémenter une sélection basée sur la charge)

Exemple :

```
Le DNS renvoie : [10.100.1.21, 10.100.5.99, 10.200.3.50]

Sélectionné :    10.100.1.21 (premier candidat)
Action :         Enregistrer dynamiquement si pas dans
upf_selection
```

Exemple de priorité de sélection


Session : APN=internet
ECGI=0x1A2B3C
Règle statique 1
APN=ims ?
Non
Oui
Sélectionner le pool UPF
IMS
Règle statique 2
APN=enterprise ?
Non
Oui
Sélectionner le pool UPF
d'entreprise
Règle statique 3
APN=internet ?
Non
Oui
Sélectionner le pool UPF
Internet
DNS activé ?
Oui
Requête DNS :
eci-1a2b3c.ecgi...
Non



Cas d'utilisation

1. Équilibrage de charge géographique

Scénario : L'opérateur a des UPFs dans plusieurs villes

Configuration DNS :

```
# Cellule de Chicago
eci-aaa.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF-Chicago
(10.1.1.21)

# Cellule de New York
eci-bbb.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF-NewYork
(10.2.1.21)

# Cellule de Los Angeles
eci-ccc.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF-
LosAngeles (10.3.1.21)
```

Avantage : Les utilisateurs sont automatiquement routés vers l'UPF le plus proche, réduisant la latence et le backhaul

2. Informatique de périphérie

Scénario : UPFs MEC (Multi-access Edge Computing) déployés sur les sites cellulaires

Configuration DNS :

```
# Chaque cellule pointe vers l'UPF de périphérie local
eci-*.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org → UPF de
périphérie local
```

Avantage : Latence ultra-faible pour les applications de périphérie

3. Topologie de réseau dynamique

Scénario : Les adresses UPF changent en raison de mises à jour ou de maintenance

Avantage : Mettre à jour les enregistrements DNS sans changer la configuration de PGW-C

Dépannage de la sélection DNS

Échecs de requête DNS

Symptômes :

- Journal : "Échec de la sélection UPF DNS : :nxdomain"
- Les sessions tombent dans le pool de secours

Causes possibles :

1. Serveur DNS mal configuré
2. Zone DNS non peuplée pour les ID de cellule
3. ULI non présent dans le message GTP-C

Résolution :

```
# Tester la requête DNS manuellement
dig eci-1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org NAPTR

# Vérifier les journaux de PGW-C pour les requêtes DNS
grep "DNS UPF selection: querying" /var/log/pgw_c.log

# Vérifier la présence de l'ULI dans la session
# Vérifier le champ "uli" dans l'état de la session
```

DNS renvoie un UPF inconnu

Comportement :

- Le DNS renvoie un candidat UPF qui n'est pas dans `upf_selection`
- Le système tente automatiquement l'enregistrement dynamique
- Si l'association PFCP réussit, l'UPF est utilisé pour la session
- Si l'association PFCP échoue, il tombe dans le pool de secours

Exemple :

```
Le DNS renvoie : [10.99.1.50]
upf_selection : [10.100.1.21, 10.100.1.22]

Action : Enregistrer dynamiquement 10.99.1.50
  - Envoyer la configuration d'association PFCP
  - Si succès : Utiliser pour la session
  - Si délai d'attente : Tomber dans le pool de secours
```

Options de résolution :

1. Pré-configurer dans `upf_selection` pour un suivi immédiat :

```
upf_selection: %{
  fallback_pool: [
    %{remote_ip_address: "10.99.1.50", remote_port: 8805, weight:
100}
  ]
}
```

2. Mettre à jour le DNS pour renvoyer les adresses IP d'UPF pré-configurées

3. Autoriser l'enregistrement dynamique (recommandé pour les scénarios MEC/périphérie)

Délai d'attente de requête

Symptômes :

- Journal : "Sélection UPF DNS : délai d'attente de requête"
- Les sessions prennent plus de temps à s'établir

Résolution :

```
upf_selection: %{
  dns_timeout_ms: 10000  # Augmenter le délai d'attente à 10
secondes
}
```

Surveillance de la sélection DNS

Métriques :

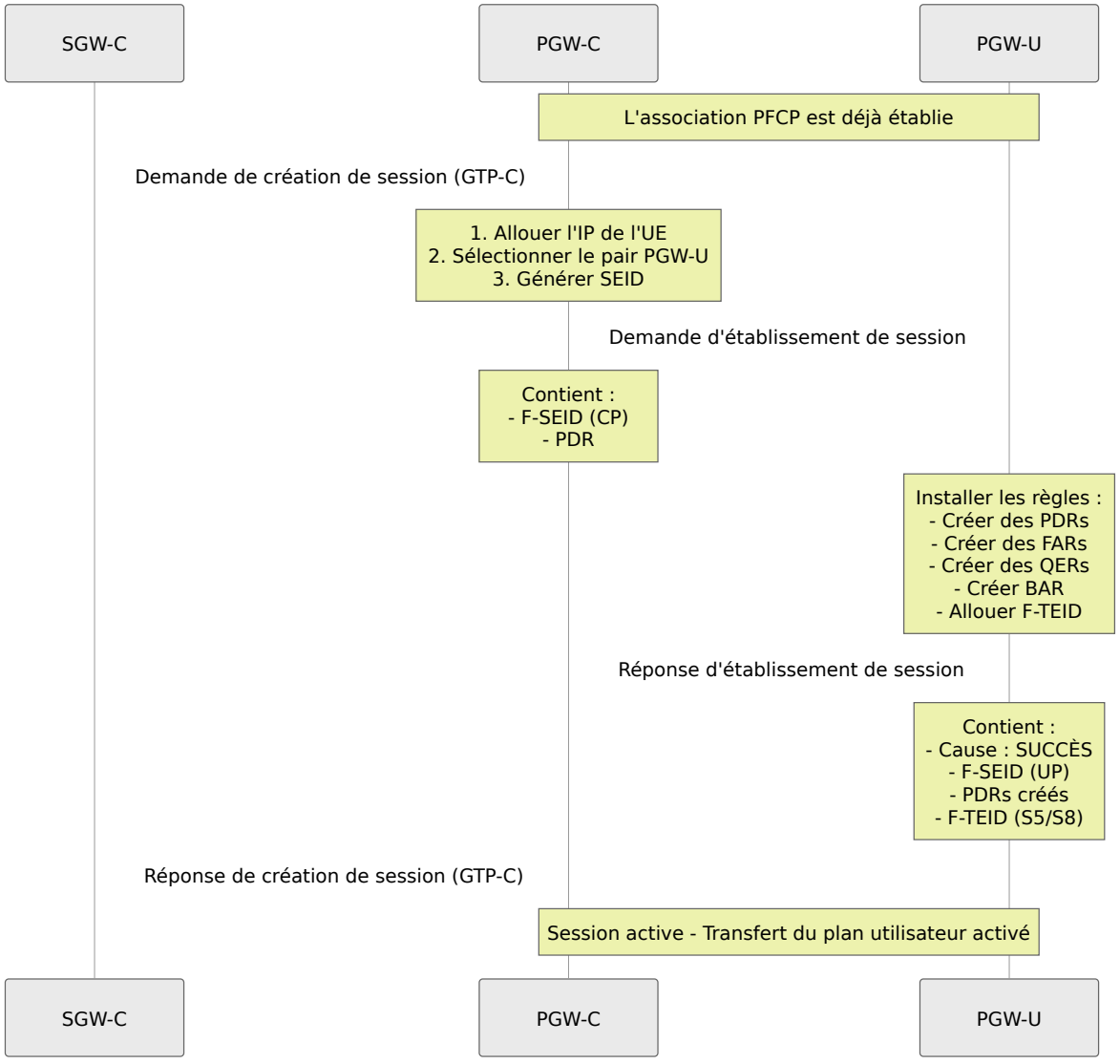
```
# Taux de réussite des requêtes DNS
rate(upf_selection_dns_success_total[5m]) /
rate(upf_selection_dns_attempts_total[5m])

# Latence des requêtes DNS
histogram_quantile(0.95,
rate(upf_selection_dns_duration_seconds_bucket[5m]))

# Utilisation du fallback (indique des problèmes DNS)
rate(upf_selection_fallback_used_total[5m])
```

Exemples de journaux :

```
[debug] DNS UPF selection: querying eci-
1a2b3c.ecgi.epc.mnc999.mcc999.epc.3gppnetwork.org
[debug] DNS UPF selection: got 2 candidates from DNS
[info]  DNS UPF selection: selected 10.100.1.21
```

Flux de messages

Flux complet d'établissement de session

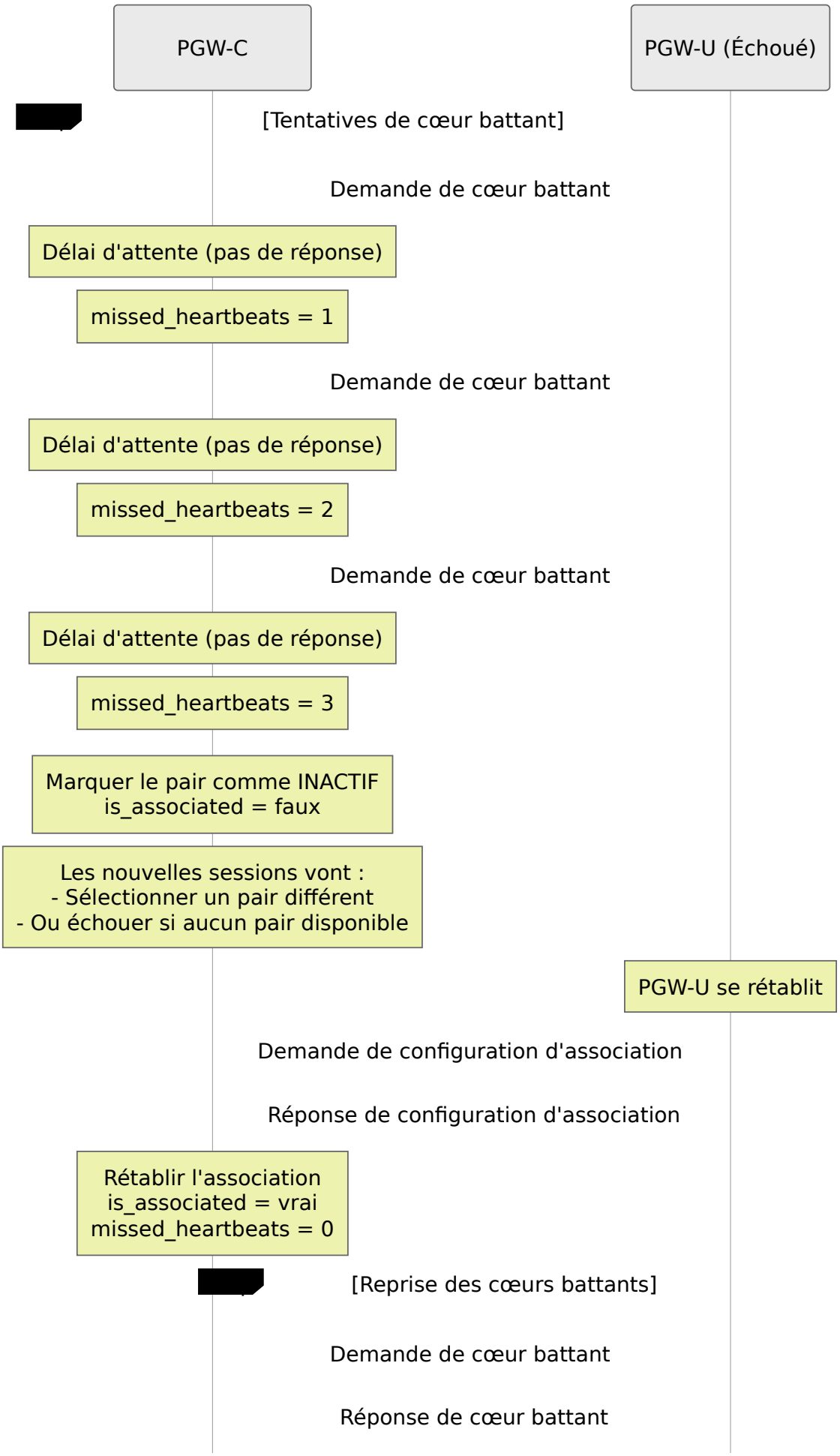


Flux de modification de session



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

Récupération après échec de cœur battant


PGW-C
PGW-U (Échoué)
[Tentatives de cœur battant]
Demande de cœur battant
Délai d'attente (pas de réponse)
missed_heartbeats = 1
Demande de cœur battant
Délai d'attente (pas de réponse)
missed_heartbeats = 2
Demande de cœur battant
Délai d'attente (pas de réponse)
missed_heartbeats = 3
Marquer le pair comme INACTIF
is_associated = faux
Les nouvelles sessions vont :
- Sélectionner un pair différent
- Ou échouer si aucun pair disponible
PGW-U se rétablit
Demande de configuration d'association
Réponse de configuration d'association
Rétablir l'association
is_associated = vrai
missed_heartbeats = 0
[Reprise des cœurs battants]
Demande de cœur battant
Réponse de cœur battant



Dépannage

Problèmes courants

1. Échec de la configuration de l'association

Symptômes :

- Message journal : "Échec de la configuration de l'association PFCP"
- Pas de réponse à la demande de configuration de l'association

Causes possibles :

- PGW-U non accessible (problème réseau)
- PGW-U non en cours d'exécution
- Pare-feu bloquant le port UDP 8805
- ID `remote_ip_address` incorrect dans la configuration

Résolution :

```
# Tester la connectivité
ping <pgw_u_ip_address>

# Tester le port UDP
nc -u -v <pgw_u_ip_address> 8805

# Vérifier le pare-feu
iptables -L -n | grep 8805
```

2. Échecs de cœurs battants

Symptômes :

- Journal : "Échecs consécutifs de cœur battant : 3"
- Association marquée comme inactive

Causes possibles :

- Latence réseau ou perte de paquets
- PGW-U surchargé
- Intervalle de cœur battant trop agressif

Résolution :

La période de cœur battant est fixée à 5 secondes avec un seuil d'échec de 3 cœurs battants manqués consécutifs.

3. Échec de l'établissement de session

Symptômes :

- Réponse de création de session avec code d'erreur
- Journal : "Échec de l'établissement de session PFCP"

Causes possibles :

- Aucun pair PGW-U disponible
- Épuisement des ressources PGW-U
- Configuration de règle invalide

Vérifiez :

1. Vérifiez qu'au moins un pair a `is_associated = true`
2. Vérifiez les journaux de PGW-U pour des erreurs
3. Vérifiez l'unicité de SEID

4. Erreurs de SEID dupliqué

Symptômes :

- Réponse d'établissement de session : Cause "Contexte de session non trouvé"

Cause :

- Collision de SEID (très rare)
- Redémarrage de PGW-U sans connaissance de PGW-C

Résolution :

- Redémarrer l'association PFCP (déclenche un nouvel horodatage de récupération)
- PGW-C détectera le redémarrage de PGW-U et nettoiera les anciennes sessions

Surveillance de la santé PFCP

Métriques à surveiller :

```
# État d'association des pairs PFCP
pfcp_peer_associated{peer="PGW-U Principal"} 1

# Sessions PFCP actives
seid_registry_count 150

# Taux de messages PFCP
rate(sxb_inbound_messages_total[5m])

# Erreurs PFCP
rate(sxb_inbound_errors_total[5m])

# Échecs de cœur battant
pfcp_consecutive_heartbeat_failures{peer="PGW-U Principal"} 0
```

Exemples d'alerte :

```
# Alerte sur association inactive
- alert: PFCPAssociationDown
  expr: pfcp_peer_associated == 0
  for: 1m
  annotations:
    summary: "Le pair PFCP {{ $labels.peer }} est hors ligne"

# Alerte sur un taux élevé d'échecs d'établissement de session
- alert: PFCPSessionEstablishmentFailureHigh
  expr:
rate(sxb_inbound_errors_total{message_type="session_establishment_res
[5m]) > 0.1
  for: 5m
  annotations:
    summary: "Taux élevé d'échecs d'établissement de session PFCP"
```

Surveillance PFCP

L'association PFCP/Sxb et l'état des pairs UPF sont surveillés via l'API REST OAM. L'API est servie sur HTTPS/TLS sur le port `8443` avec toutes les routes sous le préfixe `/api` . La documentation interactive (Swagger UI) est disponible à `https://<omnipgw-ip>:8443/api/docs`.

État des UPF / pairs PFCP

```
GET /api/upf            # tous les pairs UPF / PFCP + état
GET /api/upf/<ip>       # un UPF par IP
```

But : Surveiller l'état d'association PFCP avec tous les PGW-U configurés.

Exemple :

```
curl -k https://<omnipgw-ip>:8443/api/upf
curl -k https://<omnipgw-ip>:8443/api/upf/10.0.0.1
```

Informations par pair renvoyées pour chaque PGW-U configuré :

Champ	Description
Nom du pair	Nom convivial de la configuration
Adresse IP	IP distante de PGW-U
État de l'association	Associé ou Non associé
ID de nœud	Identifiant de nœud PFCP
Horodatage de récupération	Dernière heure de redémarrage du pair
Période de cœur battant	Intervalle de cœur battant configuré
Cœurs battants manqués consécutifs	Compte actuel des échecs
Fonctionnalités de fonction UP	Capacités annoncées par PGW-U

`GET /api/upf/<ip>` renvoie l'état complet pour un seul pair, y compris la carte des fonctionnalités de fonction UP et les horodatages d'association.

Inspection de session PFCP

Les sessions PFCP actives sont visibles via le détail de session par abonné :

```
GET /api/sessions/<imsi>
```

Le détail de la session inclut le pair PGW-U associé et les règles de transfert programmées (PDRs, FARs, QERs, BARs) et F-SEIDs pour la session.

Cas d'utilisation opérationnels

Surveiller la santé de l'association PFCP :

```
1. GET /api/upf
2. Vérifiez que tous les pairs affichent "Associé"
3. Vérifiez que le compte de cœurs battants manqués = 0
4. Si un pair affiche "Non associé" :
   - Vérifiez l'accessibilité IP du pair
   - Vérifiez que le pair est en cours d'exécution
   - Vérifiez le pare-feu (UDP 8805)
```

Dépanner les échecs d'établissement de session :

```
1. La session utilisateur échoue à s'établir
2. GET /api/sessions/<imsi> - la session existe-t-elle ?
3. Inspecter l'état PFCP de la session - session PFCP créée ?
4. Si aucune session PFCP :
   - GET /api/upf - un pair est-il associé ?
   - Vérifiez les journaux pour des erreurs PFCP
5. Si la session PFCP existe :
   - Inspecter les PDRs/FARs pour vérifier les règles programmées
   - Le problème est probablement en aval (PGW-U ou réseau)
```

Vérifier la distribution de charge des pairs :

```
1. Avec plusieurs pairs PGW-U configurés
2. GET /api/upf pour examiner l'état de chaque pair
3. Vérifiez que les sessions sont réparties entre les pairs
4. Identifier si un pair a une charge disproportionnée
```

Détecter les pannes de pairs :

- `GET /api/upf` signale l'état "Non associé" par pair
- Le compteur de cœurs battants manqués montre une dégradation avant une défaillance totale
- Configurer des alertes de surveillance basées sur les données de l'API OAM

Avantages :

- **Surveillance scriptable** - Interroger `GET /api/upf` à partir d'outils ou d'alertes
- **État explicite** - Associé / non associé par pair
- **Tendances de santé des pairs** - Le compte de cœurs battants manqués montre un avertissement précoce
- **Inspection au niveau de la session** - Voir exactement les PDRs/FARs/QERs programmés

---

Documentation connexe

Configuration

- **Guide de configuration** - Sélection de l'UPF, surveillance de la santé, configuration PFCP
- **Gestion des sessions** - Cycle de vie des sessions PDN, établissement de bearer

Facturation et surveillance

- **Interface Diameter Gx** - Règles PCC qui pilotent l'application de QoS PFCP
- **Interface Diameter Gy** - Gestion des quotas de facturation en ligne via URRs
- **Format de CDR de données** - Génération de CDR à partir des rapports d'utilisation PFCP
- **Guide de surveillance** - Métriques PFCP, suivi des sessions, alertes de santé UPF

Interfaces réseau

- **Interface S5/S8** - Gestion des bearers du plan de contrôle
- **Allocation IP UE** - Attribution d'adresse UE via PFCP

Retour au guide des opérations

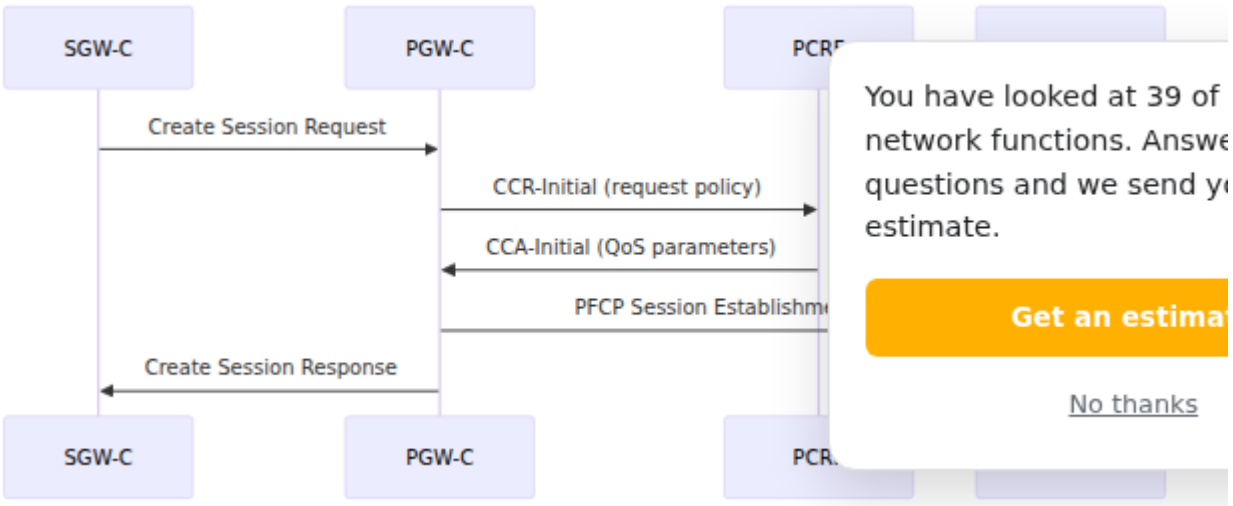
Gestion de la QoS et des Bearers

Vue d'ensemble

Le PGW-C met en œuvre un système de gestion des bearers et de la QoS basé sur des politiques qui coordonne trois interfaces clés :

- **Gx (Diameter)** - Reçoit les décisions de politique et les paramètres de QoS du PCRF
- **S5/S8 (GTP-C)** - Gère les contextes de bearer avec le SGW-C
- **Sxb (PFCP)** - Programme les règles d'application de la QoS dans le PGW-U

Flux d'architecture



Concepts clés

- **Session** : Contient des informations sur l'UE, une carte des bearers, des cartes PDR/FAR/QER/BAR, et AMBR
- **Contexte de Bearer** : Lie l'EBI (EPS Bearer ID) à des PDR, FAR et QER spécifiques

- **QER (Règle d'application de la QoS)** : Applique les limites MBR/GBR et l'état de la porte dans le plan utilisateur
- **Bearer par défaut** : Toujours créé avec la session PDN, fournit une connectivité de base
- **Bearer dédié** : Créé dynamiquement en fonction de la politique du PCRF, fournit des garanties de QoS spécifiques

Configuration

Important : Politique de QoS dynamique

Tous les paramètres de QoS sont reçus dynamiquement du PCRF via l'interface Diameter Gx et définis dans le PCRF (voir OmniHSS pour plus d'infos).

Les opérateurs configurent la **connexion PCRF** dans `config/runtime.exs` :

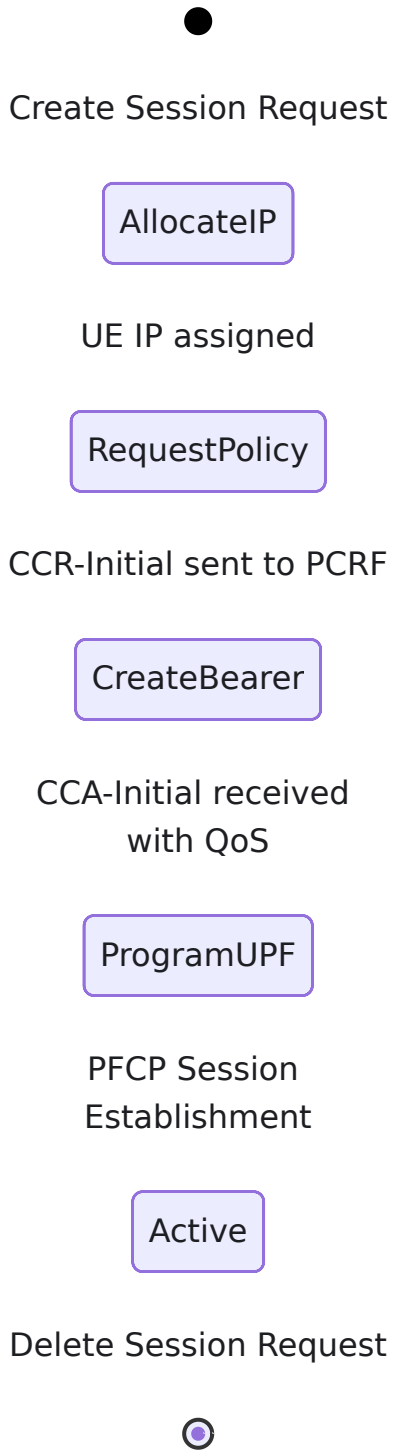
```
config :pgw_c,
  diameter: %{
    listen_ip: "0.0.0.0",
    host: "omni-pgw_c.epc.mnc999.mcc999.3gppnetwork.org",
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "192.168.1.100",
        initiate_connection: true
      }
    ]
  }
```

Les politiques de QoS, les règles de facturation et les limites de bande passante sont configurées sur le PCRF, pas dans les fichiers de configuration du PGW-C.

Cycle de vie du Bearer

Création du Bearer par défaut

Le bearer par défaut est créé lors de l'établissement de la session PDN :



Flux de travail :

1. SGW-C envoie une demande de création de session
2. PGW-C alloue une adresse IP UE à partir du pool configuré
3. PGW-C envoie CCR-Initial au PCRF avec IMSI, APN, adresse IP
4. Le PCRF répond avec CCA-Initial contenant les paramètres de QoS :
    - Default-EPS-Bearer-QoS (QCI, ARP)
    - QoS-Information (ajustements AMBR)
5. PGW-C crée le contexte de bearer avec :
    - Identifiants fixes : PDR descendant=1, PDR montant=2, FAR descendant=1, FAR montant=2, QER=1, BAR=1
    - QER programmé avec MBR à partir de la QoS du bearer
6. PGW-C envoie une demande d'établissement de session PFCP au PGW-U
7. PGW-C envoie une réponse de création de session au SGW-C

Caractéristiques du bearer par défaut :

- Existe toujours pendant la durée de la session PDN
- Utilise généralement QCI 5 ou QCI 9 (non-GBR)
- EBI suivi dans l'état de la session
- Ne peut pas être supprimé indépendamment (sa suppression termine la session)

Création du Bearer dédié

Les bearers dédiés sont créés dynamiquement en fonction de la politique du PCRF :

Déclencheur : Re-Auth Request (RAR) du PCRF avec Charging-Rule-Install

Flux de travail :

1. Le PCRF envoie un RAR avec une définition de règle de facturation contenant :
    - Charging-Rule-Name (identifiant de la règle de politique)
    - Flow-Information (filtres de paquets)
    - QoS-Information (QCI, MBR, GBR, ARP)
    - Precedence (priorité de correspondance de la règle)

2. PGW-C traduit la règle dynamique en entités PFCP :
    - Chaque entrée de Flow-Information → nouveau PDR avec filtre SDF
    - QoS-Information → nouveau QER avec application MBR/GBR
    - Flow-Description → règles de correspondance IP 5-tuple
3. PGW-C envoie une demande de modification de session PFCP pour ajouter des PDR/FAR/QER
4. PGW-C initie une demande de création de bearer au SGW-C
5. SGW-C répond avec une réponse de création de bearer confirmant l'établissement

Exemple de définition de règle de facturation :

```
Charging-Rule-Name: "video_streaming"
Flow-Information:
  - Flow-Description: "permit in ip from any to 10.0.0.1 5000-
6000"
    Flow-Direction: 1 (downlink)
QoS-Information:
  QoS-Class-Identifier: 7
  Max-Requested-Bandwidth-UL: 5000000 (5 Mbps)
  Max-Requested-Bandwidth-DL: 10000000 (10 Mbps)
  Guaranteed-Bitrate-UL: 1000000 (1 Mbps)
  Guaranteed-Bitrate-DL: 2000000 (2 Mbps)
Precedence: 100
Flow-Status: 2 (ENABLED)
```

Règles liées au Bearer par défaut

Une règle dynamique dont le QCI et l'ARP correspondent au bearer par défaut se lie au bearer par défaut (TS 29.212 §4.5.2). Aucun bearer dédié n'est créé.

Lorsque la règle correspond à tout le trafic (chaque Flow-Description est `permit out ip from any to any`), son MBR limite l'APN-AMBR sur le QER du bearer par défaut :

- Lors de l'installation (CCA-Initial ou RAR), le MBR du QER par défaut devient le plus bas entre l'APN-AMBR et le MBR de la règle, par direction. Une règle

de la CCA-Initial est déjà dans le QER que l'établissement de session transporte.
- Lors de la suppression de la règle de facturation, le QER revient à l'APN-AMBR.
- Un changement ultérieur de l'APN-AMBR du PCRF maintient la limite tant que la règle est installée.

C'est ainsi qu'un throttling d'utilisation équitable fonctionne avec Sy : le PCRF installe une règle telle que `throttle` (QCI 9, MBR 1 Mbit/s) lorsque l'allocation est épuisée, et la retire après un rechargement.

Une règle de bearer par défaut qui ne correspond qu'à une partie du trafic, ou qui contrôle une direction avec Flow-Status, aurait besoin de ses propres PDR et n'est pas prise en charge. Un RAR portant une telle règle est répondu avec DIAMETER_UNABLE_TO_COMPLY (5012). Lors de la CCA-Initial, elle est enregistrée et non appliquée.

Modification du Bearer

La QoS du bearer peut être modifiée via :

- **Gx RAR** avec une définition de règle de facturation mise à jour
- **Modification de session PFCP** pour mettre à jour les QER existants (changer les débits), les FAR (changer le transfert), ou les PDR (changer les filtres de paquets)

Suppression du Bearer

Déclencheurs :

- **Delete Session Request** (initiée par SGW) - Supprime le bearer par défaut et termine la session
- **Re-Auth Request avec Charging-Rule-Remove** (initiée par PCRF) - Supprime le bearer dédié

Flux de travail :

1. Supprimer le bearer de l'état de la session

2. Supprimer les PDR/FAR/QER associés
3. Envoyer une demande de suppression de bearer au SGW-C (si initiée par PCRF)
4. Envoyer une modification de session PFCP (supprimer les règles) ou une suppression de session (si bearer par défaut)

Paramètres de QoS

QCI (Identifiant de Classe de QoS)

Source : PCRF via Gx `QoS-Class-Identifier` AVP

Valeurs standard :

- **QCI 1** : Voix conversationnelle (GBR, budget de délai de 100 ms)
- **QCI 2** : Vidéo conversationnelle (GBR, budget de délai de 150 ms)
- **QCI 3** : Jeux en temps réel (GBR, budget de délai de 50 ms)
- **QCI 4** : Vidéo non conversationnelle (GBR, budget de délai de 300 ms)
- **QCI 5** : Signalisation IMS (non-GBR, budget de délai de 100 ms) - **Par défaut pour le bearer par défaut**
- **QCI 6** : Vidéo (basée sur TCP), Streaming en direct (non-GBR, budget de délai de 300 ms)
- **QCI 7** : Voix, Jeux interactifs (non-GBR, budget de délai de 100 ms)
- **QCI 8** : Vidéo (basée sur TCP), par exemple, YouTube (non-GBR, budget de délai de 300 ms)
- **QCI 9** : Internet par défaut (non-GBR, budget de délai de 300 ms)

Remarque de l'opérateur :

- Le QCI est reçu du PCRF et signalé au SGW-C dans l'IE Bearer-Level-QoS
- PGW-C n'applique pas directement le comportement du QCI - l'application réelle se fait via MBR/GBR dans les QER
- Des valeurs de QCI plus faibles indiquent généralement une priorité plus élevée

- Le QCI détermine le traitement de transfert de paquets et la priorité de planification

ARP (Priorité d'Allocation et de Rétention)

Source : PCRF via `Allocation-Retention-Priority` AVP groupé

Composants :

- **Priority-Level** : 1 (priorité la plus élevée) à 15 (priorité la plus basse)
- **Pre-emption-Capability** : Ce bearer peut-il préempter des bearers de moindre priorité ?
  - 0 = ENABLED (peut préempter d'autres)
  - 1 = DISABLED (ne peut pas préempter)
- **Pre-emption-Vulnerability** : Ce bearer peut-il être préempté par des bearers de plus haute priorité ?
  - 0 = ENABLED (peut être préempté)
  - 1 = DISABLED (ne peut pas être préempté)

Valeurs par défaut :

- Priority-Level : 1
- Pre-emption-Capability : ENABLED (0)
- Pre-emption-Vulnerability : DISABLED (1)

Remarque de l'opérateur :

- L'ARP est signalé au SGW-C et finalement à l'eNodeB
- **Non appliqué par PGW-C** - l'application se fait généralement au niveau de l'eNodeB lors du contrôle d'admission radio
- Utilisé lors de la congestion du réseau pour déterminer quels bearers admettre ou supprimer
- Critique pour les services d'urgence (niveau de priorité 1) et les services de grande valeur

MBR (Débit Maximum)

Source : PCRF via `Max-Requested-Bandwidth-UL` et `Max-Requested-Bandwidth-DL` AVPs

Format : Octets par seconde (convertis en kbps en interne : `bytes / 1000`)

Appliqué à : Tous les bearers (par défaut et dédiés)

Comment ça fonctionne :

- PGW-C crée un QER avec `mbr: %Bitrate{ul: kbps_ul, dl: kbps_dl}`
- QER envoyé au PGW-U via PFCP
- **PGW-U applique la limitation de débit** (police de trafic)
- Le trafic excessif au-dessus du MBR est supprimé

Exemple :

```
Max-Requested-Bandwidth-UL: 5000000 (5 Mbps)
Max-Requested-Bandwidth-DL: 10000000 (10 Mbps)

→ QER créé avec mbr: {ul: 5000, dl: 10000} kbps
→ PGW-U supprime les paquets montants dépassant 5 Mbps
→ PGW-U supprime les paquets descendants dépassant 10 Mbps
```

GBR (Débit Garanti)

Source : PCRF via `Guaranteed-Bitrate-UL` et `Guaranteed-Bitrate-DL` AVPs

Format : Octets par seconde (convertis en kbps)

Appliqué à : Bearers dédiés uniquement (bearers GBR)

Comment ça fonctionne :

- Si le GBR est spécifié dans la définition de la règle de facturation, le bearer est de **type GBR**
- PGW-U applique une garantie de débit minimum via QER

- Nécessite une planification appropriée au niveau de l'eNodeB pour réserver des ressources radio
- Les bearers GBR ont un contrôle d'admission - peuvent être rejetés si les ressources ne sont pas disponibles

Exemple :

```
Guaranteed-Bitrate-UL: 1000000 (1 Mbps)
Guaranteed-Bitrate-DL: 2000000 (2 Mbps)

→ QER créé avec gbr: {ul: 1000, dl: 2000} kbps
→ Le réseau garantit au moins 1 Mbps en montant et 2 Mbps en
descendant
→ Utilisé pour VoIP, appels vidéo, streaming en direct
```

Remarque de l'opérateur :

- Le GBR nécessite une planification de capacité réseau suffisante
- Une surabonnement des ressources GBR entraîne des échecs d'admission
- Surveillez l'utilisation du GBR via les comptes de session et les métriques de bearer

AMBR (Débit Maximum Agrégé)

Source : PCRF via `APN-Aggregate-Max-Bitrate-UL` et `APN-Aggregate-Max-Bitrate-DL` AVPs

Portée : S'applique à **tous les bearers non-GBR** pour l'APN (pas par bearer)

Comment ça fonctionne :

- L'AMBR est une limite agrégée sur tous les bearers non-GBR dans une session
- Envoyé au SGW-C dans la réponse de création de session
- Application généralement au niveau de l'eNodeB/SGW
- PGW-C stocke l'AMBR dans l'état de la session et le signale au SGW-C

Exemple :

```
APN-Aggregate-Max-Bitrate-UL: 50000000 (50 Mbps)
APN-Aggregate-Max-Bitrate-DL: 100000000 (100 Mbps)

→ Tous les bearers non-GBR combinés ne peuvent pas dépasser 50
Mbps en montant / 100 Mbps en descendant
→ Les bearers individuels sont limités par leur propre MBR
→ L'AMBR fournit une limite supplémentaire globale par UE/APN
```

Remarque de l'opérateur :

- Défini via le profil d'abonné dans HSS/PCRF
- Utilisé pour appliquer des niveaux d'abonnement (par exemple, plan de 10 Mbps contre plan de 100 Mbps)
- N'affecte pas les bearers GBR

État du Flux et Contrôle

Mapping de l'État du Flux (Gx) à l'État de la Porte (PFCP)

Le PCRF contrôle si le trafic est autorisé via l'AVP `Flow-Status` dans la définition de la règle de facturation :

Flow-Status (Gx)	Gate-Status (PFCP QER)	Signification
0 = ENABLED-UPLINK	ul: OPEN, dl: CLOSED	Seul le trafic montant est autorisé
1 = ENABLED-DOWNLINK	ul: CLOSED, dl: OPEN	Seul le trafic descendant est autorisé
2 = ENABLED	ul: OPEN, dl: OPEN	Les deux directions sont autorisées
3 = DISABLED	ul: CLOSED, dl: CLOSED	Aucun trafic autorisé
4 = REMOVED	ul: CLOSED, dl: CLOSED	Bearer en cours de suppression

Cas d'utilisation :

- **DISABLED** : Utilisé pour les services en attente ou l'épuisement de crédit (paquets supprimés mais bearer conservé)
- **ENABLED-UPLINK** : Inhabituel, mais pourrait être utilisé pour des services uniquement de téléchargement
- **ENABLED-DOWNLINK** : Services uniquement de téléchargement ou scénarios limités par le crédit
- **ENABLED** : Fonctionnement normal

Surveillance et Observabilité

Métriques Prometheus

Métriques au niveau de la session :

```
session_registry_count        # Bearers actifs (paires IMSI, EBI)
address_registry_count        # IPs UE allouées
charging_id_registry_count    # Sessions de facturation actives
```

Métriques de l'interface Gx :

```
gx_inbound_messages_total{message_type="gx_RAR"}     # Mises à
jour de politique du PCRF
gx_outbound_messages_total{message_type="gx_CCR"}    # Demandes de
politique au PCRF
gx_outbound_transaction_duration_bucket              # Latence
vers le PCRF
```

Métriques de l'interface PFCP :

```
sxb_outbound_messages_total{message_type="pfcp_session_establishment_
sxb_outbound_messages_total{message_type="pfcp_session_modification_r
sxb_outbound_transaction_duration_bucket
```

Métriques de création de bearer :

```
s5s8_inbound_messages_total{message_type="create_session_request"}
# Bearers par défaut
s5s8_outbound_messages_total{message_type="create_bearer_request"}
# Bearers dédiés
```

Surveillance via l'API OAM

L'état des bearers et de la QoS est inspecté via l'API REST OAM (HTTPS/TLS sur le port `8443`, routes sous le préfixe `/api`). La documentation interactive (Swagger UI) est disponible à `https://<omnipgw-ip>:8443/api/docs`.

Les informations sur les bearers et la QoS sont exposées via les détails de session par abonné :

```
GET /api/sessions/<imsi>       # détails de la session incl.
bearers, QCI/5QI, règles de facturation
```

Exemple :

```
curl -k https://<omnipgw-ip>:8443/api/sessions/999000123456789
```

Les détails de la session incluent les bearers actifs pour la session, leurs paramètres de QoS (QCI/5QI, MBR, GBR, AMBR), les règles de facturation installées, et la carte QER programmée (état de la porte et débits). Il n'y a pas de point de terminaison pour les bearers dédiés ; l'état du bearer est toujours lu à partir de la session à laquelle il appartient.

Messages de journalisation clés

```
[debug] Sending Credit Control Request: ...          # CCR au PCRF
[debug] Handling Credit Control Answer: ...          # CCA du PCRF
(contient la QoS)
[debug] Handling Re-Auth Request                     # RAR du PCRF
(changement de politique)
[debug] Sending Session Establishment Request        # PFCP au
PGW-U (programmer les QER)
[debug] Sending Session Modification Request         # PFCP au
PGW-U (mettre à jour les QER)
```

Tâches opérationnelles

Vérifier la QoS appliquée à la session

1. `GET /api/sessions/<imsi>` (par exemple, `curl -k https://<omnipgw-ip>:8443/api/sessions/999000123456789`)
2. Localisez la session pour l'abonné dans la réponse
3. Inspectez les détails de la session
4. Vérifiez la section **qer_map** :

```
qer_id: 1
gate_status: {ul: OPEN, dl: OPEN}
mbr: {ul: 50000, dl: 100000}  # kbps
gbr: {ul: 10000, dl: 20000}   # kbps (ou nil pour non-GBR)
```

5. Vérifiez que les valeurs correspondent à la politique attendue du PCRF

Dépanner la QoS manquante

Symptôme : Session créée mais QoS non appliquée

Étapes :

1. **Vérifiez la connectivité PCRF :**
   - Vérifiez que l'état du pair PCRF = "connecté" (voir les métriques Diameter/Gx et l'état du pair)
   - Si déconnecté, vérifiez la connectivité réseau et la configuration Diameter
2. **Vérifiez l'échange CCR/CCA :**
   - Recherchez dans les journaux d'OmniPGW "Credit Control Answer"
   - Vérifiez que l'AVP `QoS-Information` est présent dans le journal CCA
   - Vérifiez les erreurs dans la CCA (Result-Code devrait être 2001 = SUCCESS)
3. **Vérifiez la programmation PFCP :**
   - Recherchez les journaux pour "PFCP Session Establishment Request"
   - Vérifiez que le QER est inclus dans le message
   - Vérifiez les journaux PGW-U pour les erreurs de traitement PFCP
4. **Vérifiez la configuration de la politique PCRF :**
   - Vérifiez le profil d'abonné dans le PCRF
   - Confirmez que des règles de politique spécifiques à l'APN existent
   - Vérifiez les journaux PCRF pour les erreurs d'évaluation de la politique

Surveiller le taux de création de Bearer

Requêtes Prometheus :

```
# Taux de création de bearer par défaut (sessions/seconde)
rate(s5s8_inbound_messages_total{message_type="create_session_request
[5m])

# Taux de création de bearer dédié
rate(s5s8_outbound_messages_total{message_type="create_bearer_request
[5m])

# Taux de mise à jour de la politique du PCRF
rate(gx_inbound_messages_total{message_type="gx_RAR"}[5m])
```

Planification de la capacité

Métriques clés à surveiller :

```
# Utilisation des adresses IP UE (pourcentage)
(address_registry_count / <configured_pool_size>) * 100

# Compte des bearers actifs
session_registry_count

# Latence de requête PCRF (P95)
histogram_quantile(0.95, gx_outbound_transaction_duration_bucket)
```

Limites de capacité :

- Taille du pool d'adresses : configurée dans `config/runtime.exs` sous `ue.subnet_map`
- Espace TEID : 32 bits (4 milliards d'identifiants uniques, gérés automatiquement)
- Sessions simultanées : généralement limitées par la taille du pool d'adresses

Directives de planification :

- Surveillez l'utilisation des adresses IP - augmentez le pool avant de dépasser 80%
- Surveillez la latence PCRF - une latence élevée impacte le temps de configuration de session
- Surveillez le taux de création de bearers dédiés - indique la complexité de la politique

Documentation Connexe

- Gestion des Sessions - Cycle de vie de la session PDN
- Interface Diameter Gx - Détails du protocole de politique PCRF
- Interface PFCP - Programmation du plan utilisateur
- Guide de Configuration - Configuration du système
- Guide de Surveillance - Métriques et observabilité

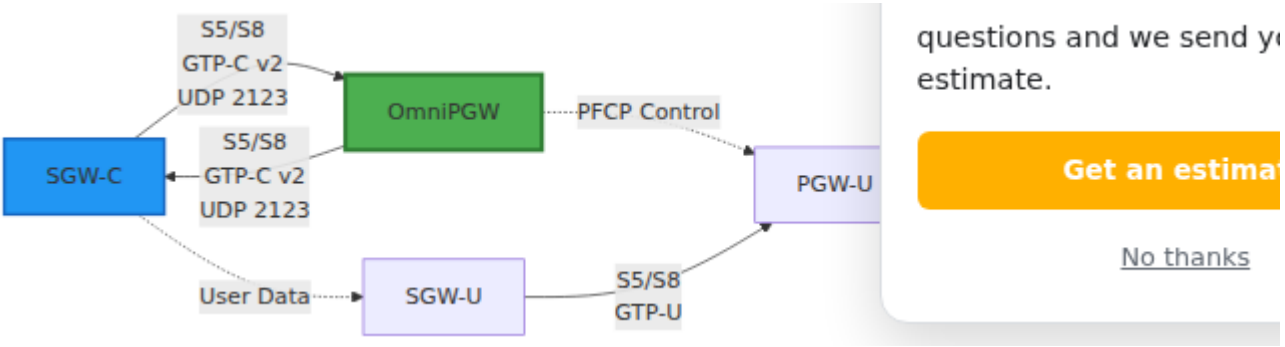
Documentation de l'Interface S5/S8

Communication GTP-C avec SGW-C

OmniPGW par Omnitouch Network Services

Vue d'ensemble

L'**interface S5/S8** connecte OmniPGW au SGW-C (Serving Gateway Control plane) en utilisant le protocole **GTP-C v2** (GPRS Tunnelling Protocol - Control plane). Cette interface gère le signalement de gestion de session entre les passerelles.



Détails du Protocole

GTP-C Version 2

- **Protocole :** GTP-C v2 (3GPP TS 29.274)
- **Transport :** UDP
- **Port :** 2123 (standard)
- **Type d'interface :** Plan de contrôle

TEID (Identifiant de Point de Terminaison de Tunnel)

Chaque session a un **TEID** unique pour le routage des messages :

- **TEID Local** - Alloué par OmniPGW pour les messages entrants
- **TEID Distant** - Alloué par SGW-C pour les messages sortants

```
Flux de Message :
  SGW-C → OmniPGW : TEID de destination = TEID local d'OmniPGW
  OmniPGW → SGW-C : TEID de destination = TEID distant de SGW-C
```

Configuration

Configuration de Base

```
# config/runtime.exs
config :pgw_c,
  s5s8: %{
    # Adresse IPv4 locale pour l'interface S5/S8
    local_ipv4_address: "10.0.0.20",

    # Optionnel : Adresse IPv6 locale
    local_ipv6_address: nil,

    # Optionnel : Remplacer le port par défaut
    local_port: 2123,

    # Délai d'attente de la requête GTP-C en millisecondes (par
défaut : 500ms)
    # Délai d'attente par tentative lors de l'attente des réponses
GTP-C (Créer Bearer, Supprimer Bearer, etc.)
    request_timeout_ms: 500,

    # Nombre de tentatives de réessai pour les requêtes GTP-C (par
défaut : 3)
    # Temps d'attente maximum total = request_timeout_ms *
request_attempts
    # Exemple : 500ms * 3 tentatives = 1500ms (1,5 secondes) au
total
    request_attempts: 3
  }
```

Configuration des Délais

L'interface S5/S8 utilise des délais configurables pour les transactions de requête/réponse GTP-C.

Paramètres :

- **`request_timeout_ms`** - Délai en millisecondes par tentative de réessai (par défaut : 500ms)

- `request_attempts` - Nombre de tentatives de réessai avant d'abandonner (par défaut : 3)

Temps d'attente total : `request_timeout_ms × request_attempts`

Comportement par défaut : 500ms × 3 tentatives = **1,5 secondes au maximum**

Directives de Réglage :

Latence Réseau	`request_timeout_ms` recommandé	Temps d'attente total
Faible latence (<50ms)	200-300ms	600-900ms (3 tentatives)
Normal (50-150ms)	500ms (par défaut)	1.5s (3 tentatives)
Haute latence (>150ms)	1000-2000ms	3-6s (3 tentatives)
Instable/satellite	2000-3000ms	6-9s (3 tentatives)

Exemple - Réseau à Haute Latence :

```
s5s8: %{
  local_ipv4_address: "10.0.0.20",
  request_timeout_ms: 1500,  # 1,5 secondes par tentative
  request_attempts: 3         # Total : 4,5 secondes max
}
```

Lorsque le délai d'attente se produit :

- OmniPGW enregistre une erreur : `"Délai d'attente de la requête de création de bearer"`
- Retourne une erreur au PCRF (Code de résultat Diameter : 5012 UNABLE_TO_COMPLY)

- Le bearer reste en stockage précoce pour nettoyage via Charging-Rule-Remove

Exigences Réseau

Règles de Pare-feu :

```
# Autoriser GTP-C depuis le réseau SGW-C
iptables -A INPUT -p udp --dport 2123 -s <sgw_network>/24 -j
ACCEPT

# Autoriser GTP-C sortant vers SGW-C
iptables -A OUTPUT -p udp --dport 2123 -d <sgw_network>/24 -j
ACCEPT
```

Routage :

```
# Assurez-vous d'une route vers le réseau SGW-C
ip route add <sgw_network>/24 via <gateway_ip> dev eth0
```

Types de Messages

L'interface S5/S8 gère le signalement GTP-C pour la gestion des sessions PDN. Pour une description détaillée du cycle de vie des sessions et de la gestion des états, voir Guide de Gestion des Sessions.

Gestion des Sessions

Requête de Création de Session

Direction : SGW-C → OmniPGW

But : Établir une nouvelle connexion PDN

IEs Clés (Éléments d'Information) :

Nom de l'IE	Type	Description
IMSI	Identité	Identité Internationale de l'Abonné Mobile
MSISDN	Identité	Numéro de téléphone mobile
APN	Chaîne	Nom du Point d'Accès (par exemple, "internet")
Type de RAT	Enum	Technologie d'Accès Radio (EUTRAN)
Contexte de Bearer	Groupé	Informations sur le bearer par défaut
Fuseau Horaire UE	Horodatage	Fuseau horaire de l'UE
ULI	Groupé	Informations de Localisation de l'Utilisateur (TAI, ECGI)
Réseau de Service	PLMN	MCC/MNC du réseau de service

Exemple :

```
Requête de Création de Session
├── IMSI : 310260123456789
├── MSISDN : 14155551234
├── APN : internet
├── Type de RAT : EUTRAN (6)
├── Contexte de Bearer
│   ├── EBI : 5
│   ├── QoS de Bearer (QCI 9, ARP, débits)
│   └── S5/S8 F-TEID (point de terminaison de tunnel SGW-U)
└── ULI
    ├── TAI : MCC 310, MNC 260, TAC 12345
    └── ECGI : MCC 310, MNC 260, ECI 67890
```

Réponse de Création de Session

Direction : OmniPGW → SGW-C

But : Accuser réception de la création de session

IEs Clés :

Nom de l'IE	Type	Description
Cause	Résultat	Code de succès ou d'erreur
Contexte de Bearer	Groupé	Informations sur le bearer
Allocation d'Adresse PDN	IP	Adresse IP UE allouée (voir Allocation d'IP UE)
Restriction APN	Enum	Restrictions d'utilisation de l'APN
PCO	Options	Options de Configuration de Protocole (voir Configuration PCO)

Réponse de Succès :

```
Réponse de Création de Session
├── Cause : Requête acceptée (16)
├── Allocation d'Adresse PDN
│   └── IPv4 : 100.64.1.42
├── Contexte de Bearer
│   ├── EBI : 5
│   ├── Cause : Requête acceptée
│   └── S5/S8 F-TEID (point de terminaison de tunnel PGW-U depuis
PFCP)
├── Restriction APN : Public-1 (1)
└── PCO
    ├── Serveur DNS : 8.8.8.8
    ├── Serveur DNS : 8.8.4.4
    └── MTU de Lien : 1400
```

> La disposition de l'Allocation d'Adresse PDN suit le type de PDN négocié (TS 29.274 §8.14) : une adresse IPv4 de 4 octets ; un octet de longueur de préfixe plus un préfixe IPv6 de 16 octets ; ou pour IPv4v6 un octet de longueur de préfixe plus un préfixe IPv6 de 16 octets plus une adresse IPv4 de 4 octets. Si un PDN IPv4v6 est demandé pour un APN d'urgence (SOS) qui n'a pas de pool IPv6, le PGW-C dégrade à un PAA IPv4 uniquement (les APN réguliers sont rejetés) — voir demande IPv4v6 lorsque l'APN n'a pas de pool IPv6.

Requête de Suppression de Session

Direction : SGW-C → OmniPGW

But : Terminer la connexion PDN

IEs Clés :

Nom de l'IE	Description
EBI	ID de Bearer EPS à supprimer
EBI Lié	Bearer associé (optionnel)

Réponse de Suppression de Session

Direction : OmniPGW → SGW-C

But : Accuser réception de la suppression de session

IEs Clés :

Nom de l'IE	Description
Cause	Code de succès ou d'erreur

Gestion des Bearers

Requête de Création de Bearer

Direction : OmniPGW → SGW-C

But : Créer un bearer dédié (initié par la politique PCRF)

Déclenché par :

- Le PCRF envoie une nouvelle règle PCC nécessitant un bearer dédié
- OmniPGW demande à SGW-C d'établir le bearer

Requête de Suppression de Bearer

Direction : OmniPGW → SGW-C ou SGW-C → OmniPGW

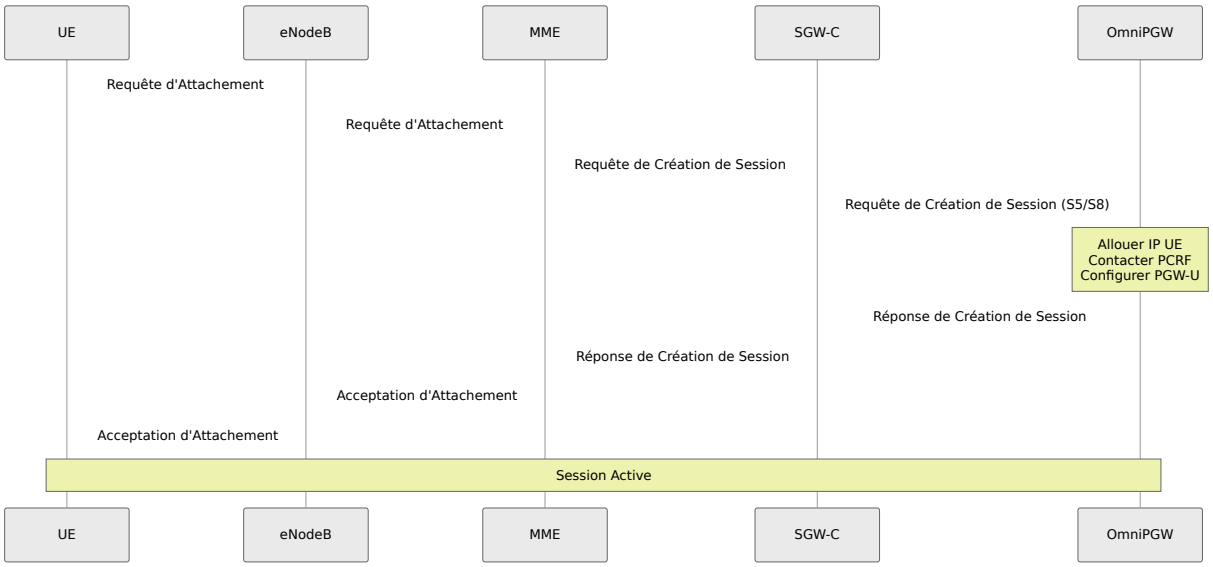
But : Supprimer le bearer dédié

Scénarios :

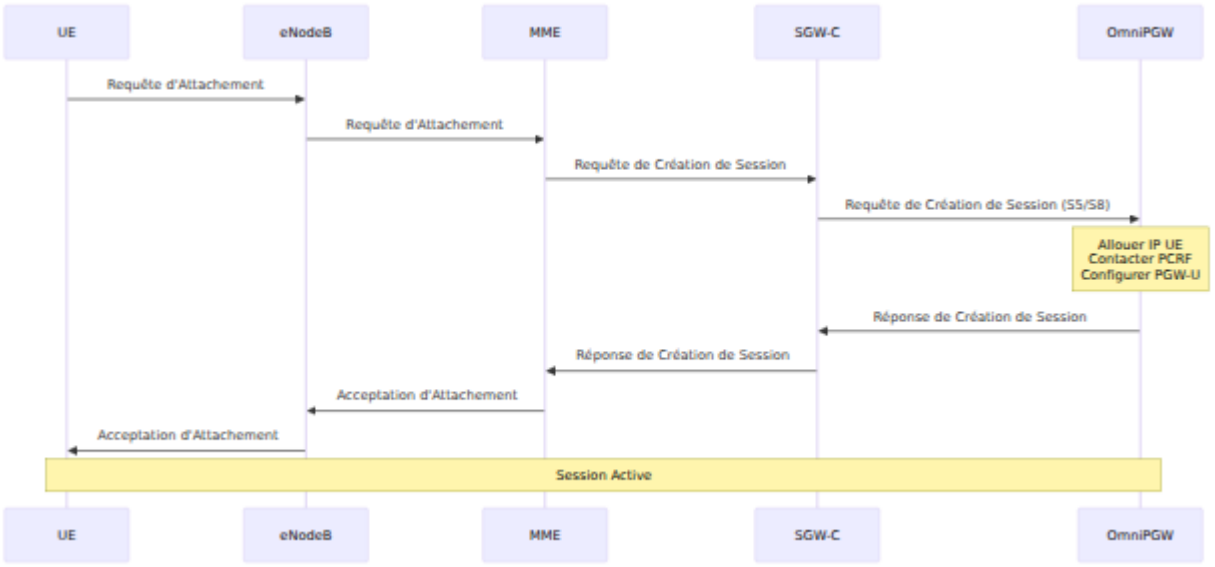
- **Initié par PGW :** Changement de politique PCRF supprime le bearer dédié
- **Initié par SGW :** Libération de ressources radio

Flux de Messages

Établissement de Session



Terminaison de Session



Codes de Cause

Succès

Code	Nom	Description
16	Requête acceptée	Opération réussie

Erreurs (Échecs Permanents)

Code	Nom	Quand Utilisé
65	Utilisateur Inconnu	Rejet PCRF (IMSI non trouvé)
66	Pas de ressources disponibles	Pool IP épuisé
93	Service non pris en charge	APN invalide
94	Erreur sémantique dans TFT	Modèle de flux de trafic invalide

Erreurs (Échecs Transitoires)

Code	Nom	Quand Utilisé
72	Pair distant ne répond pas	Délai d'attente PCRF/PGW-U
73	Collision avec une requête initiée par le réseau	Opérations simultanées

---

Surveillance

Métriques S5/S8

```
# Compteurs de messages
s5s8_inbound_messages_total{message_type="create_session_request"}
s5s8_inbound_messages_total{message_type="delete_session_request"}

# Compteurs d'erreurs
s5s8_inbound_errors_total

# Latence de traitement des messages
s5s8_inbound_handling_duration_bucket

# TEIDs actifs
teid_registry_count
```

Requêtes Utiles

Taux de Création de Session :

```
rate(s5s8_inbound_messages_total{message_type="create_session_request
[5m])
```

Taux d'Erreur :

```
rate(s5s8_inbound_errors_total[5m])
```

Latence (p95) :

```
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea
[5m])
)
```

Dépannage

Problème : Pas de Réponse d'OmniPGW

Symptômes :

- SGW-C envoie une Requête de Création de Session
- Pas de réponse reçue
- Délai d'attente à SGW-C

Causes :

1. Problème de connectivité réseau
2. OmniPGW ne répond pas à l'IP configurée
3. Pare-feu bloquant UDP 2123
4. Mauvais TEID dans la requête

Débogage :

```
# Vérifiez qu'OmniPGW écoute
netstat -ulnp | grep 2123

# Vérifiez les paquets entrants
tcpdump -i any -n port 2123

# Vérifiez la configuration
grep "local_ipv4_address" config/runtime.exs

# Vérifiez le pare-feu
iptables -L -n | grep 2123
```

Problème : Échec de Création de Session

Symptômes :

- Réponse de Création de Session avec cause d'erreur
- Session non établie

Causes Courantes :

```
Cause 65 (Utilisateur Inconnu) :
  → PCRF a rejeté l'abonné
  → Vérifiez l'IMSI dans HSS/SPR

Cause 66 (Pas de ressources) :
  → Pool IP épuisé
  → Vérifiez : curl http://pgw:9090/metrics | grep
address_registry_count
  → Étendre le pool IP

Cause 72 (Pair distant ne répond pas) :
  → Délai d'attente PCRF ou PGW-U hors service
  → Vérifiez la connectivité Gx
  → Vérifiez l'association PFCP
```

Problème : Collision de TEID

Symptômes :

- Message routé vers la mauvaise session
- Comportement inattendu

Cause :

- TEID réutilisé avant nettoyage
- Bug dans l'allocation de TEID

Résolution :

- Assurez-vous d'une allocation unique de TEID
- Vérifiez le registre de TEID pour les fuites

Meilleures Pratiques

Conception Réseau

1. **Interface Réseau Dédiée**
    - Utilisez un VLAN séparé pour S5/S8
    - Isolez du trafic de gestion
2. **Optimisation MTU**
    - Assurez-vous que le MTU prend en charge les en-têtes GTP
    - MTU minimum : 1500 octets (1464 charge utile + 36 GTP)
3. **Redondance**
    - Plusieurs instances d'OmniPGW
    - Équilibrage de charge basé sur DNS depuis SGW-C

Performance

1. **Tailles de Tampon UDP**
    - Augmentez les tampons de socket pour une charge élevée

    - Typique : 4-8 Mo par socket

2. **Limites de Connexion**
    - Planifiez le nombre de sessions attendues
    - Surveillez le compte du registre de TEID

Sécurité

1. **Filtrage IP**
    - Autorisez uniquement GTP-C depuis des IP SGW-C connues
    - Utilisez iptables ou ACL réseau

2. **Validation des Messages**
    - OmniPGW valide tous les messages entrants
    - Rejette les paquets GTP-C malformés

Documentation Connexe

Fonctions Principales

- **Guide de Configuration** - Configuration de l'interface S5/S8, configuration de l'IP locale
- **Gestion des Sessions** - Cycle de vie de la session PDN, établissement de bearer
- **Allocation d'IP UE** - Livraison d'adresse IP via Réponse de Création de Session
- **Configuration PCO** - Paramètres PCO dans les messages GTP-C

Interfaces Connexes

- **Interface Gn/Gp** - GTP-C v1 pour la fonctionnalité GGSN 2G/3G

- **Interface PFCP** - Coordination du plan utilisateur avec le plan de contrôle S5/S8
- **Interface Diameter Gx** - Intégration de la politique avec l'établissement de bearer
- **Interface Diameter Gy** - Intégration de la facturation avec la gestion de bearer

Opérations

- **Guide de Surveillance** - Métriques GTP-C S5/S8, suivi des messages
- **Format CDR de Données** - Génération de CDR à partir des sessions GTP-C

Retour au Guide des Opérations

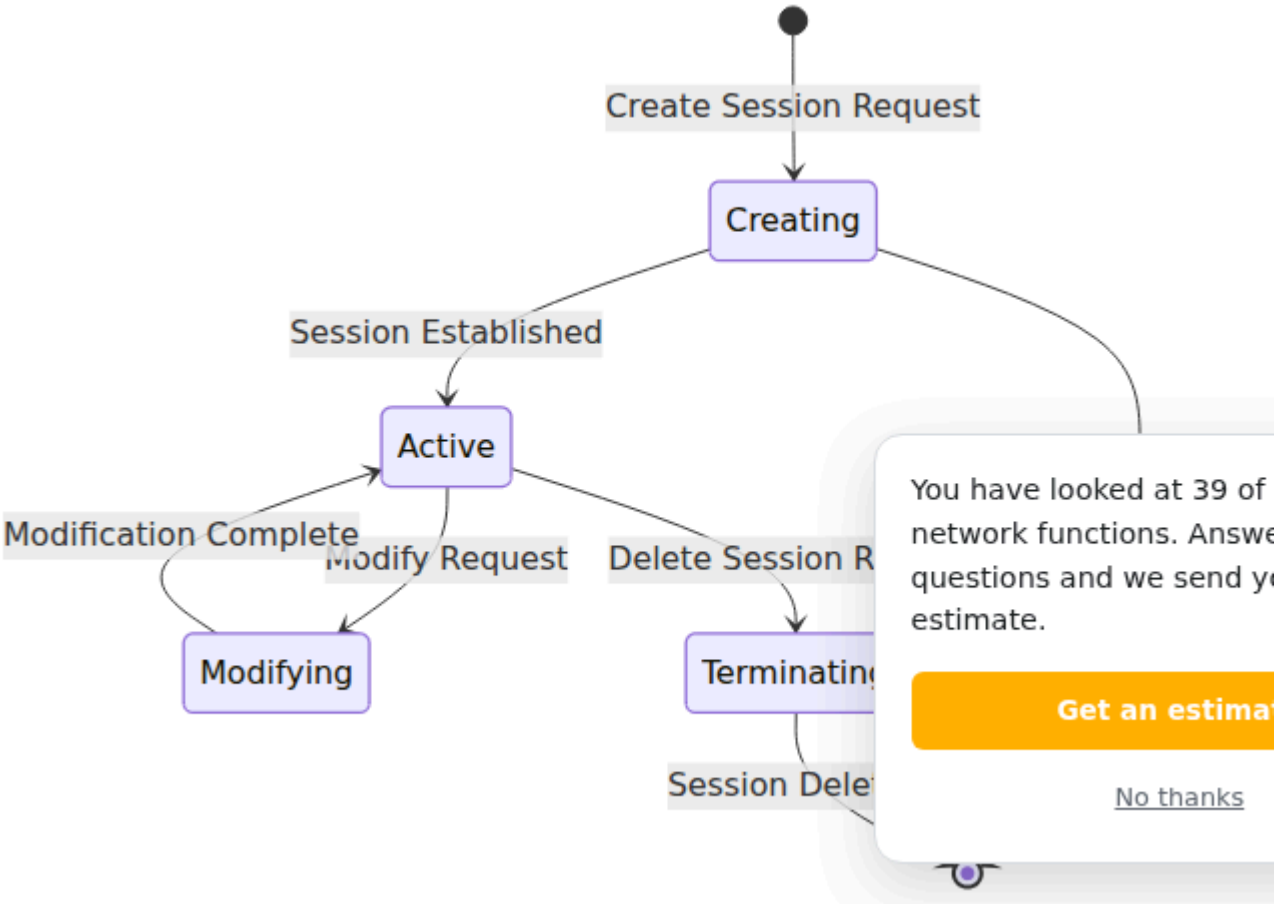
Guide de Gestion des Sessions

Cycle de Vie et Opérations de la Connexion PDN

OmniPGW par Omnitouch Network Services

Vue d'ensemble

Une **Session PDN (Packet Data Network)** représente la connexion de données d'un UE à travers OmniPGW. Chaque session coordonne plusieurs interfaces et ressources pour permettre la connectivité des données.



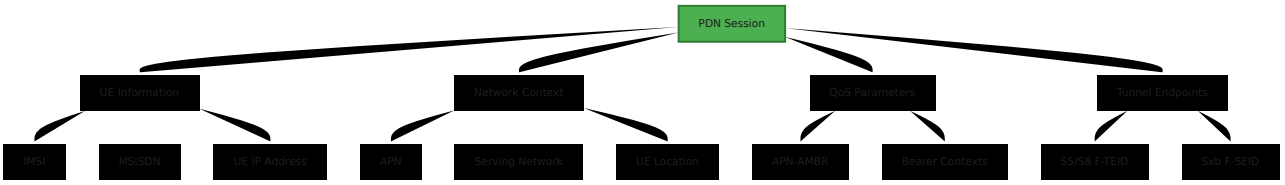
Composants de la Session

Identifiants de Session

Chaque session a plusieurs identifiants pour différentes interfaces :

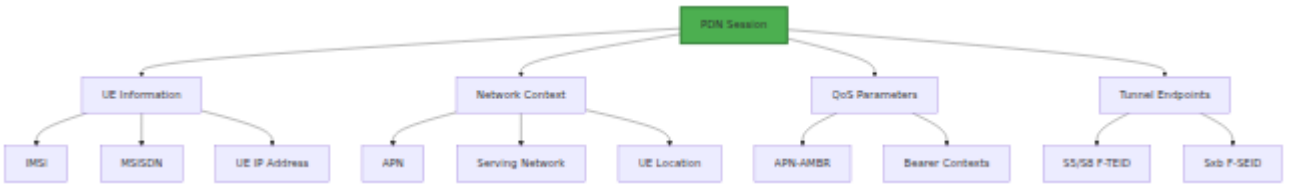
Identifiant	Interface	But
TEID	S5/S8 (GTP-C)	ID de point de terminaison de tunnel pour la communication SGW-C
SEID	Sxb (PFCP)	ID de point de terminaison de session pour la communication PGW-U
Session-ID	Gx (Diameter)	Session Diameter pour la communication PCRF
Charging-ID	Comptabilité	ID unique pour la facturation

Données de Session



Création de Session

Flux d'Appel



Étapes

1. Recevoir la Demande de Création de Session (S5/S8)

La création de session est initiée via la signalisation GTP-C sur l'interface S5/S8. Voir Interface S5/S8 pour les détails complets du protocole GTP-C et des formats de message.

Entrée :

- IMSI, MSISDN, IMEI
- APN (par exemple, "internet")
- Type de RAT (EUTRAN)
- Localisation de l'UE (TAI, ECGI)
- Contexte de porteur (QoS, F-TEID)

2. Allocation des Ressources

```
- Allouer IP UE à partir du pool APN
- Générer ID de facturation
- Générer Gx Session-ID
- Allouer S5/S8 TEID
- Sélectionner le pair PGW-U
```

3. Demande de Politique (Gx)

Demander une politique au PCRF :

- Envoyer CCR-Initial
- Recevoir CCA-Initial avec QoS et règles PCC

4. Configuration du Plan Utilisateur (PFCP)

Programmer le PGW-U avec les règles de transfert :

- Envoyer la Demande d'Établissement de Session
- Inclure PDRs, FARs, QERs, BAR
- Recevoir F-TEID pour le tunnel S5/S8

5. Réponse au SGW-C

Envoyer la Réponse de Création de Session :

- Adresse IP de l'UE
- S5/S8 F-TEID (de PGW-U)
- PCO (DNS, P-CSCF, MTU)
- Contexte de porteur

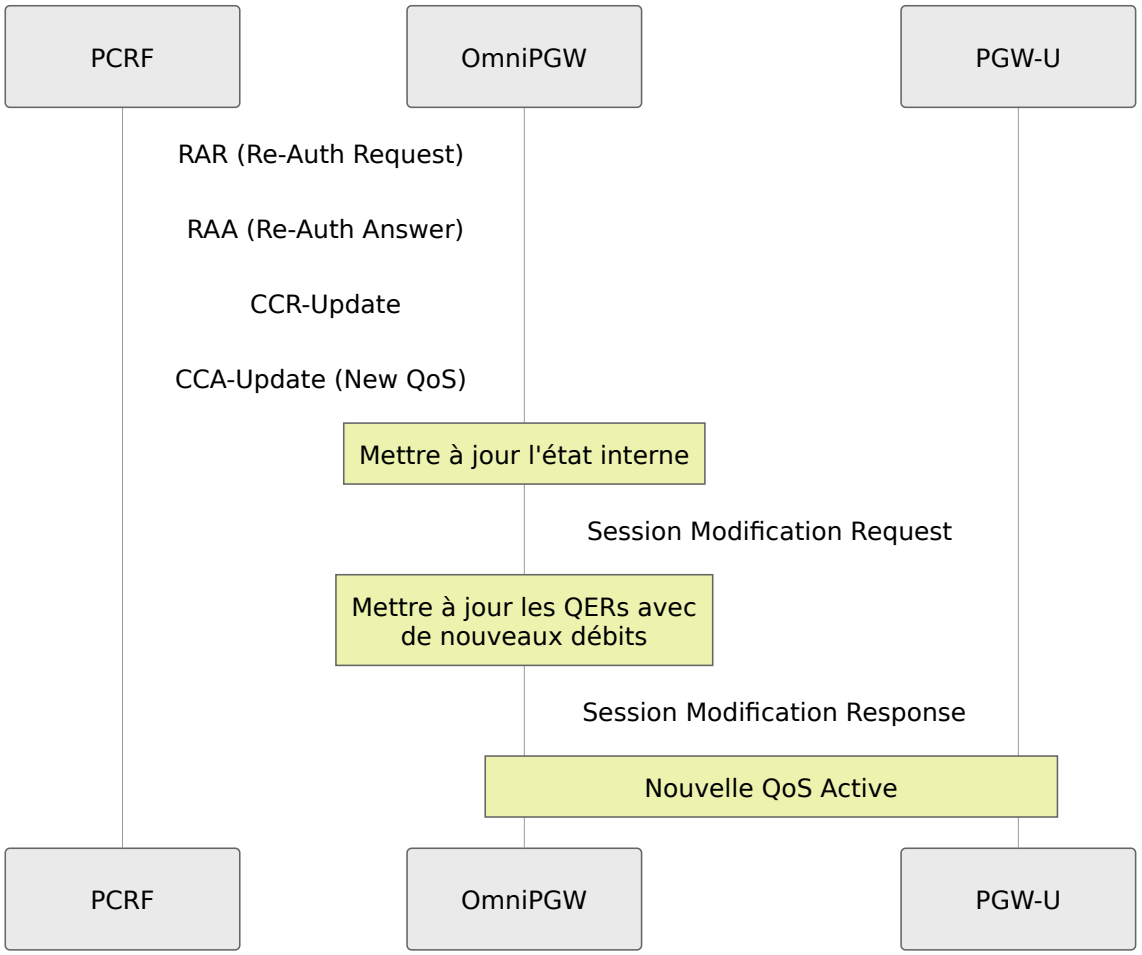
Modification de Session

Déclencheurs

Les sessions peuvent être modifiées en raison de :

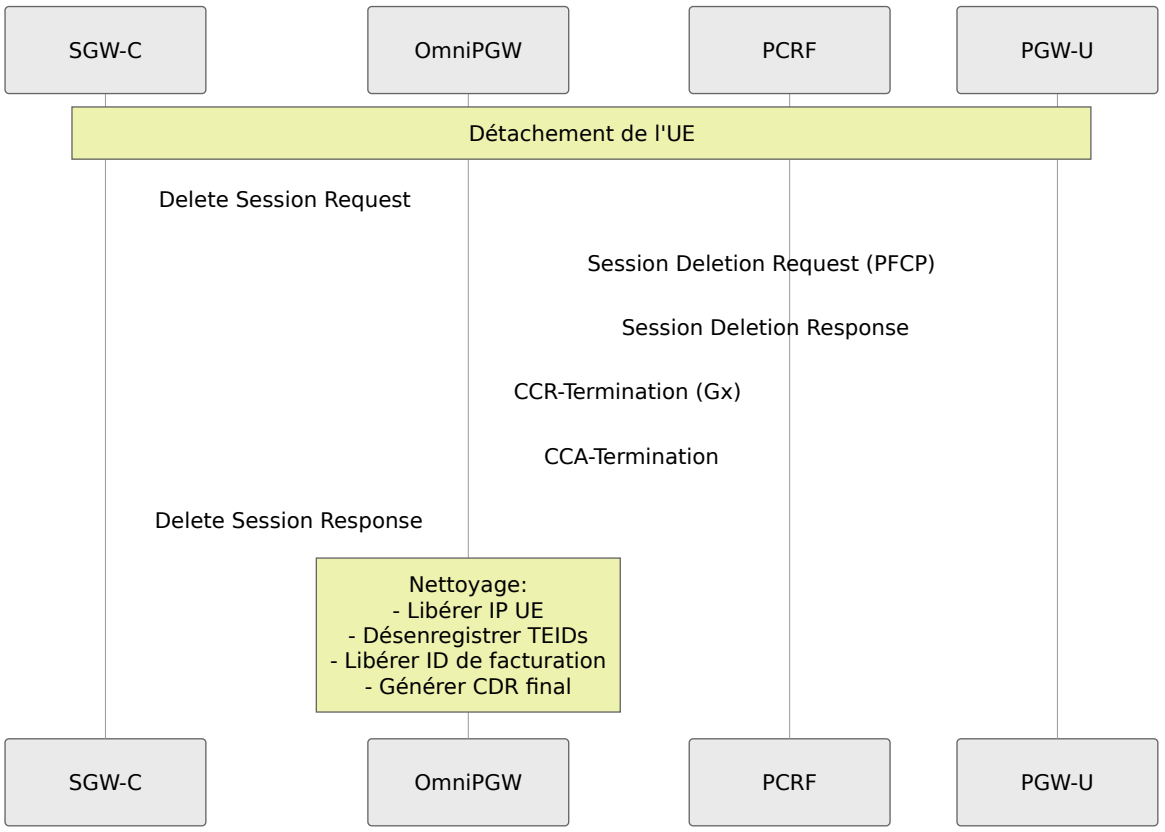
- **Changements de QoS** - mises à jour des débits par le PCRF
- **Opérations de porteur** - Ajouter/retirer des porteurs dédiés
- **Transfert** - Changement de SGW
- **Mises à jour de politique** - Nouvelles règles PCC du PCRF

Flux de Modification de QoS



Suppression de Session

Flux d'Appel



Processus de Nettoyage

Ressources Libérées :

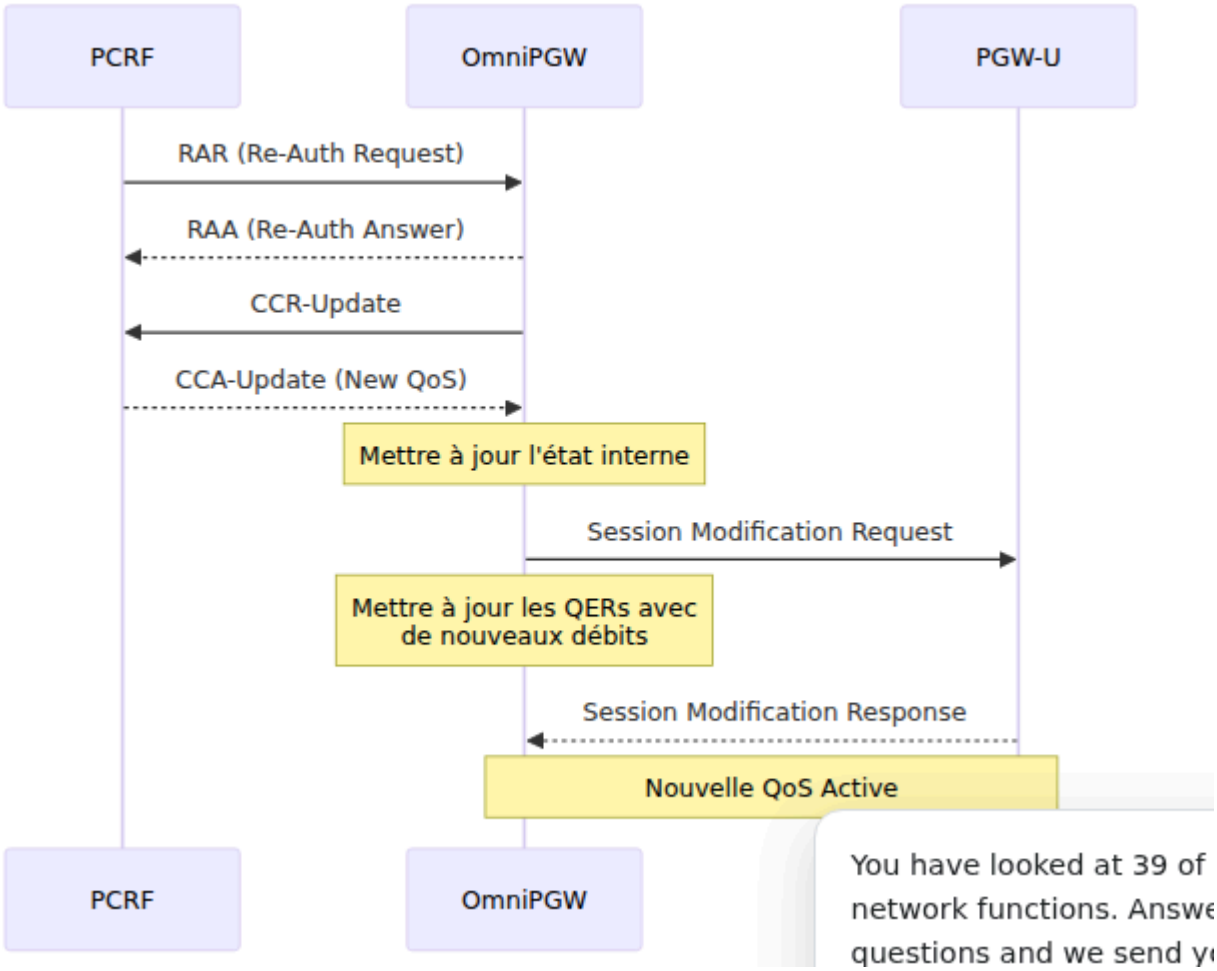
1. Adresse IP UE → retour au pool
2. TEID → retiré du registre
3. SEID → retiré du registre
4. Session-ID → retiré du registre
5. Charging-ID → libéré
6. Processus de session terminé

Enregistrements de Facturation Générés :

- CDR final (Charging Data Record) écrit pour la facturation hors ligne - Voir Format de CDR de Données

État de Session

Machine d'État



Suivi de Session

Recherches dans le Registre :

```
Par TEID (S5/S8):
  TEID 0x12345678 → Session PID

Par SEID (Sxb):
  SEID 0xABCDEF → Session PID

Par Session-ID (Gx):
  "pgw.example.com;123;456" → Session PID

Par IP UE:
  100.64.1.42 → Session PID

Par IMSI + EBI:
  "310260123456789" + EBI 5 → Session PID
```

Surveillance des Sessions

Compte des Sessions Actives

```
# Total des sessions actives
teid_registry_count

# Sessions PFCP
seid_registry_count

# Sessions Gx
session_id_registry_count
```

Métriques de Session

```
# Taux de création de session
rate(s5s8_inbound_messages_total{message_type="create_session_request

# Taux de suppression de session
rate(s5s8_inbound_messages_total{message_type="delete_session_request

# Latence de création de session (p95)
histogram_quantile(0.95,

rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea
[5m])
)
```

Problèmes Courants

Échec de Création de Session

Causes :

1. **Pool IP Épuisé** - Pas d'IP disponibles
2. **PCRF Injoignable** - Délai d'attente Gx
3. **PGW-U Hors Service** - Pas de pair PFCP disponible
4. **Rejet PCRF** - Utilisateur inconnu, non autorisé

Débogage :

```
# Vérifier le pool IP
curl http://pgw:9090/metrics | grep address_registry_count

# Vérifier la connectivité PCRF
# Vérifier les erreurs Gx dans les journaux

# Vérifier l'association PGW-U
# Vérifier l'état du pair PFCP
```

Session Bloquée/Obsolète

Symptômes :

- Session non supprimée correctement
- Ressources non libérées
- Registres montrent un nombre plus élevé que prévu

Causes :

1. Demande de suppression de session non reçue
2. Plantage du processus de session sans nettoyage
3. Fuite de registre

Résolution :

```
# Redémarrer OmniPGW (libère toutes les sessions)
# Mettre en œuvre un mécanisme de délai d'attente pour les
sessions obsolètes
```

L'UE Ne Peut Pas Établir de Session

Symptômes :

- Échec de l'attachement de l'UE
- Réponse de création de session avec cause d'erreur

Causes et Réponses Courantes :

Valeur de Cause	Signification	Action
Utilisateur Inconnu	Rejet PCRF (IMSI non dans la base de données)	Provisionner l'abonné
Pas de Ressources Disponibles	Pool IP épuisé	Étendre le pool IP
Pair Distant Ne Répond Pas	Délai d'attente PCRF/PGW-U	Vérifier la connectivité
Service Non Supporté	APN invalide	Configurer le pool APN

Meilleures Pratiques

Limites de Session

Configurer une capacité appropriée :

```
Utilisateurs concurrents attendus : 10,000
Surcharge de session par utilisateur : ~10 Ko RAM
RAM totale pour les sessions : ~100 Mo

Paramètres de la VM Erlang :
  - Max processus : 262,144 (par défaut)
  - Taille du tas de processus : Ajuster en fonction de la charge
```

Nettoyage de Session

Assurer un nettoyage approprié :

1. Toujours répondre aux demandes de suppression de session
2. Mettre en œuvre un délai d'attente de session pour les sessions obsolètes

3. Surveiller les comptes de registre pour les fuites

Haute Disponibilité

Redondance de Session :

- Utiliser un design sans état (sessions liées à l'instance)
- Mettre en œuvre une base de données de session pour la HA (futur)
- DNS/équilibreur de charge pour la bascule

Éléments de Données de Session

Quelles Informations Une Session Stocke-t-elle ?

Chaque session PDN active maintient les informations suivantes :

Identification de l'UE :

- IMSI : "310260123456789" (identité de l'abonné)
- MSISDN : "14155551234" (numéro de téléphone)
- MEI/IMEI : Identifiant de l'appareil

Détails de la Connexion PDN :

- APN : "internet" (nom du réseau)
- Adresse IP de l'UE : 100.64.1.42 (IP allouée)
- Type de PDN : IPv4, IPv6, ou IPv4v6

Identifiants de Session :

- ID de Facturation : Identifiant de facturation unique
- EBI de Porteur par Défaut : Identifiant de Porteur EPS (typiquement 5)

Paramètres de QoS :

- APN-AMBR : Débit Maximum Agrégé
  - Montant : 100 Mbps
  - Descendant : 50 Mbps

Règles de Transfert :

- PDRs (Règles de Détection de Paquet) : Correspondre aux paquets
- FARs (Règles d'Action de Transfert) : Actions de transfert/abandon
- QERs (Règles d'Application de QoS) : Limitation de débit
- BAR (Règle d'Action de Mise en Tampon) : Mise en tampon descendant

Contexte d'Interface :

- État S5/S8 : TEIDs locaux/à distance, adresse SGW-C
- État Sxb : SEIDs locaux/à distance, adresse PGW-U
- État Gx : Session-ID Diameter, compteur de requêtes

Surveillance des Sessions en Direct

OmniPGW expose une **API REST OAM** en temps réel pour inspecter et gérer les sessions actives sans avoir besoin de consulter les métriques ou les journaux. L'API est servie sur HTTPS/TLS sur le port `8443` avec toutes les routes sous le préfixe `/api` :

```
GET    /api/sessions                 # toutes les sessions actives
GET    /api/sessions?search=<imsi>   # filtrer par IMSI / MSISDN /
IP
GET    /api/sessions/<imsi>          # détails complets d'une
session
PUT    /api/sessions/<imsi>          # mettre à jour l'état de la
session en direct (action de l'opérateur)
DELETE /api/sessions/<imsi>          # détruire une session
(action de l'opérateur)
```

Des documents API interactifs (Swagger UI) sont disponibles à `https://<omnipgw-ip>:8443/api/docs`.

```
# Lister toutes les sessions actives
curl -k https://<omnipgw-ip>:8443/api/sessions

# Filtrer par IMSI, MSISDN ou adresse IP
curl -k "https://<omnipgw-ip>:8443/api/sessions?
search=310170123456789"
```

Recherche UE & Exploration Approfondie

Points de terminaison :

```
GET /api/sessions?search=<imsi|msisdn|ip>   # trouver les sessions
d'un abonné
GET /api/sessions/<imsi>                    # détails complets
pour une session
```

Le paramètre `search` accepte n'importe lequel de :

- **IMSI** (par exemple, "310170123456789")
- **MSISDN** (numéro de téléphone)
- **Adresse IP** (par exemple, "100.64.1.42")

```
# Exploration approfondie de la session d'un abonné unique
curl -k https://<omnipgw-ip>:8443/api/sessions/310170123456789
```

Les détails de la session retournent des informations complètes :

a) Sessions Actives

- Toutes les sessions actives pour cet abonné
- IMSI, MSISDN, Adresse IP de l'UE
- APN, Type de RAT
- PGW TEID, SGW TEID

b) Localisation Actuelle Données de localisation en temps réel de la session :

- **TAC** (Code de Zone de Suivi) - Zone de suivi où se trouve l'UE

- **Cell ID (ECI)** - Identifiant de Cellule E-UTRAN
- **ECGI** - Identifiant Global de Cellule E-UTRAN (PLMN + ECI)
- **MCC/MNC** - Code de Pays Mobile / Code de Réseau Mobile

c) Informations de Porteur Liste détaillée des porteurs avec paramètres QoS :

Porteur par Défaut :

- EBI (Identifiant de Porteur EPS)
- QCI (Identifiant de Classe de QoS)
- Nom de la Règle de Facturation
- APN-AMBR (montant/descendant)

Porteurs Dédiés (s'ils sont actifs) :

- EBI, QCI, Nom de la Règle de Facturation
- MBR UL/DL (Débit Maximum)
- GBR UL/DL (Débit Garanti)

d) Informations de Facturation (Interface Gy)

- ID de Session Gy
- Quota Accordé, Quota Utilisé
- Caractéristiques de Facturation

e) Informations de Politique (Interface Gx)

- ID de Session Gx
- Hôte d'Origine/Destination PCRF
- Numéro de Requête CC
- Règles de Facturation Installées (règles PCC des porteurs)

f) Événements Récents

- Historique des événements pour cet abonné (voir Historique de Session & Journal d'Audit)
- Événements de création/mise à jour/suppression de session

Cas d'utilisation :

- Dépanner des problèmes spécifiques d'abonné
- Vérifier l'établissement de session
- Vérifier l'adresse IP assignée
- Inspecter les paramètres de session

Toutes les Sessions Actives

Points de terminaison :

```
GET    /api/sessions                # lister toutes les sessions
PDN actives
GET    /api/sessions?search=<term>  # filtrer la liste
PUT    /api/sessions/<imsi>         # mettre à jour l'état d'une
session en direct
DELETE /api/sessions/<imsi>         # détruire une session
```

But : Énumérer toutes les sessions PDN actives. Interroger `GET /api/sessions` à votre intervalle préféré pour une vue en direct du tableau des sessions.

```
# Instantané de toutes les sessions actives
curl -k https://<omnipgw-ip>:8443/api/sessions

# Filtrer par préfixe IMSI, IP UE, MSISDN ou APN
curl -k "https://<omnipgw-ip>:8443/api/sessions?search=100.64"

# Actions de l'opérateur sur une session spécifique
curl -k -X PUT    https://<omnipgw-
ip>:8443/api/sessions/310170123456789   # mettre à jour l'état
curl -k -X DELETE https://<omnipgw-
ip>:8443/api/sessions/310170123456789   # détruire
```

Informations Rapides sur la Session (retournées par session) :

- **IMSI** - Identité de l'abonné
- **UE IP** - Adresse IP allouée
- **SGW TEID** - ID de tunnel S5/S8 du SGW

- **PGW TEID** - ID de tunnel S5/S8 de l'OmniPGW
- **APN** - Nom du Point d'Accès

Filtrage (via `?search=`) :

- IMSI (par exemple, "310260")
- Adresse IP de l'UE (par exemple, "100.64")
- MSISDN / numéro de téléphone
- Nom de l'APN

Détails Complets (`GET /api/sessions/<imsi>`):

- Informations complètes sur l'abonné (IMSI, MSISDN, IMEI)
- Contexte réseau (type de RAT, réseau de service MCC/MNC)
- Paramètres QoS (AMBR montant/descendant)
- Identifiants de tunnel (les deux TEIDs en format hexadécimal)
- ID de processus pour le débogage
- État complet de la session (structure de données brute)

Points de terminaison de Santé Croisée

L'API OAM expose également la santé des pairs/topologies dont dépend la vue de session :

```
GET /api/upf         # statut des pairs PFCP / UPF et associations
GET /api/diameter    # connectivité des pairs Diameter (Gx/Gy)
```

```
# Vérifier la connectivité du plan utilisateur et de Diameter
curl -k https://<omnipgw-ip>:8443/api/upf
curl -k https://<omnipgw-ip>:8443/api/diameter
```

Cas d'utilisation :

- Vérifier les connexions de pairs (associations UPF, connectivité PCRF/OCS)
- Corréler les échecs d'établissement de session avec la santé des pairs
- Vérification rapide de la santé du réseau

Historique de Session & Journal d'Audit

Point de terminaison :

```
GET /api/session_history[?type=&search=&page=&page_size=]
```

But : Suivre les événements historiques de session et la piste d'audit.

Paramètres :

- `type` - filtrer par type d'événement (par exemple, session créée, session supprimée)
- `search` - filtrer par IMSI, MSISDN, adresse IP ou TEID
- `page`, `page_size` - paginer à travers les résultats

```
# Événements de suppression récents pour un abonné, première page
curl -k "https://<omnipgw-ip>:8443/api/session_history?
type=session_deleted&search=310170123456789&page=1&page_size=50"
```

Types d'Événements Suivis :

- Événements de création de session
- Événements de suppression de session
- Événements de modification
- Événements d'erreur

Cas d'utilisation :

- Piste d'audit pour la conformité (parcourir et archiver les résultats)
- Analyse historique des sessions
- Dépanner des problèmes passés
- Générer des rapports d'utilisation
- Suivre les modèles de session au fil du temps

Cas d'Utilisation Opérationnels

Vérification de Session :

```
1. L'utilisateur signale un problème de connectivité
2. GET /api/sessions?search=<imsi|msisdn> pour localiser l'abonné
3. GET /api/sessions/<imsi> pour vérifier que la session existe et
que l'UE a une adresse IP
4. Vérifier que les valeurs QoS correspondent au plan de l'abonné
5. Vérifier que les points de terminaison de tunnel sont établis
```

Surveillance de Capacité :

- Interroger `GET /api/sessions` (ou la métrique `teid_registry_count`) pour le compte des sessions actives
- Comparer avec la capacité sous licence
- Identifier les modèles d'utilisation par APN (via `?search=<apn>`)

Dépannage :

- Trouver une session spécifique par n'importe quel identifiant via `?search=`
- Inspecter l'état complet de la session (`GET /api/sessions/<imsi>`) sans SSH/IEx
- Vérifier que les TEIDs SGW et PGW correspondent entre les systèmes
- Vérifier les valeurs AMBR appliquées par le PCRF
- Détruire une session bloquée avec `DELETE /api/sessions/<imsi>`

Avantages par rapport aux Métriques :

- Voir les détails de session individuels (les métriques montrent des agrégats)
- Capacités de recherche et de filtrage via `?search=`
- Inspection de l'état en temps réel
- Scriptable/automatisable (JSON sur HTTPS) pour intégration avec des outils

---

Documentation Connexe

Fonctions de Session de Base

- **Interface PFCP** - Établissement de session du plan utilisateur, PDRs, FARs, QERs, URRs
- **Allocation IP UE** - Attribution d'adresse IP, gestion du pool APN
- **Configuration PCO** - Paramètres DNS, P-CSCF, MTU livrés à l'UE
- **Guide de Configuration** - Sélection UPF, flux d'établissement de session

Politique et Facturation

- **Interface Diameter Gx** - Contrôle de politique PCRF, règles PCC, gestion de QoS
- **Interface Diameter Gy** - Facturation en ligne OCS, suivi des quotas
- **Format de CDR de Données** - Génération d'enregistrements de facturation hors ligne

Interfaces Réseau

- **Interface S5/S8** - Protocole GTP-C, communication SGW-C
- **Gestion de QoS & de Porteurs** - Application de QoS des porteurs

Opérations

- **Guide de Surveillance** - Métriques de session, suivi des sessions actives, alertes
- **Surveillance P-CSCF** - Surveillance des sessions IMS

Retour au Guide des Opérations

Guide de Dépannage OmniPGW

Procédures de Dépannage et Problèmes Courants

par Omnitouch Network Services

Table des Matières

1. Aperçu
2. Outils de Dépannage
3. Problèmes d'Établissement de Session
4. Problèmes PFCP / Plan Utilisateur
5. Problèmes Diameter (Gx/Gy)
6. Problèmes d'Allocation IP
7. Référence Rapide

Aperçu

Ce guide fournit des procédures de dépannage étape par étape pour les problèmes opérationnels courants d'OmniPGW. Chaque problème comprend :

- **Symptôme** : Ce que vous allez observer
- **Causes Probables** : Causes racines courantes
- **Diagnostic** : Comment confirmer la cause
- **Résolution** : Solution étape par étape
- **Prévention** : Comment éviter la récurrence

Documentation Connexe

- **Guide de Surveillance** - Métriques Prometheus, alertes, surveillance des performances
- **Guide de Configuration** - Référence de configuration système

---

Outils de Dépannage

API OAM

URL de Base : `https://<omnipgw_ip>:8443` (HTTPS/TLS ; toutes les routes sous le préfixe `/api`)

Docs API Interactives (Swagger UI) :
`https://<omnipgw_ip>:8443/api/docs`

Points de Terminaison Clés :

- `GET /api/sessions[?search=<imsi|msisdn|ip>]`, `GET /api/sessions/<imsi>` - Visualiseur de sessions actives (recherche par IMSI, IP, MSISDN)
- `GET /api/diameter[?status=connected|disconnected]`, `GET /api/diameter/<origin-host>` - Statut des pairs Diameter (Gx PCRF, Gy OCS)
- `GET /api/charging[?search=<imsi>]`, `GET /api/charging/<imsi>` - Statut de la facturation en ligne (Gy)
- `GET /api/upf`, `GET /api/upf/<ip>` - Statut des pairs UPF / PFCP (connectivité PGW-U)
- `GET /api/ip_pools`, `GET /api/ip_pools/<name>` - Utilisation des pools IP
- `GET /api/session_history[?type=&search=&page=&page_size=]` - Historique des sessions
- `GET /api/status` - Vérification de l'état (`{"result":"ok"}`)

Exemple :

```
curl -k https://<omnipgw_ip>:8443/api/sessions
```

Le streaming de logs n'est plus servi par la plateforme ; utilisez les logs hôtes (`journalctl`/fichiers de logs) avec les filtres ci-dessous.

Métriques Prometheus

Accès : `http://<omnipgw_ip>:9090/metrics`

Métriques Clés :

- `teid_registry_count` - Sessions actives
- `address_registry_count` - IPs UE allouées
- `sxb_inbound_errors_total` - Erreurs PFCP
- `gx_inbound_errors_total` - Erreurs Diameter Gx
- `gy_inbound_errors_total` - Erreurs Diameter Gy

Voir le Guide de Surveillance pour la référence complète des métriques.

Analyse de Logs

Logs : Inspectez les logs hôtes (`journalctl -u omnipgw_c` ou le fichier de log configuré) et grep pour les chaînes ci-dessous. Pour une vue structurée et interrogeable des événements du cycle de vie des sessions, utilisez `GET /api/session_history[?type=&search=&page=&page_size=]`.

Filtres de Logs Courants :

- "create_session_request" - Établissement de session
- "Credit Control" - Interactions Gx/Gy
- "PFCP Session" - Programmation du plan utilisateur
- "error" ou "ERROR" - Messages d'erreur
- "timeout" - Problèmes de délai d'attente

Problèmes d'Établissement de Session

Problème : Requête de Création de Session Rejetée avec "Aucune Ressource Disponible"

Symptôme :

- SGW-C reçoit une Réponse de Création de Session avec la cause "Aucune ressource disponible" (73)
- Tous les nouveaux essais de session échouent
- Les sessions existantes continuent de fonctionner
- Logs : `[PGW-C] Requête de Création de Session bloquée - licence invalide`

Capture Wireshark montrant la Réponse de Création de Session avec la cause "Aucune ressource disponible"

Cause Probable :

- Licence OmniPGW invalide ou expirée
- Serveur de licence injoignable

Diagnostic :

1. **Vérifiez la métrique de licence :**

   ```
   license_status
   ```

   - Une valeur de 0 indique une licence invalide

2. **Vérifiez les logs pour les avertissements de licence :**

   - Recherchez "license" ou "License"
   - Recherchez les messages "Impossible de contacter le serveur de licence"

3. **Vérifiez la connectivité du serveur de licence :**

   - Vérifiez l'URL configurée dans `config/runtime.exs` sous `:license_client`
   - Par défaut : `https://localhost:10443/api`

Résolution :

1. **Vérifiez que le serveur de licence est joignable :**

   ```
   curl -k https://<license_server_ip>:10443/api/status
   ```

2. **Vérifiez la configuration de la licence dans `config/runtime.exs` :**

   ```
   config :license_client,
     license_server_api_urls:
   ["https://<license_server_ip>:10443/api"],
     licensee: "Votre Nom de Société"
   ```

3. **Vérifiez que le produit est sous licence :**

   - Nom du produit : `omnipgwc`
   - Contactez Omnitouch pour vérifier l'état de la licence

4. **Redémarrez OmniPGW après les modifications de configuration**

Prévention :

- Surveillez la métrique `license_status` avec des alertes critiques
- Assurez-vous de la haute disponibilité du serveur de licence
- Mettez en place des alertes d'expiration de licence avant l'expiration

---

Problème : Requête de Création de Session Rejetée (Autres Causes)

Symptôme :

- SGW-C reçoit une Réponse de Création de Session avec une cause d'erreur
- Les utilisateurs ne peuvent pas établir de connexions PDN
- Métrique : `s5s8_inbound_errors_total` en augmentation

Causes Probables :

1. Pool IP épuisé
2. PCRF (Gx) injoignable ou rejetant la politique
3. PGW-U (PFCP) indisponible
4. Configuration APN invalide

Diagnostic :

1. **Vérifiez l'utilisation du pool IP :**

   ```
   address_registry_count
   ```

   - Si égal à la taille du pool configuré, le pool est épuisé

2. **Vérifiez la connectivité PCRF :**

   - Interrogez `GET /api/diameter` (ou `GET /api/diameter?status=disconnected`)
   - Recherchez le statut du pair PCRF = "disconnected"

- Logs : Recherchez "Credit Control Answer" pour des erreurs

3. **Vérifiez le statut du pair PFCP :**
   - Interrogez `GET /api/upf` (ou `GET /api/upf/<ip>` pour un PGW-U spécifique)
   - Recherchez "Association : DOWN"
   - Métrique : `pfcp_peer_associated` = 0
4. **Vérifiez la configuration APN :**
   - Examinez `config/runtime.exs` sous `ue.apn_map`
   - Vérifiez que l'APN demandé existe dans la configuration

Résolution :

Pour l'Épuisement du Pool IP :

1. Identifiez les sessions obsolètes : interrogez `GET /api/sessions` (optionnellement `GET /api/sessions?search=<imsi|msisdn|ip>`), recherchez les anciennes sessions
2. Étendez le pool IP dans `config/runtime.exs` :

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => "10.0.0.0/23"  # Changé de /24 à /23
(double la capacité)
    }
  }
```

3. Redémarrez OmniPGW
4. Vérifiez : `curl http://<ip>:9090/metrics | grep address_registry_count`

Pour les Problèmes de Connectivité PCRF :

1. Vérifiez la connectivité réseau : `ping <pcrf_ip>`
2. Vérifiez le service Diameter PCRF : `telnet <pcrf_ip> 3868`
3. Vérifiez la configuration du pair Diameter dans `config/runtime.exs`

4. Redémarrez OmniPGW si la configuration a changé
5. Vérifiez via `GET /api/diameter?status=connected` (le pair devrait afficher "connected")

Pour les Problèmes PFCP :

- Voir la section Problèmes PFCP / Plan Utilisateur

Prévention :

- Surveillez l'utilisation du pool IP avec des alertes à 80%
- Surveillez la connectivité PCRF avec des alertes de pair Diameter
- Mettez en œuvre un nettoyage des sessions pour les sessions inactives

---

Problème : Sessions Bloquées dans un État Intermédiaire

Symptôme :

- La session apparaît dans `GET /api/sessions` mais est incomplète
- Les métriques montrent un nombre de sessions croissant mais pas de trafic utilisateur
- La Requête de Suppression de Session échoue ou expire

Causes Probables :

1. Établissement de Session PFCP échoué mais session S5/S8 créée
2. CCR-Initial PCRF expiré
3. Requête de Création de Porteuse (porteuse dédiée) échouée
4. Disruption réseau pendant l'établissement de session

Diagnostic :

1. **Recherchez la session via l'API OAM :**
   - `GET /api/sessions/<imsi>` (ou `GET /api/sessions?search=<imsi>`)
   - Vérifiez si `pfcp_seid` est présent (s'il manque, PFCP a échoué)

- Vérifiez si `gx_session_id` est présent (s'il manque, Gx a échoué)

2. **Vérifiez les logs pour l'IMSI :**
   - Filtrez les logs par IMSI
   - Recherchez "Session Establishment Request" (PFCP)
   - Recherchez "Credit Control Request" (Gx)
   - Recherchez des messages de délai d'attente ou d'erreur
3. **Vérifiez les métriques :**

```
# Sessions avec TEID mais pas de session PFCP
teid_registry_count - seid_registry_count

# Sessions avec TEID mais pas de session Gx
teid_registry_count - session_id_registry_count
```

Résolution :

1. **Pour les échecs d'établissement PFCP :**
   - Vérifiez la santé et les logs de PGW-U
   - Vérifiez l'association PFCP : `GET /api/upf`
   - Envoyez une Requête de Suppression de Session de SGW-C pour nettoyer
2. **Pour les problèmes de délai d'attente Gx :**
   - Vérifiez la latence PCRF : `histogram_quantile(0.95, rate(gx_outbound_transaction_duration_bucket[5m]))`
   - Augmentez le délai d'attente Gx dans `config/runtime.exs` si nécessaire
   - Envoyez une Requête de Suppression de Session pour nettoyer
3. **Nettoyage manuel (dernier recours) :**
   - Actuellement, nécessite un redémarrage d'OmniPGW pour effacer les sessions bloquées

- Surveillez `teid_registry_count` avant/après le redémarrage pour confirmer le nettoyage

Prévention :

- Surveillez les métriques de latence PFCP et Gx
- Mettez en œuvre un délai d'attente/nettoyage des sessions pour les sessions incomplètes
- Alertez sur les incohérences de compte de registre

Problèmes PFCP / Plan Utilisateur

Problème : Association PFCP Hors Ligne

Symptôme :

- `GET /api/upf` montre "Association : DOWN"
- Tous les nouveaux établissements de session échouent
- Métrique : `pfcp_peer_associated` = 0
- Logs : "Délai d'attente de cœur PFCP" ou "Échec de l'établissement de l'association"

Causes Probables :

1. PGW-U injoignable (problème réseau)
2. PGW-U planté ou redémarré
3. Incompatibilité de configuration PFCP (IP, port)
4. Pare-feu bloquant UDP 8805

Diagnostic :

1. **Vérifiez la connectivité réseau :**

```
ping <pgw_u_ip>
nc -u -v <pgw_u_ip> 8805
```

2. **Vérifiez la configuration PFCP :**
    - Examinez `config/runtime.exs` sous `upf.peer_list`
    - Vérifiez que l'adresse IP et l'ID de nœud correspondent à la configuration PGW-U
3. **Vérifiez l'état de PGW-U :**
    - Accédez aux logs de PGW-U
    - Vérifiez que PGW-U fonctionne : `systemctl status omnipgw_u` (ou équivalent)
4. **Vérifiez les métriques :**

```
# Échecs de cœur
pfcp_consecutive_heartbeat_failures

# Taux d'erreur PFCP
rate(sxb_inbound_errors_total[5m])
```

Résolution :

1. **Pour les problèmes réseau :**
    - Vérifiez le routage : `traceroute <pgw_u_ip>`
    - Vérifiez les règles de pare-feu : Assurez-vous que UDP 8805 est autorisé
    - Vérifiez les groupes de sécurité (si déploiement cloud)
2. **Pour les plantages de PGW-U :**
    - Redémarrez le service PGW-U
    - Attendez 30 secondes pour la réétablissement de l'association
    - Vérifiez via `GET /api/upf` (devrait montrer "Association : UP")
3. **Pour les problèmes de configuration :**
    - Corrigez la configuration du pair PFCP dans `config/runtime.exs`
    - Redémarrez OmniPGW
    - Vérifiez que l'association est établie

Prévention :

- Surveillez la métrique `pfcp_peer_associated` avec des alertes critiques
- Surveillez `pfcp_consecutive_heartbeat_failures` (alerte à > 2)
- Mettez en œuvre des instances redondantes de PGW-U
- Activez le keepalive/heartbeat PFCP (devrait être par défaut)

Problème : Échecs de Modification de Session PFCP

Symptôme :

- La création de porteuse dédiée échoue
- Les mises à jour de politique QoS (de PCRF RAR) échouent
- Logs : "Échec de la Requête de Modification de Session"
- Métrique : `sxb_inbound_errors_total{message_type="session_modification_response"}` en augmentation

Causes Probables :

1. Règles PFCP invalides (références PDR/FAR/QER)
2. Épuisement des ressources PGW-U
3. Conflits d'ID de règle
4. Bug logiciel PGW-U

Diagnostic :

1. **Vérifiez les logs :**
    - Filtrez pour "Session Modification" et SEID
    - Recherchez des codes de cause d'erreur dans la réponse PFCP
    - Causes courantes : "L'ID de règle existe déjà", "Ressources épuisées"
2. **Vérifiez les logs de PGW-U :**
    - Recherchez des erreurs de traitement PFCP

- Vérifiez l'utilisation des ressources (CPU, mémoire)

3. **Vérifiez l'état de la session via l'API OAM :**
   - `GET /api/sessions/<imsi>` (ou `GET /api/sessions?search=<imsi>`)
   - Examinez `pdr_map`, `far_map`, `qer_map` pour des conflits
   - Recherchez des ID en double

Résolution :

1. **Pour les conflits de règles :**
   - Supprimez et recréez la porteuse dédiée
   - Si persistant, supprimez la session et faites reconnecter l'UE
2. **Pour les problèmes de ressources PGW-U :**
   - Vérifiez la capacité de PGW-U (sessions, PDRs, débit)
   - Élargissez PGW-U si nécessaire
   - Réduisez la charge de session sur l'instance PGW-U affectée
3. **Pour les bugs logiciels :**
   - Capturez l'état complet de la session (`GET /api/sessions/<imsi>`)
   - Capturez les logs des messages PFCP
   - Signalez au fournisseur avec des étapes de reproduction

Prévention :

- Surveillez l'utilisation des ressources PGW-U
- Testez la création de porteuse dédiée en staging
- Surveillez `sxb_inbound_errors_total` avec des alertes

Problèmes Diameter (Gx/Gy)

Problème : Pair PCRF Déconnecté (Gx)

Symptôme :

- `GET /api/diameter` (ou `GET /api/diameter?status=disconnected`) montre le pair PCRF "déconnecté"
- Sessions créées sans politiques QoS (QCI par défaut=5 appliqué)
- Logs : "Échec de la connexion au pair Diameter" ou "Délai d'attente CER/CEA"

Causes Probables :

1. PCRF injoignable (problème réseau)
2. Service PCRF hors service
3. Incompatibilité de configuration Diameter (Origin-Host, Realm)
4. Pare-feu bloquant TCP 3868

Diagnostic :

1. **Vérifiez la connectivité réseau :**

   ```
   ping <pcrf_ip>
   telnet <pcrf_ip> 3868
   ```

2. **Vérifiez la configuration Diameter :**

   - Examinez `config/runtime.exs` sous `diameter.peer_list`
   - Vérifiez que `host`, `realm`, `ip` correspondent à la configuration PCRF
   - Vérifiez que `origin_host` correspond à ce que PCRF attend

3. **Vérifiez les logs PCRF :**

   - Recherchez CER (Capabilities-Exchange-Request) de PGW-C
   - Recherchez les raisons de rejet

4. **Vérifiez les métriques :**

```
# Erreurs de connexion Diameter
diameter_peer_connected{peer="<pcrf_host>"}
```

Résolution :

1. **Pour les problèmes réseau :**
   - Vérifiez le routage vers PCRF
   - Vérifiez les règles de pare-feu : Assurez-vous que TCP 3868 est autorisé
   - Testez la connectivité : `nc -v <pcrf_ip> 3868`
2. **Pour le service PCRF hors service :**
   - Redémarrez le service PCRF
   - Attendez la reconnexion automatique (intervalle de réessai de 30s)
   - Vérifiez via `GET /api/diameter?status=connected`
3. **Pour l'incompatibilité de configuration :**
   - Corrigez la configuration Diameter dans `config/runtime.exs` :

```
config :pgw_c,
  diameter: %{
    host: "pgw-c.epc.mnc999.mcc999.3gppnetwork.org",  #
Doit correspondre à la config PCRF
    realm: "epc.mnc999.mcc999.3gppnetwork.org",
    peer_list: [
      %{
        host: "pcrf.epc.mnc999.mcc999.3gppnetwork.org",
        realm: "epc.mnc999.mcc999.3gppnetwork.org",
        ip: "192.168.1.100",
        initiate_connection: true
      }
    ]
  }
```

   - Redémarrez OmniPGW
   - Vérifiez que la connexion est établie

Prévention :

- Surveillez la connectivité des pairs Diameter avec des alertes critiques
- Mettez en œuvre des instances PCRF redondantes (si supportées)
- Documentez la configuration Diameter dans le runbook

Problème : Délai d'Attente CCR/CCA (Demandes de Politique Gx)

Symptôme :

- Établissement de session lent (> 5 secondes)
- Logs : "Délai d'attente de la Demande de Contrôle de Crédit"
- Métrique : `gx_outbound_transaction_duration` très élevée (> 5s)
- Sessions créées avec QoS par défaut (comportement de secours)

Causes Probables :

1. PCRF surchargé
2. Base de données PCRF lente
3. Latence réseau
4. Problème logiciel PCRF

Diagnostic :

1. **Vérifiez la latence Gx :**

```
# Latence P95
histogram_quantile(0.95,
rate(gx_outbound_transaction_duration_bucket[5m]))

# Latence P99 (valeurs aberrantes)
histogram_quantile(0.99,
rate(gx_outbound_transaction_duration_bucket[5m]))
```

2. **Vérifiez la santé de PCRF :**
   - Accédez aux tableaux de bord de surveillance PCRF
   - Vérifiez CPU, mémoire, connexions à la base de données

- Examinez les logs PCRF pour des requêtes lentes

3. **Vérifiez la latence réseau :**

```
ping -c 100 <pcrf_ip> | tail -1  # Vérifiez la latence moyenne
```

4. **Vérifiez les logs :**

- Comptez les échanges CCR/CCA : Filtrez "Credit Control"
- Mesurez le temps entre "Envoi CCR" et "Reçu CCA"

Résolution :

1. **Pour la surcharge de PCRF :**

- Élargissez PCRF (ajoutez des instances)
- Réduisez la taille des messages CCR si possible
- Ajustez les pools de threads/travailleurs de PCRF

2. **Pour la latence réseau :**

- Enquêtez sur le chemin réseau (routeurs, commutateurs)
- Envisagez de co-localiser PGW-C et PCRF

3. **Solution temporaire (augmenter le délai d'attente) :**

- Modifiez `config/runtime.exs` :

```
config :pgw_c,
  diameter: %{
    transaction_timeout_ms: 10000  # Augmentez de 5000 par
défaut
  }
```

- Redémarrez OmniPGW
- **Remarque :** Cela masque seulement le problème ; corrigez la cause profonde

Prévention :

- Surveillez la latence Gx avec des alertes (avertissement > 1s, critique > 5s)
- Planifiez la capacité de PCRF pour le taux de session attendu
- Testez les performances de PCRF sous charge

Problème : Pair OCS Déconnecté (Gy)

Symptôme :

- `GET /api/diameter` (ou `GET /api/diameter?status=disconnected`) montre le pair OCS "déconnecté" ; `GET /api/charging` reflète le statut de la facturation en ligne (Gy)
- Les sessions ne peuvent pas être facturées (la facturation en ligne échoue)
- Logs : "Échec de la connexion au pair Gy"

Diagnostic et Résolution :

Similaire à Pair PCRF Déconnecté, mais pour l'interface Gy.

Différences Clés :

- Port : Généralement TCP 3868 (même que Gx)
- Impact : La facturation échoue, les sessions peuvent être rejetées ou autorisées sans facturation (dépend de la config)
- Configuration : Vérifiez `diameter.peer_list` pour l'entrée OCS

Voir : Interface Diameter Gy pour le dépannage spécifique à Gy

---

Problèmes d'Allocation IP

Problème : Pool IP Épuisé

Symptôme :

- Requête de Création de Session rejetée avec la cause "Aucune ressource disponible"
- Métrique : `address_registry_count` égale à la taille du pool configuré
- `GET /api/sessions` montre de nombreuses sessions actives ; `GET /api/ip_pools` montre le pool à capacité
- Logs : "Échec de l'allocation IP : pool épuisé"

Causes Probables :

1. Pool trop petit pour la base d'abonnés
2. Sessions ne libérant pas les IP (échecs de Suppression de Session)
3. Churn rapide de sessions sans nettoyage
4. Fuite d'adresse IP

Diagnostic :

1. **Vérifiez l'utilisation du pool :**

```
# Pour un sous-réseau /24 (254 IPs)
(address_registry_count / 254) * 100
```

2. **Vérifiez la taille du pool configuré :**
   - Examinez `config/runtime.exs` sous `ue.subnet_map`
   - Exemple : "10.0.0.0/24" = 254 IPs utilisables
3. **Comparez le nombre de sessions au nombre d'IP :**

```
# Devrait être à peu près égal
teid_registry_count
address_registry_count
```

4. **Examinez les sessions actives :**
   - `GET /api/sessions` (et `GET /api/ip_pools/<name>` pour l'utilisation par pool)
   - Triez par heure de début de session

- Recherchez des sessions très anciennes (fuites potentielles)

Résolution :

Immédiate (étendre le pool) :

1. Modifiez `config/runtime.exs` (chaque APN mappe à une **liste** de chaînes CIDR) :

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => ["10.0.0.0/22"]  # 1022 IPs (était /24 =
254 IPs)
    }
  }
```

2. Redémarrez OmniPGW
3. Vérifiez : Les sessions peuvent maintenant s'établir

À Long Terme (nettoyage) :

1. Identifiez les sessions obsolètes via `GET /api/sessions`
2. Coordonnez-vous avec SGW-C pour envoyer des Requêtes de Suppression de Session
3. Mettez en œuvre une politique de délai d'attente de session sur PCRF/SGW
4. Surveillez `address_registry_count` pour vérifier que le pool a été libéré après nettoyage

Prévention :

- Surveillez l'utilisation du pool IP avec des alertes :
  - Avertissement : > 70%
  - Critique : > 85%
- Analyse de tendance pour prédire l'épuisement
- Mettez en œuvre un délai d'attente pour les sessions inactives
- Audits réguliers des sessions

Problème : Adresse IP Dupliquée Assignée

Symptôme :

- UE signale un conflit d'adresse IP
- Logs : Avertissement "IP déjà allouée"
- Deux sessions dans `GET /api/sessions` avec la même adresse IP

Causes Probables :

1. Bug logiciel (rare)
2. Incohérence de base de données après un crash
3. Erreur d'intervention manuelle

Diagnostic :

1. **Recherchez l'IP via l'API OAM :**
    - `GET /api/sessions?search=<ip>`
    - Vérifiez si plusieurs IMSIs ont la même IP
2. **Vérifiez les logs :**
    - Recherchez l'adresse IP
    - Recherchez des événements "allocation IP"

Résolution :

1. **Identifiez les sessions affectées :**
    - Notez les deux IMSIs avec l'IP dupliquée
2. **Supprimez une session :**
    - Coordonnez-vous avec SGW-C pour envoyer une Requête de Suppression de Session pour un IMSI
    - Préférez supprimer la session la plus récente
3. **Reconnectez l'UE :**
    - L'UE devrait se reconnecter automatiquement

- Recevra une nouvelle IP unique

4. **Si persistant :**
   - Redémarrez OmniPGW pour reconstruire le registre IP
   - Toutes les sessions seront perdues (coordonnez la fenêtre de maintenance)

Prévention :

- Surveillez les allocations dupliquées (aucune métrique intégrée actuellement)
- Vérifications régulières de l'intégrité de la base de données (si applicable)

Problème : Session IPv4v6 sur un APN Sans Pool IPv6

Symptôme :

- Un UE demande un PDN **IPv4v6** mais l'APN correspondant n'a qu'un CIDR IPv4 configuré dans `ue.subnet_map`
- Un APN **d'urgence (SOS)** se présente **IPv4 uniquement** (pas d'IPv6 dans le PAA) ; les logs montrent un `avertissement` comme "Pas de pool IPv6 pour l'APN d'urgence ... ; accordant IPv4 uniquement"
- Un APN **régulier** a la session **rejetée** ; les logs montrent une `erreur` comme "Pas de pool IPv6 pour l'APN ... ; rejet de la demande IPv4v6"

Causes Probables :

1. L'APN n'a intentionnellement pas de plage IPv6 (commun sur les déploiements IPv4 uniquement)
2. Un CIDR IPv6 a été omis de l'entrée `ue.subnet_map` de l'APN par erreur

Comportement :

- L'allocateur IPv6 signale qu'il n'y a pas de pool IPv6 (il ne plante pas sur un pool vide).

- Les APNs d'urgence (le nom contient `"sos"`) se dégradent en IPv4 uniquement — adresse IPv4 conservée, IPv6 effacée, type PDN défini sur IPv4, et le P-CSCF est toujours retourné dans le PCO, donc l'enregistrement IMS d'urgence fonctionne.
- Les APNs réguliers sont rejetés (pas de dégradation silencieuse), faisant surface la demande non prise en charge / la mauvaise configuration.

Résolution :

1. Si l'APN doit offrir un double empilement, ajoutez un CIDR IPv6 à son entrée `ue.subnet_map` et redémarrez OmniPGW.
2. Si le comportement IPv4 uniquement est intentionnel (par exemple, un APN SOS sur un réseau IPv4 uniquement), aucune action n'est requise.

Voir Demande IPv4v6 lorsque l'APN n'a pas de pool IPv6 et Allocation IP UE → Problème 4.

Référence Rapide

Requêtes Prometheus Courantes

```
# Sessions actives
teid_registry_count

# Taux de configuration de session (par seconde)
rate(s5s8_inbound_messages_total{message_type="create_session_request

# Utilisation du pool IP (pour un sous-réseau /24)
(address_registry_count / 254) * 100

# Latence de configuration de session P95
histogram_quantile(0.95,
rate(s5s8_inbound_handling_duration_bucket{request_message_type="crea
[5m]))

# Taux d'erreur
rate(s5s8_inbound_errors_total[5m])

# Latence PCRF
histogram_quantile(0.95, rate(gx_outbound_transaction_duration_bucket

# Statut d'association PFCP
pfcp_peer_associated
```

Filtres de Logs Courants

Grep ces chaînes dans les logs hôtes (`journalctl -u omnipgw_c` ou le fichier de log configuré). Pour les événements du cycle de vie des sessions, `GET /api/session_history[?type=&search=]` fournit les mêmes informations sous une forme structurée.

Filtre	But
IMSI	Trouver tous les logs pour un abonné spécifique
"create_session"	Flux d'établissement de session
"delete_session"	Flux de démantèlement de session
"Credit Control"	Interactions Gx PCRF
"PFCP Session"	Programmation du plan utilisateur
"error"	Tous les messages d'erreur
"timeout"	Problèmes de délai d'attente
"Association"	Événements d'association PFCP

Commandes de Vérification de Santé

```
# Vérifiez l'état du service
systemctl status omnipgw_c

# Vérifiez la santé de l'API OAM (HTTPS/TLS)
curl -k https://<omnipgw_ip>:8443/api/status

# Vérifiez le point de terminaison des métriques
curl http://<omnipgw_ip>:9090/metrics

# Vérifiez les sessions actives
curl http://<omnipgw_ip>:9090/metrics | grep teid_registry_count

# Vérifiez l'association PFCP
curl http://<omnipgw_ip>:9090/metrics | grep pfcp_peer_associated

# Vérifiez l'utilisation du pool IP
curl http://<omnipgw_ip>:9090/metrics | grep
address_registry_count
```

Documentation Connexe

- **Guide de Surveillance** - Métriques Prometheus, tableaux de bord Grafana, alertes
- **Guide de Configuration** - Référence de configuration système
- **Gestion des Sessions** - Détails du cycle de vie des sessions
- **Interface PFCP** - Détails de dépannage PFCP
- **Interface Diameter Gx** - Dépannage de la politique Gx
- **Interface Diameter Gy** - Dépannage de la facturation Gy
- **Gestion de QoS et de Porteurs** - Problèmes liés à la QoS

Retour au Guide des Opérations

Documentation sur l'Allocation de Pool IP UE

Gestion des adresses IP pour les appareils mobiles

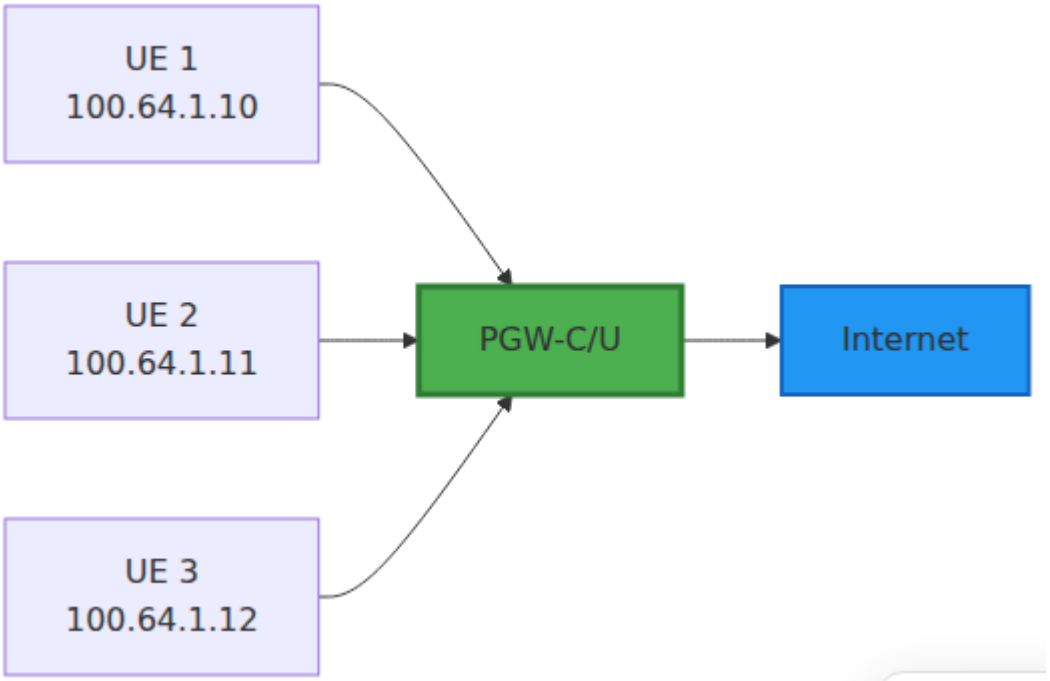
Table des Matières

1. Aperçu
2. Concepts d'Allocation IP
3. Configuration
4. Processus d'Allocation
5. Sujets Avancés
6. Surveillance
7. Dépannage

Aperçu

Le PGW-C alloue des adresses IP aux appareils UE (Équipement Utilisateur) lorsqu'ils établissent des connexions PDN (Packet Data Network). C'est une fonction critique qui permet aux appareils mobiles de communiquer avec des réseaux externes.

Pourquoi l'Allocation IP est Importante



Chaque UE reçoit une **adresse IP unique** du PGW-C qui :

- Identifie l'appareil sur le réseau
- Achemine le trafic vers/depuis l'appareil
- Permet la facturation et l'application des politiques
- Persiste pendant la durée de la connexion PDN

Versions IP Supportées

Version IP	Support	Description
IPv4	□ Complet	Adresses IPv4 standard
IPv6	□ Complet	Adresses et préfixes IPv6
IPv4v6	□ Complet	Dual-stack (IPv4 et IPv6)

Concepts d'Allocation IP

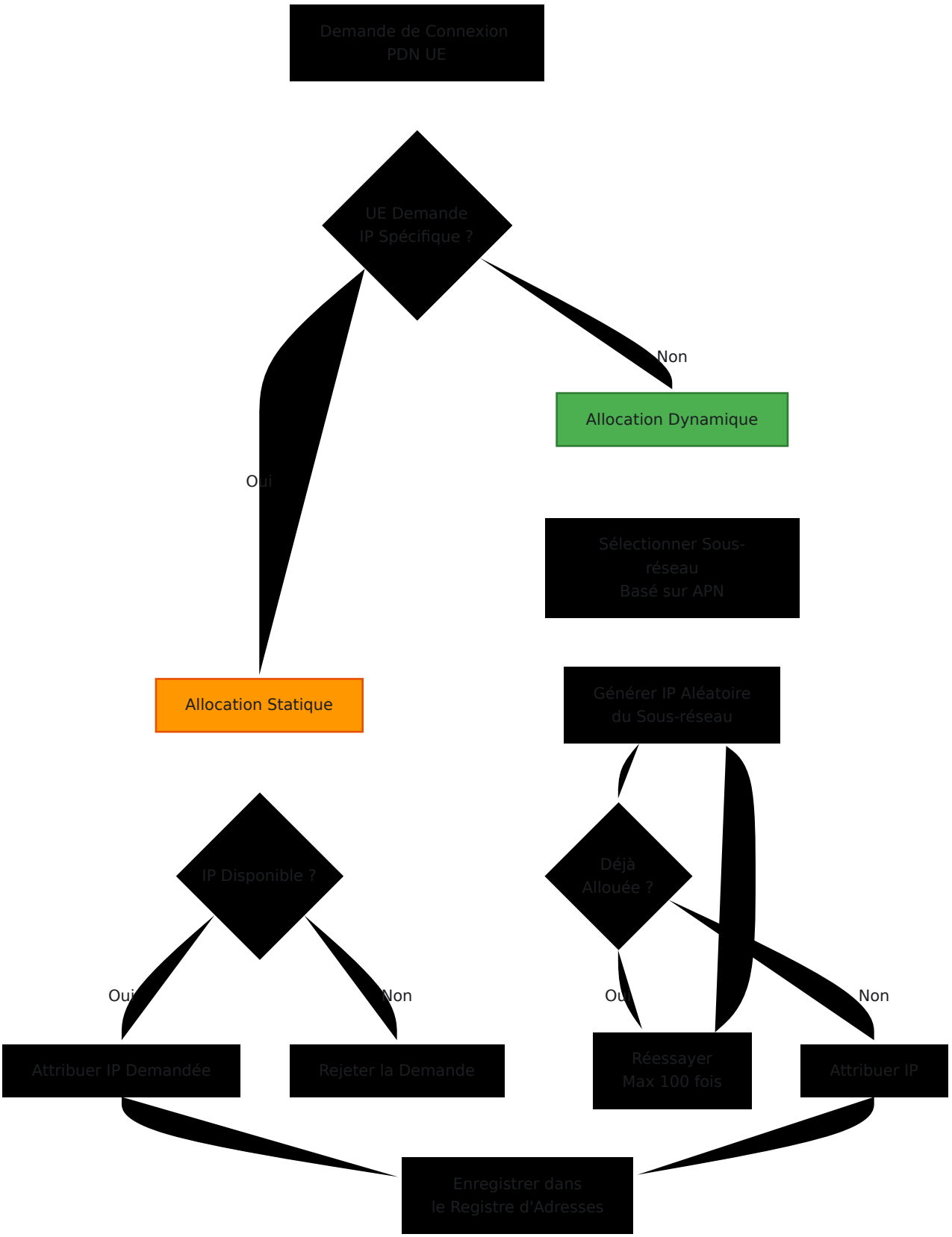
Type de PDN

Lorsqu'un UE demande une connexion PDN, il spécifie un **Type de PDN** :

Type de PDN	Description	Adresses Allouées
IPv4	Connexion uniquement IPv4	Adresse IPv4 unique
IPv6	Connexion uniquement IPv6	Préfixe IPv6 (par ex., /64)
IPv4v6	Connexion dual-stack	Adresse IPv4 et préfixe IPv6

Méthodes d'Allocation

Le PGW-C prend en charge deux méthodes d'allocation IP :



1. Allocation Dynamique (la Plus Courante) :

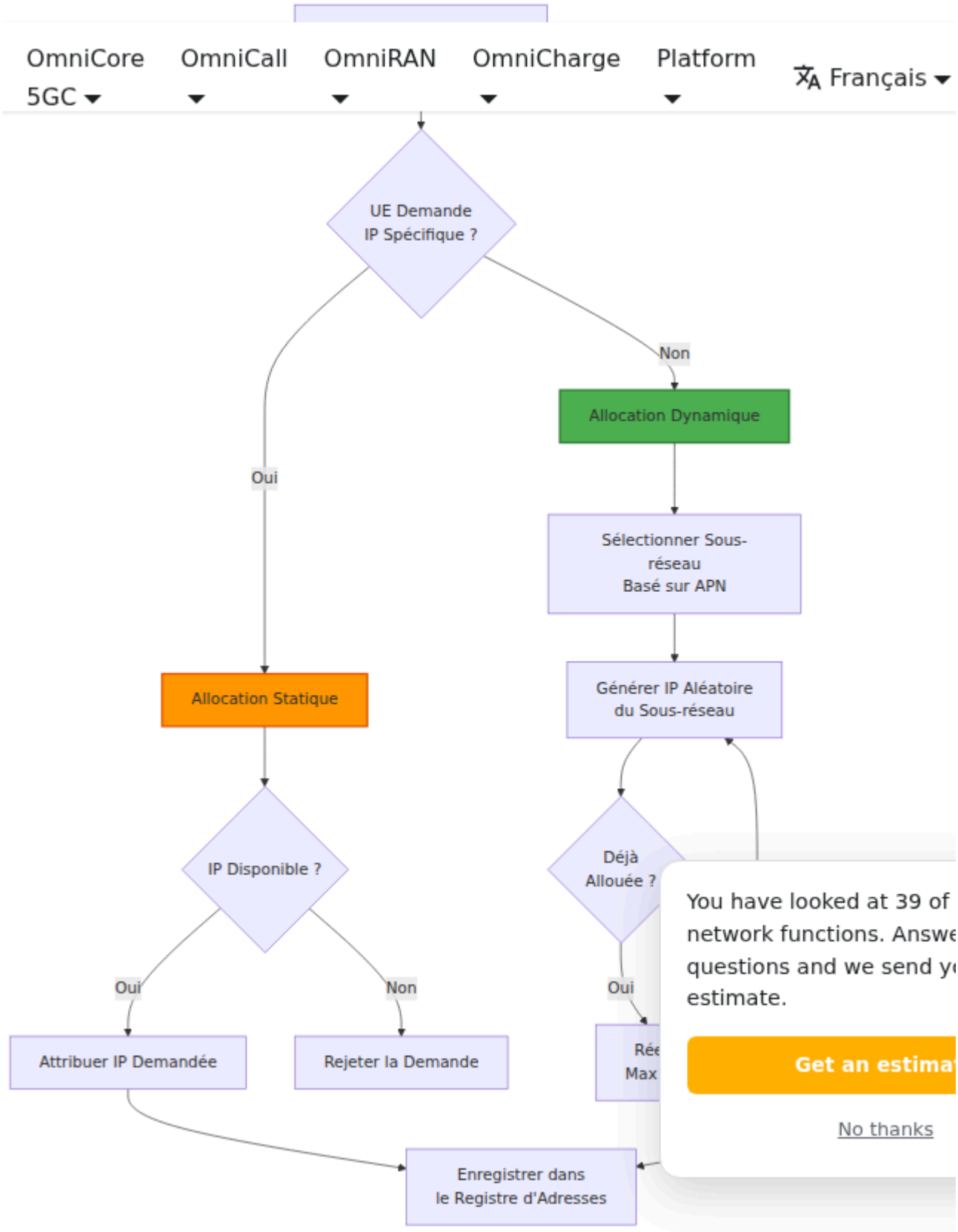
- Le PGW-C sélectionne une IP dans le pool configuré
- Sélection aléatoire pour éviter la prévisibilité
- Détection de collision pour garantir l'unicité

2. Allocation Statique :

- L'UE demande une IP spécifique dans le message GTP-C
- Le PGW-C valide la disponibilité
- Utile pour les appareils d'entreprise avec des IP fixes

Sélection de Sous-réseau Basée sur l'APN

Différents **APN (Noms de Point d'Accès)** peuvent utiliser différents pools IP :



Avantages :

- **Séparation du Trafic** - Différents APN acheminent vers différents réseaux
- **Différenciation des Politiques** - Appliquer différentes politiques par APN

- **Planification de Capacité** - Dimensionner les pools en fonction de l'utilisation prévue
- **Facturation** - Suivre l'utilisation par type de service

Registre d'Adresses

Le **Registre d'Adresses** suit les IP allouées :

Fonction	Description
Enregistrement	Map UE IP → PID du Processus de Session
Recherche	Trouver la session par UE IP
Désenregistrement	Libérer l'IP lorsque la session se termine
Détection de Collision	Prévenir les allocations en double

Configuration

Configuration de Base

Modifier `config/runtime.exs` :

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # L'APN "internet" utilise deux sous-réseaux
      "internet" => [
        "100.64.1.0/24",    # 254 IPs utilisables
        "100.64.2.0/24"     # 254 IPs utilisables
      ],

      # L'APN "ims" utilise un sous-réseau
      "ims" => [
        "100.64.10.0/24"
      ],

      # Pool par défaut pour les APN inconnus
      default: [
        "42.42.42.0/24"
      ]
    }
  }
```

Correspondance de Modèle Regex pour les APN

Pour les scénarios où plusieurs APN partagent le même pool de sous-réseau, vous pouvez utiliser des **modèles regex** au lieu de noms d'APN exacts. Cela est utile pour la correspondance d'APN avec joker.

Règles de Modèle :

- Les clés commençant par `^` sont traitées comme des modèles regex
- Les clés sans `^` sont correspondantes exactement (rétrocompatible)
- Les modèles sont évalués dans l'ordre - la première correspondance l'emporte
- Fallback vers `default` si aucune correspondance de modèle

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # Regex : APNs commençant par "ims" (par ex., "ims",
"ims.apn", "ims.something.else")
      "^ims" => [
        "100.64.10.0/24"
      ],

      # Regex : APNs commençant par "m2m." (par ex., "m2m.test",
"m2m.prod")
      "^m2m\." => [
        "100.64.20.0/24"
      ],

      # Correspondance exacte uniquement - "enterprise.corp"
exactement
      "enterprise.corp" => [
        "10.100.0.0/16"
      ],

      # Pool par défaut pour les APN non correspondants
      default: [
        "42.42.42.0/24"
      ]
    }
  }
```

Remarques Importantes :

- Les modèles regex utilisent la syntaxe regex standard Elixir/Erlang
- Les barres obliques doivent être échappées dans les chaînes Elixir (utilisez `\` pour `\`)
- Le `^` au début est requis pour indiquer un modèle regex
- Les correspondances exactes et les modèles regex peuvent être mélangés dans la même configuration
- L'ordre est important pour les modèles regex - placez les modèles plus spécifiques en premier

Exemples de Modèles Courants :

Type de Modèle	Clé Regex	Correspondances
Commence par	`"^ims"`	`ims`, `ims.apn`, `ims.anything`
Finit par	`"^.*\.corp$"`	`foo.corp`, `bar.corp`
Contient	`"^.*test.*"`	`test`, `foo.test.bar`, `testing`
Exact (avec des points)	`"^internet\.apn$"`	`internet.apn` uniquement

Exemple de Correspondance de Suffixe :

Pour correspondre aux APN se terminant par un suffixe spécifique (comme `.corp`), utilisez `^.*\.suffix$` :

```
subnet_map: %{
  # Correspondre aux APN se terminant par ".corp"
  "^.*\.corp$" => ["10.100.0.0/16"],

  # Correspondre aux APN se terminant par ".iot"
  "^.*\.iot$" => ["10.200.0.0/16"],

  default: ["42.42.42.0/24"]
}
```

Exemple de Correspondance de Modèle :

Demande APN	Clé Correspondante	Pool Utilisé
`ims`	`^ims`	100.64.10.0/24
`ims.apn`	`^ims`	100.64.10.0/24
`ims.something.else`	`^ims`	100.64.10.0/24
`m2m.test`	`^m2m\.`	100.64.20.0/24
`m2m`	*default*	42.42.42.0/24
`enterprise.corp`	`enterprise.corp`	10.100.0.0/16
`foo.corp`	`^.*\.corp$`	10.100.0.0/16
`unknown.apn`	*default*	42.42.42.0/24

Notation de Sous-réseau

Notation CIDR : `<network>/<prefix_length>`

CIDR	IPs Utilisables	Plage d'Exemple
/24	254	100.64.1.1 - 100.64.1.254
/23	510	100.64.0.1 - 100.64.1.254
/22	1022	100.64.0.1 - 100.64.3.254
/20	4094	100.64.0.1 - 100.64.15.254
/16	65534	100.64.0.1 - 100.64.255.254

Remarques :

- L'adresse réseau (par ex., 100.64.1.0) n'est pas allouée
- L'adresse de diffusion (par ex., 100.64.1.255) n'est pas allouée
- Le PGW-C alloue de `<network> + 1` à `<broadcast> - 1`

Plusieurs Sous-réseaux par APN

Équilibrage de Charge à Travers les Sous-réseaux :

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.1.0/24",
        "100.64.2.0/24",
        "100.64.3.0/24",
        "100.64.4.0/24"
      ]
    }
  }
```

Méthode de Sélection :

- Le PGW-C sélectionne aléatoirement un sous-réseau dans la liste
- Fournit un équilibrage de charge de base
- Chaque session sélectionne indépendamment un sous-réseau

Avantages :

- Distribuer la charge à travers plusieurs sous-réseaux
- Expansion de capacité plus facile (ajouter de nouveaux sous-réseaux)
- Flexibilité pour les politiques de routage

Exemple du Monde Réel

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # Accès général à Internet
      "internet" => [
        "100.64.0.0/20"     # 4094 IPs pour usage général
      ],

      # IMS (Voix sur LTE)
      "ims" => [
        "100.64.16.0/22"    # 1022 IPs pour IMS
      ],

      # APN d'entreprise
      "enterprise.corp" => [
        "10.100.0.0/16"     # 65534 IPs pour entreprise
      ],

      # Appareils IoT (débit faible)
      "iot.m2m" => [
        "100.64.20.0/22"    # 1022 IPs pour IoT
      ],

      # Fallback par défaut
      default: [
        "42.42.42.0/24"     # 254 IPs pour APNs inconnus
      ]
    }
  }
```

Configuration IPv6

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => [
        # Pools IPv4
        "100.64.1.0/24"
      ],
      "internet.ipv6" => [
        # Pools IPv6 (délégation de préfixe)
        "2001:db8:1::/48"
      ],
      default: [
        "42.42.42.0/24"
      ]
    }
  }
```

Délégation de Préfixe IPv6 :

- L'UE reçoit généralement un préfixe /64
- Permet à l'UE d'attribuer plusieurs IPs (par ex., pour le partage de connexion)
- Exemple : L'UE reçoit `2001:db8:1:a::/64`

Configuration Dual-Stack (IPv4v6)

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.1.0/24",        # Pool IPv4
        "2001:db8:1::/48"       # Pool IPv6 (sera utilisé pour
l'allocation IPv6)
      ]
    }
  }
```

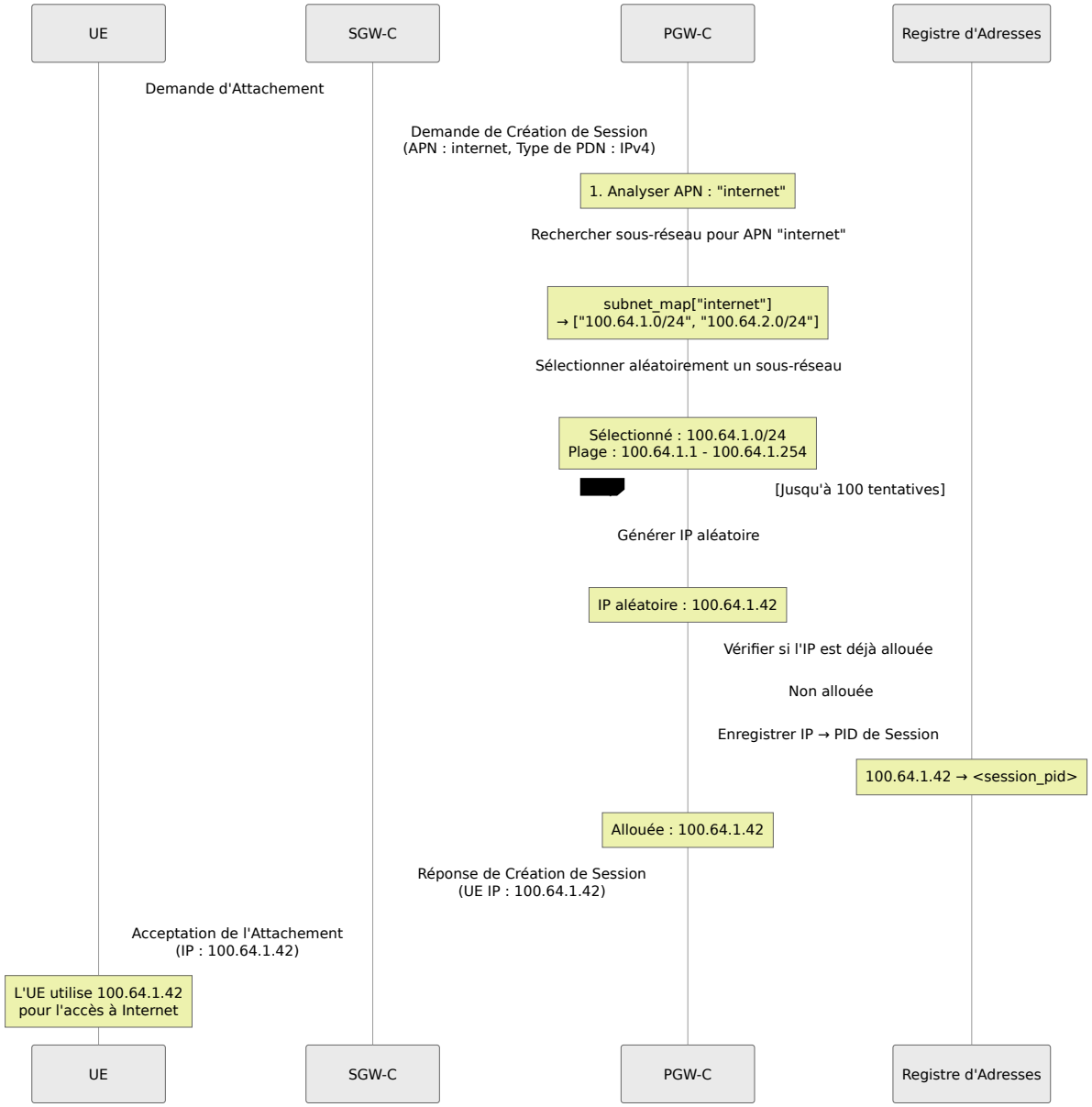
Allocation Dual-Stack :

- L'UE demande le Type de PDN : IPv4v6
- Le PGW-C alloue à la fois une adresse IPv4 et un préfixe IPv6
- Les deux adresses sont actives simultanément

Processus d'Allocation

L'allocation IP se produit lors de la création de session lorsque le PGW-C reçoit une Demande de Création de Session via l'interface S5/S8. Voir Interface S5/S8 pour les détails des messages GTP-C et Gestion de Session pour le cycle de vie de la session.

Étape par Étape : Allocation Dynamique IPv4



Comment Cela Fonctionne

Processus d'Allocation Dynamique :

1. **Recherche de Sous-réseau :** Le système récupère les sous-réseaux configurés pour l'APN demandé
2. **Sélection Aléatoire :** Un sous-réseau est sélectionné aléatoirement dans la liste disponible
3. **Génération d'IP :** Une IP aléatoire est générée dans la plage du sous-réseau

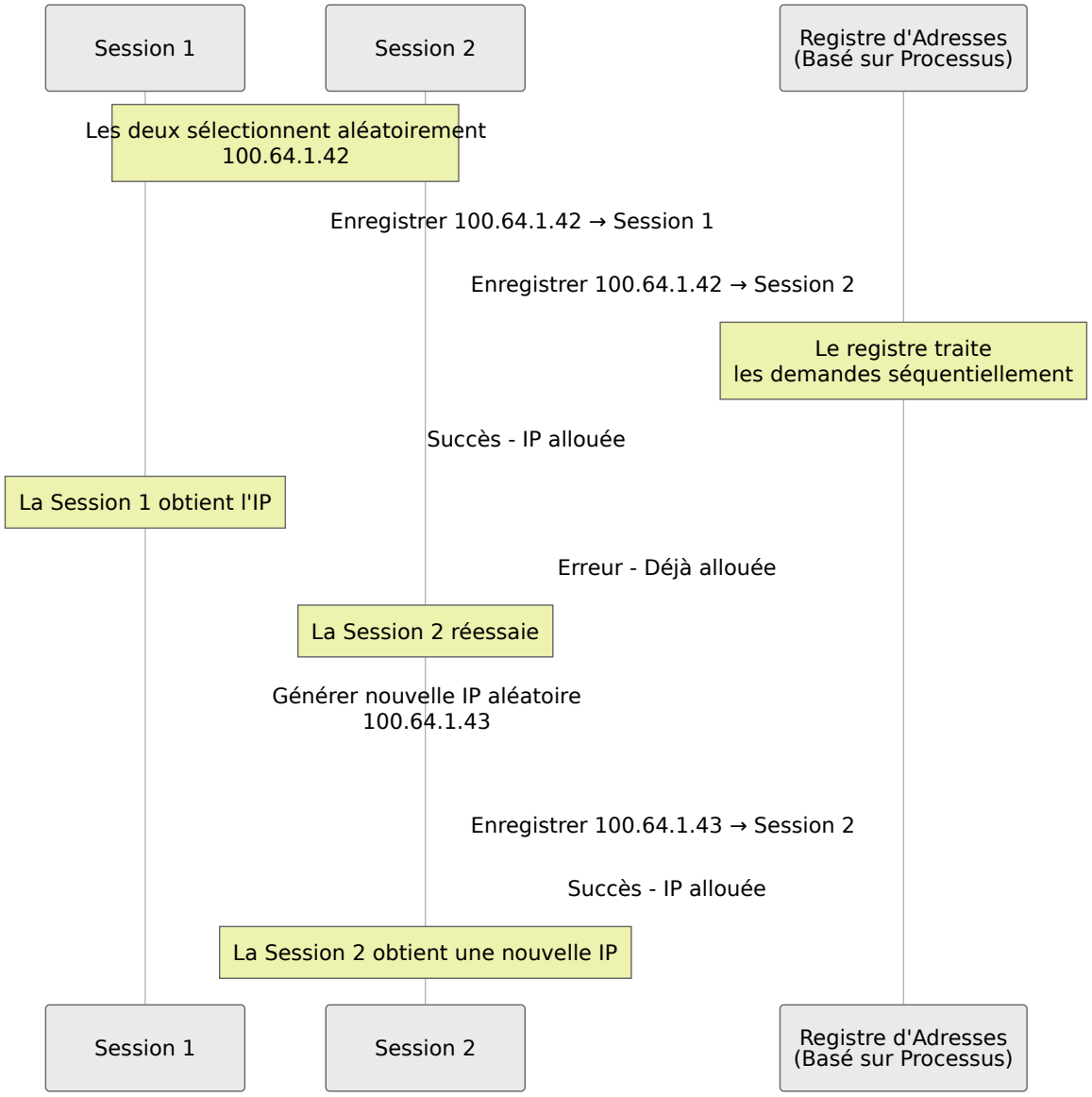
4. **Vérification d'Unicité :** Le système vérifie que l'IP n'a pas été allouée
5. **Logique de Réessai :** En cas de détection de collision, réessayer jusqu'à 100 fois avec une nouvelle IP aléatoire
6. **Enregistrement :** Une fois l'IP unique trouvée, elle est enregistrée à la session

Points de Conception Clés :

- **Max 100 tentatives :** Empêche les boucles infinies lorsque le pool est presque épuisé
- **Sélection aléatoire :** Évite les modèles d'attribution d'IP prévisibles pour la sécurité
- **Opérations atomiques :** Le registre basé sur les processus garantit qu'il n'y a pas d'allocations en double
- **Fallback vers le par défaut :** Si l'APN n'est pas trouvé dans la configuration, utilise le pool par défaut

Gestion des Collisions

Scénario : Deux sessions essaient d'allouer la même IP simultanément



Comment la Prévention des Collisions Fonctionne :

- Le registre traite les demandes une à la fois (sérialisé)
- Aucune condition de course possible
- La première demande à enregistrer une IP réussit
- Les demandes suivantes pour la même IP sont rejetées
- Les sessions rejetées réessaient automatiquement avec une nouvelle IP aléatoire

Fallback de Sous-réseau par Défaut

Scénario : L'UE demande un APN inconnu

Configuration d'Exemple :

```
# Config
subnet_map: %{
  "internet" => ["100.64.1.0/24"],
  default: ["42.42.42.0/24"]
}
```

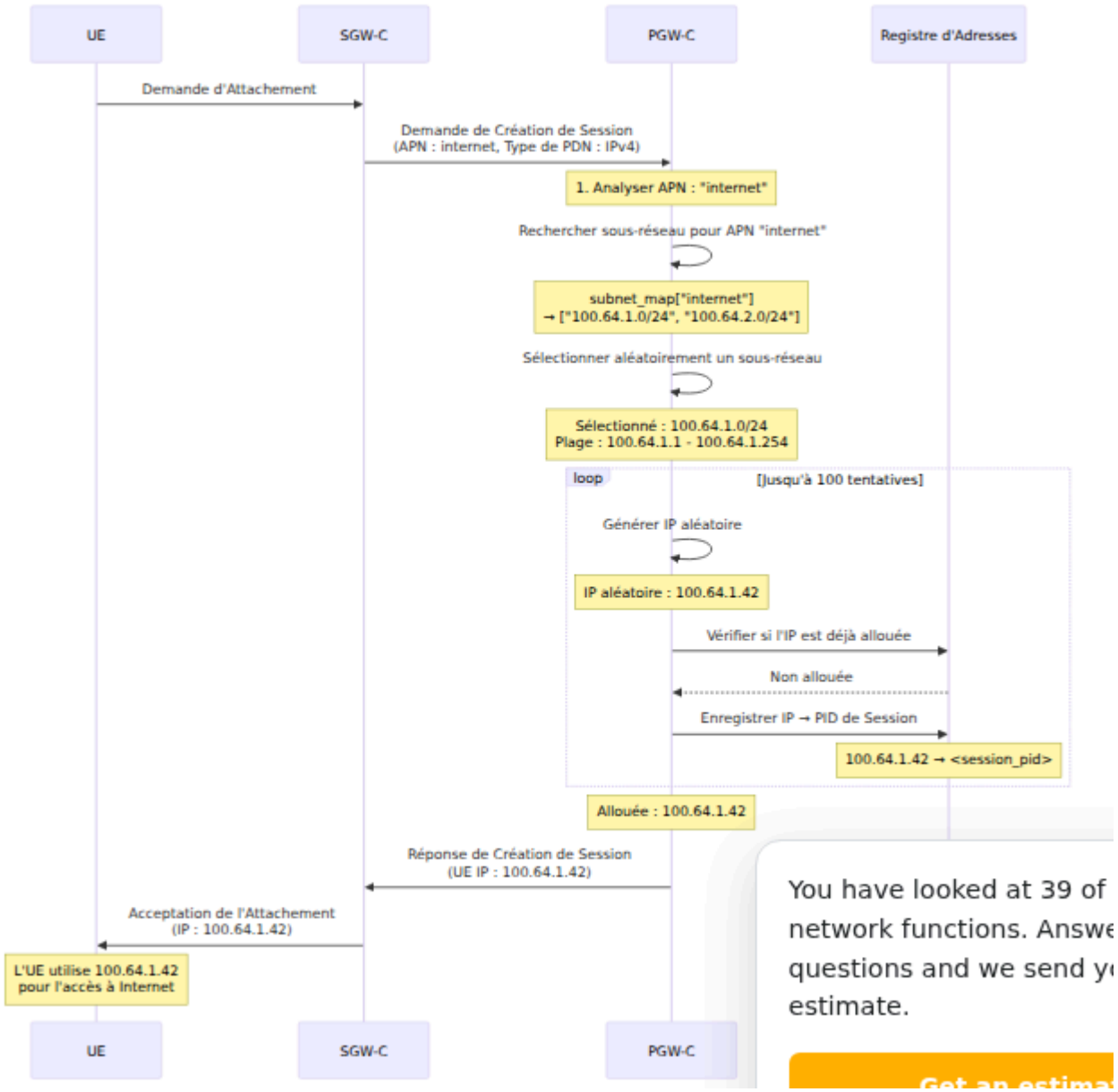
Comportement :

- L'UE demande APN : "unknown.apn"
- Le système recherche "unknown.apn" dans subnet_map
- Non trouvé, donc revient au pool par défaut
- Alloue une IP du 42.42.42.0/24

Logique de Fallback :

1. D'abord, essayer de trouver le pool spécifique à l'APN dans la configuration
2. Si non trouvé, utiliser le pool `default`
3. Si aucun défaut configuré, l'allocation échoue

Désallocation à la Fin de Session



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

Nettoyage Automatique :

- Lorsque le processus de session se termine, le registre nettoie
- L'IP est immédiatement disponible pour de nouvelles allocations
- Aucune intervention manuelle requise

Sujets Avancés

Épuisement du Pool

Scénario : Toutes les IPs dans le pool sont allouées

```
Pool : 100.64.1.0/24 (254 IPs utilisables)
Allouées : 254 IPs
Nouvelle demande arrive → Épuisement
```

Que se passe-t-il :

1. Le PGW-C tente 100 allocations aléatoires
2. Toutes les tentatives trouvent l'IP déjà allouée
3. Retourne : `{:error, :ue_ip_address_allocation_failed}`
4. L'établissement de session échoue
5. Le SGW-C reçoit une réponse d'erreur

Prévention :

```
# Surveiller l'utilisation du pool
address_registry_count / total_pool_size > 0.8  # Alerte à 80%

# Élargir le pool avant épuisement
"internet" => [
  "100.64.1.0/24",
  "100.64.2.0/24",  # Ajouter un sous-réseau supplémentaire
  "100.64.3.0/24"
]
```

Allocation IP Statique

Cas d'Utilisation : L'appareil d'entreprise a besoin d'une IP fixe

Format de Message GTP-C :

```
Demande de Création de Session
├── IMSI : 310260123456789
├── APN : enterprise.corp
├── Allocation d'Adresse PDN (IE)
│   └── Type de PDN : IPv4
│   └── Adresse IPv4 : 10.100.0.50  ← L'UE demande une IP
spécifique
```

Traitement OmniPGW :

1. **Extraire l'IP Demandée :** Analyser l'IE d'Allocation d'Adresse PDN de la demande
2. **Valider l'IP :** Vérifier si l'IP demandée est dans le pool configuré pour cet APN
3. **Vérifier la Disponibilité :** Vérifier que l'IP n'est pas déjà allouée à une autre session
4. **Allouer ou Rejeter :**
   - Si disponible : Allouer l'IP demandée à cette session
   - Si indisponible : Rejeter la session avec le code de cause approprié

Résultats Possibles :

- **Succès :** L'UE reçoit exactement l'adresse IP qu'elle a demandée
- **Échec (IP utilisée) :** Session rejetée - IP déjà allouée
- **Échec (IP pas dans le pool) :** Session rejetée - IP pas dans la plage configurée

Délégation de Préfixe IPv6

L'UE demande IPv6 :

```
Demande de Création de Session
├── Type de PDN : IPv6
```

Le PGW-C alloue un préfixe /64 :

```
Préfixe Alloué : 2001:db8:1:a::/64

L'UE peut utiliser :
- 2001:db8:1:a::1
- 2001:db8:1:a::2
- ... (18 quintillions d'adresses)
```

Avantages :

- L'UE peut attribuer plusieurs IPs (par ex., partage de connexion)
- Prend en charge SLAAC (Configuration Automatique d'Adresse Stateless)
- Élimine la nécessité de NAT

Allocation Dual-Stack

L'UE demande IPv4v6 :

```
Demande de Création de Session
├── Type de PDN : IPv4v6
```

Le PGW-C alloue les deux :

```
IPv4 : 100.64.1.42
IPv6 : 2001:db8:1:a::/64
```

Gestion du Trafic :

- Le trafic IPv4 utilise l'adresse IPv4
- Le trafic IPv6 utilise le préfixe IPv6
- Les deux actifs simultanément
- Tunnels GTP séparés (ou tunnel dual-stack)

Lorsque l'APN n'a pas de pool IPv6 :

Si un PDN IPv4v6 est demandé mais que l'APN correspondant n'a qu'un CIDR IPv4 (pas de plage IPv6), l'allocateur IPv6 retourne `{:error, :no_ipv6_pool}`

au lieu de planter, et la procédure de session décide par type d'APN :

Type d'APN	Résultat
APN d'Urgence (SOS) (nom contient `"sos"`)	**Dégrade à IPv4 uniquement** — garde l'adresse IPv4, efface l'adresse IPv6, et définit le type de PDN négocié sur IPv4. Le PAA, PFCP et Gx sortent tous en IPv4 uniquement, et le P-CSCF est toujours retourné dans le PCO afin que l'enregistrement IMS d'urgence puisse se poursuivre (TS 29.274 / TS 23.401 porteur à adresse unique / rétrogradation de type PDN).
APN Régulier	**Rejeter** — la demande n'est pas rétrogradée silencieusement ; la mauvaise configuration (ou demande non prise en charge) est affichée.

Les demandes IPv6 uniquement contre un APN sans pool IPv6 sont également rejetées. Voir Demande IPv4v6 lorsque l'APN n'a pas de pool IPv6 dans le guide IPv6 / Dual-Stack pour le tableau de comportement complet.

Adresses IP Privées vs Publiques

Pools IP Privés (RFC 1918) :

```
# Non routable sur Internet public
subnet_map: %{
  "internet" => [
    "10.0.0.0/8",
    "172.16.0.0/12",
    "192.168.0.0/16"
  ]
}
```

Nécessite NAT au PGW-U pour accéder à Internet

Pools IP Publiques :

```
# IPs publiques routables (exemple seulement)
subnet_map: %{
  "internet" => [
    "203.0.113.0/24"  # Bloc IP public
  ]
}
```

Aucun NAT requis - routage direct vers Internet

Recommandation :

- Utiliser **IPs privées (RFC 6598)** : `100.64.0.0/10` (NAT de Niveau Opérateur)
- Réserver les IPs publiques uniquement pour des services spéciaux

---

Surveillance

Gestion du Pool IP

L'allocation et l'utilisation du pool IP sont surveillées via l'API REST OAM (HTTPS/TLS sur le port `8443`, routes sous le préfixe `/api`). La documentation interactive (Swagger UI) est disponible à `https://<omnipgw-ip>:8443/api/docs`.

```
GET /api/ip_pools            # tous les pools + utilisation
GET /api/ip_pools/<name>     # un pool par nom
```

Exemple :

```
curl -k https://<omnipgw-ip>:8443/api/ip_pools
curl -k https://<omnipgw-ip>:8443/api/ip_pools/Internet
```

Informations par Pool retournées pour chaque pool configuré :

Champ	Description
Nom	Identifiant du Pool / APN (par ex., "default", "ims.something.else", "Internet")
Total	Nombre d'IPs dans le pool
Utilisé	IPs actuellement allouées
Libre	IPs restantes disponibles pour allocation
Taux d'Utilisation %	Pourcentage du pool actuellement alloué

Cas d'Utilisation :

- Vérification rapide de la capacité avant maintenance
- Identifier les pools approchant de l'épuisement
- Vérifier la configuration du pool
- Surveiller les modèles d'allocation par APN

Métriques Clés

Nombre d'Adresses dans le Registre :

```
# IPs actuellement allouées
address_registry_count

# Utilisation du pool (nécessite un calcul)
address_registry_count / <total_pool_size> * 100
```

Exemple :

```
Pool : 100.64.1.0/24 (254 IPs)
Allouées : 150 IPs
Utilisation : 150 / 254 = 59%
```

Alertes

```
# Alerte sur une utilisation élevée du pool
- alert: UEIPPoolUtilizationHigh
  expr: address_registry_count > 200  # Pour pool /24
  for: 10m
  annotations:
    summary: "Utilisation du pool IP UE au-dessus de 80%"
    description: "Actuel : {{ $value }} / 254 IPs allouées"

# Alerte sur l'épuisement du pool
- alert: UEIPPoolExhausted
  expr: address_registry_count >= 254  # Pour pool /24
  for: 1m
  annotations:
    summary: "Pool IP UE épuisé - aucune IP disponible"

# Alerte sur les échecs d'allocation
- alert: UEIPAllocationFailures
  expr: rate(ue_ip_allocation_failures_total[5m]) > 0
  for: 5m
  annotations:
    summary: "Échecs d'allocation IP UE en cours"
```

Tableau de Bord Grafana

Panneau 1 : Utilisation du Pool IP

```
# Jauge montrant le pourcentage
(address_registry_count / 254) * 100
```

Panneau 2 : IPs Allouées au Fil du Temps

```
# Série temporelle
address_registry_count
```

Panneau 3 : Taux d'Allocation

```
# Taux de nouvelles allocations
rate(address_registry_count[5m])
```

Panneau 4 : Risque d'Épuisement du Pool

```
# Jours jusqu'à l'épuisement (basé sur le taux actuel)
(254 - address_registry_count) / rate(address_registry_count[1h])
```

Dépannage

Problème 1 : Établissement de Session Échoue (Aucune IP Disponible)

Symptômes :

- Réponse de Création de Session : Cause "Demande rejetée"
- Journal : "Échec de l'allocation de l'adresse IP UE"

Causes Possibles :

1. **Pool Épuisé**

   ```
   # Vérifier l'allocation actuelle
   curl http://<pgw_c_ip>:42069/metrics | grep
   address_registry_count
   ```

2. **Erreur de Configuration**

```
# Vérifier la configuration des sous-réseaux
config :pgw_c,
  ue: %{
    subnet_map: %{
      "internet" => [
        "100.64.1.0/24"  # Assurez-vous que le CIDR est valide
      ]
    }
  }
```

3. **Mauvaise Configuration de l'APN**

```
# Si l'APN n'est pas trouvé, revient au par défaut
# Assurez-vous que le pool par défaut existe
subnet_map: %{
  default: ["42.42.42.0/24"]
}
```

Résolution :

- **Élargir le pool :** Ajouter plus de sous-réseaux
- **Nettoyer les sessions obsolètes :** Redémarrer le PGW-C pour libérer les IPs fuyantes
- **Vérifier la config :** Vérifier `runtime.exs` pour des fautes de frappe

Problème 2 : Collision d'Adresse IP

Symptômes :

- Deux UEs reçoivent la même IP (très rare)
- Problèmes de routage

Cause :

- Bug dans le Registre d'Adresses (ne devrait pas se produire)

Débogage :

```
# Vérifier les IPs dupliquées dans les journaux
grep "already_registered" /var/log/pgw_c.log
```

Résolution :

- Devrait se corriger tout seul (la deuxième session réessaie)
- Si persistant, signaler un bug

Problème 3 : Mauvais Pool Utilisé

Symptômes :

- L'UE reçoit une IP d'un sous-réseau inattendu
- L'APN "internet" obtient une IP du pool "ims"

Cause :

- Configuration incorrecte de subnet_map

Vérifier :

```
# Vérifier la correspondance exacte des chaînes APN
subnet_map: %{
  "internet" => [...],    # Sensible à la casse
  "Internet" => [...]     # APN différent !
}
```

Résolution :

- Assurez-vous que les noms d'APN correspondent exactement (sensible à la casse)
- Utilisez le pool par défaut pour attraper tout

Problème 4 : Échec de l'Allocation IPv6

Symptômes :

- L'UE demande IPv6, reçoit une erreur
- Une UE demande un **IPv4v6** PDN contre un APN sans plage IPv6 et obtient soit une session **IPv4 uniquement** (APN SOS) soit a la session **rejetée** (APN régulier)

Causes Possibles :

1. **Aucun pool IPv6 configuré**

```
# Sous-réseaux IPv6 manquants
subnet_map: %{
  "internet" => [
    "100.64.1.0/24"  # Seulement IPv4
  ]
}
```

2. **Préfixe IPv6 invalide**

```
# Préfixe trop petit (devrait être /48 ou plus)
"internet" => [
  "2001:db8::/128"  # Mauvais - pas de place pour l'allocation
]
```

IPv4v6 sans pool IPv6 (dégradation vs. rejet) :

Lorsque une demande **IPv4v6** touche un APN sans plage IPv6, `register_new_ipv6/1` retourne `{:error, :no_ipv6_pool}` (cela ne plante **pas**). La session ensuite :

- **APN d'Urgence (SOS)** (nom contient `"sos"`): dégrade à **IPv4 uniquement** — attribue l'adresse IPv4, efface l'adresse IPv6, définit le type de PDN négocié sur IPv4, et retourne toujours le P-CSCF dans le PCO. Journalisé comme un `warning`.
- **APN Régulier** : **rejette** avec `{:error, :no_ipv6_pool}` — pas de rétrogradation silencieuse. Journalisé comme une `error`.

Voir Demande IPv4v6 lorsque l'APN n'a pas de pool IPv6.

Résolution :

```
# Ajouter un pool IPv6 (pour un APN régulier qui devrait offrir
dual-stack)
subnet_map: %{
  "internet" => [
    "100.64.1.0/24",
    "2001:db8:1::/48"  # Pool IPv6
  ]
}
```

Pour un APN d'urgence (SOS) sur un déploiement IPv4 uniquement, la dégradation IPv4 uniquement est intentionnelle — ajoutez un CIDR IPv6 uniquement si les PDNs SOS doivent être dual-stack.

Problème 5 : Utilisation Élevée du Pool

Symptômes :

- Approche de l'épuisement du pool
- `address_registry_count` approchant du maximum

Mesures Proactives :

1. **Ajouter des Sous-réseaux :**

   ```
   "internet" => [
     "100.64.1.0/24",   # Existant
     "100.64.2.0/24",   # Nouveau sous-réseau (ajoute 254 IPs)
     "100.64.3.0/24"    # Nouveau sous-réseau (ajoute 254 IPs)
   ]
   ```

2. **Utiliser des Sous-réseaux Plus Grands :**

   ```
   # Remplacer /24 par /22
   "internet" => [
     "100.64.0.0/22"  # 1022 IPs utilisables
   ]
   ```

3. **Nettoyage des Sessions :**
    - Surveiller les sessions obsolètes
    - Assurer un traitement correct de la Demande de Suppression de Session

---

Meilleures Pratiques

Planification de Capacité

Calculer la taille de pool requise :

```
Utilisateurs simultanés attendus : 10,000
Concurrence de pointe : 30% (3,000 sessions simultanées)
Tampon de croissance : 50%
IPs requises : 3,000 * 1.5 = 4,500 IPs

Sous-réseau : /20 (4,094 IPs utilisables) - Trop petit
Sous-réseau : /19 (8,190 IPs utilisables) - Suffisant
```

Sélection de Sous-réseau

Recommandé :

- Utiliser `100.64.0.0/10` (RFC 6598 - NAT de Niveau Opérateur)
- Fournit 4 millions d'IPs
- Réservé pour le NAT des fournisseurs de services

À Éviter :

- IPs publiques (coûteuses, limitées)
- Plages privées communes qui entrent en conflit avec les VPN d'entreprise

Mise en Page de Configuration

```
config :pgw_c,
  ue: %{
    subnet_map: %{
      # APN principal pour Internet - grand pool
      "internet" => [
        "100.64.0.0/18"  # 16,382 IPs
      ],

      # IMS - pool dédié plus petit
      "ims" => [
        "100.64.64.0/22"  # 1,022 IPs
      ],

      # Entreprise - pool moyen
      "enterprise.corp" => [
        "100.64.68.0/22"  # 1,022 IPs
      ],

      # IoT - grand pool pour de nombreux appareils
      "iot.m2m" => [
        "100.64.72.0/20"  # 4,094 IPs
      ],

      # Par défaut - petit fallback
      default: [
        "100.64.127.0/24"  # 254 IPs
      ]
    }
  }
```

Documentation Connexe

Configuration

- **Guide de Configuration** - Configuration du pool IP UE, mappage de sous-réseau APN

- **Configuration PCO** - DNS, P-CSCF, MTU livrés avec l'adresse IP
- **Gestion de Session** - Cycle de vie de la session, allocation IP lors de la configuration PDN
- **Interface PFCP** - Attribution d'adresse UE via PFCP au UPF

Planification Réseau

- **Interface S5/S8** - Livraison d'adresses IP via GTP-C
- **Interface Diameter Gx** - Contrôle des politiques pour l'allocation IP

Opérations

- **Guide de Surveillance** - Métriques d'utilisation du pool IP, suivi des allocations
- **Format CDR de Données** - Adresses IP UE dans les CDR pour la corrélation de facturation

---

Retour au Guide des Opérations