Guia de Operações do OmniUPF

Índice

1. Visão Geral
2. Entendendo a Arquitetura do Plano do Usuário 5G
3. Componentes do UPF
4. Integração do Protocolo PFCP e SMF
5. Operações Comuns
6. Solução de Problemas
7. Documentação Adicional
8. Glossário

Visão Geral

OmniUPF (Função de Plano do Usuário baseada em eBPF) é uma Função de Plano do Usuário 5G/LTE de alto desempenho que fornece encaminhamento de pacotes de grau de operadora, aplicação de QoS e gerenciamento de tráfego para redes móveis. Construído sobre a tecnologia eBPF do Linux (Extended Berkeley Packet Filter) e aprimorado com capacidades de gerenciamento abrangentes, o OmniUPF fornece a infraestrutura central de processamento de pacotes necessária para redes 5G SA, 5G NSA e LTE.

O que é uma Função de Plano do Usuário?

A Função de Plano do Usuário (UPF) é o elemento de rede padronizado pela 3GPP responsável pelo processamento e encaminhamento de pacotes em redes 5G e LTE. Ela fornece:

- **Encaminhamento de pacotes em alta velocidade** entre dispositivos móveis e redes de dados
- **Aplicação de Qualidade de Serviço (QoS)** para diferentes tipos de tráfego
- **Detecção e roteamento de tráfego** com base em filtros e regras de pacotes
- **Relatórios de uso** para cobrança e análises
- **Buffering de pacotes** para cenários de mobilidade e gerenciamento de sessões
- Suporte a **interceptação legal** para conformidade regulatória

OmniUPF implementa toda a funcionalidade de UPF definida na 3GPP TS 23.501 (5G) e TS 23.401 (LTE), fornecendo uma solução de plano do usuário completa e pronta para produção usando a tecnologia eBPF do kernel do Linux para desempenho máximo.

Principais Capacidades do OmniUPF

Processamento de Pacotes:

- Processamento de pacotes de plano do usuário totalmente compatível com 3GPP
- Caminho de dados baseado em eBPF para desempenho em nível de kernel
- Encapsulação e desencapsulação de GTP-U (GPRS Tunneling Protocol)
- Suporte a IPv4 e IPv6 para redes de acesso e de dados
- XDP (eXpress Data Path) para processamento de latência ultra-baixa
- Processamento de pacotes multithread

QoS e Gerenciamento de Tráfego:

- Regras de Aplicação de QoS (QER) para gerenciamento de largura de banda
- Regras de Detecção de Pacotes (PDR) para classificação de tráfego
- Regras de Ação de Encaminhamento (FAR) para decisões de roteamento
- Filtragem de Fluxo de Dados de Serviço (SDF) para roteamento específico de aplicação

- Regras de Relatório de Uso (URR) para rastreamento de volume e cobrança

Controle e Gerenciamento:

- Interface PFCP (Packet Forwarding Control Protocol) para SMF/PGW-C
- API RESTful para monitoramento e diagnósticos
- Estatísticas e métricas em tempo real
- Monitoramento de capacidade de mapas eBPF
- Painel de controle baseado na web

Características de Desempenho:

- Processamento de pacotes sem cópia via eBPF
- Encaminhamento de pacotes em nível de kernel (sem sobrecarga de espaço de usuário)
- Escalabilidade multi-core
- Capacidade de descarregar para aceleração de hardware
- Otimizado para implantações nativas em nuvem

Para detalhes sobre o uso do painel de controle, consulte Operações da Interface Web.

Entendendo a Arquitetura do Plano do Usuário

OmniUPF é uma solução unificada de plano do usuário que fornece encaminhamento de pacotes de grau de operadora para redes 5G Standalone (SA), 5G NSA e 4G LTE/EPC. **OmniUPF é um único produto** que pode funcionar simultaneamente como:

- **UPF (Função de Plano do Usuário)** - plano do usuário 5G/NSA (controlado pelo OmniSMF via N4/PFCP)
- **PGW-U (PDN Gateway User Plane)** - gateway EPC 4G para redes externas (controlado pelo OmniPGW-C via Sxc/PFCP)
- **SGW-U (Serving Gateway User Plane)** - gateway de serviço EPC 4G (controlado pelo OmniSGW-C via Sxb/PFCP)

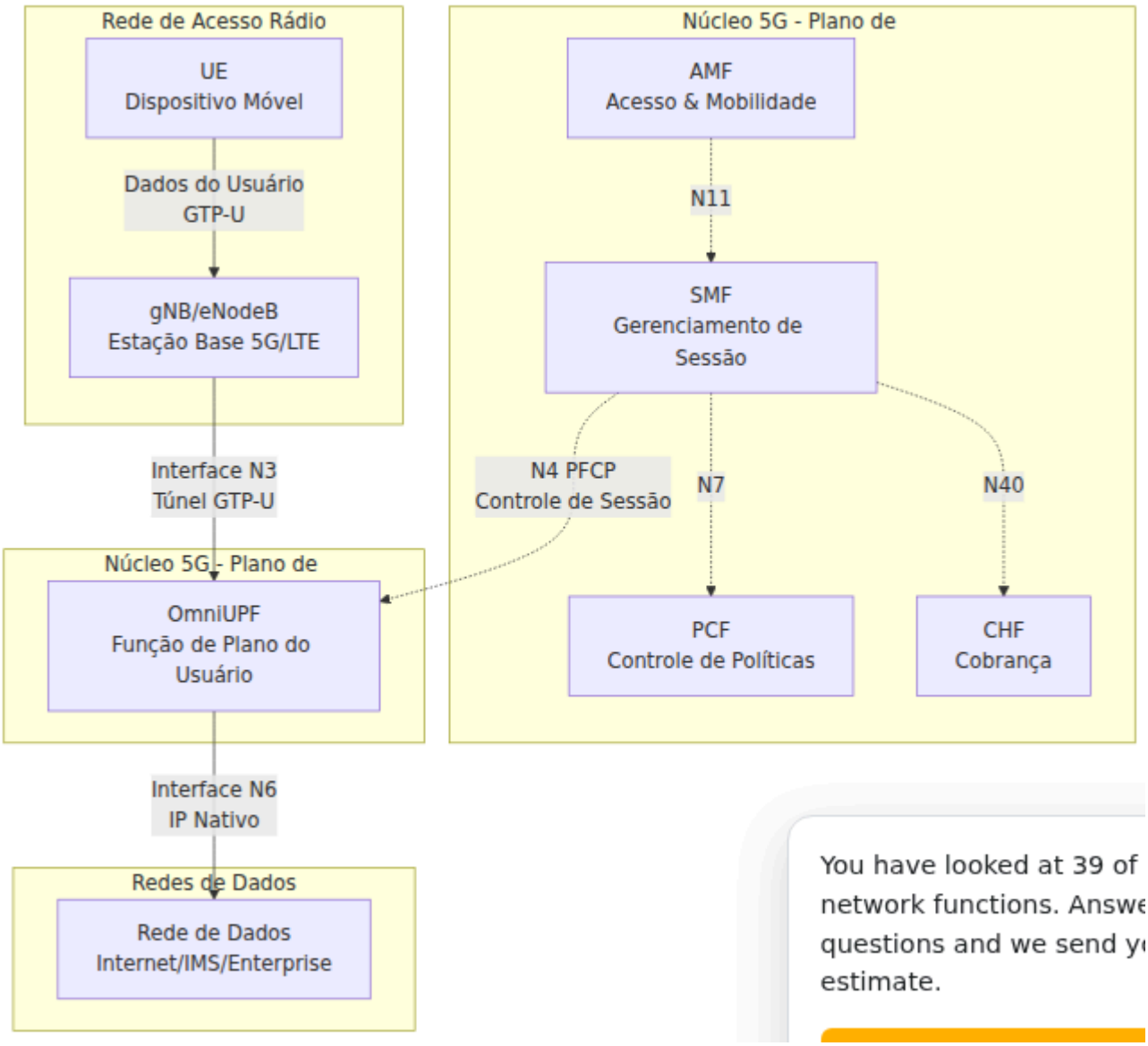
OmniUPF pode operar em **qualquer combinação** desses modos:

- **Apenas UPF**: Implantação pura de 5G
- **PGW-U + SGW-U**: Gateway 4G combinado (implantação típica de EPC)
- **UPF + PGW-U + SGW-U**: Suporte simultâneo a 4G e 5G (cenário de migração)

Todos os modos usam o mesmo mecanismo de processamento de pacotes baseado em eBPF e protocolo PFCP, proporcionando desempenho alto e consistente, seja operando como UPF, PGW-U, SGW-U ou todos os três simultaneamente.

Arquitetura da Rede 5G (Modo SA)

A solução OmniUPF está situada no plano de dados das redes 5G, fornecendo a camada de encaminhamento de pacotes em alta velocidade que conecta dispositivos móveis a redes e serviços de dados.



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

Arquitetura da Rede 4G LTE/EPC

OmniUPF também suporta implantações 4G LTE e EPC (Evolved Packet Core), funcionando como OmniPGW-U ou OmniSGW-U, dependendo da arquitetura da rede.

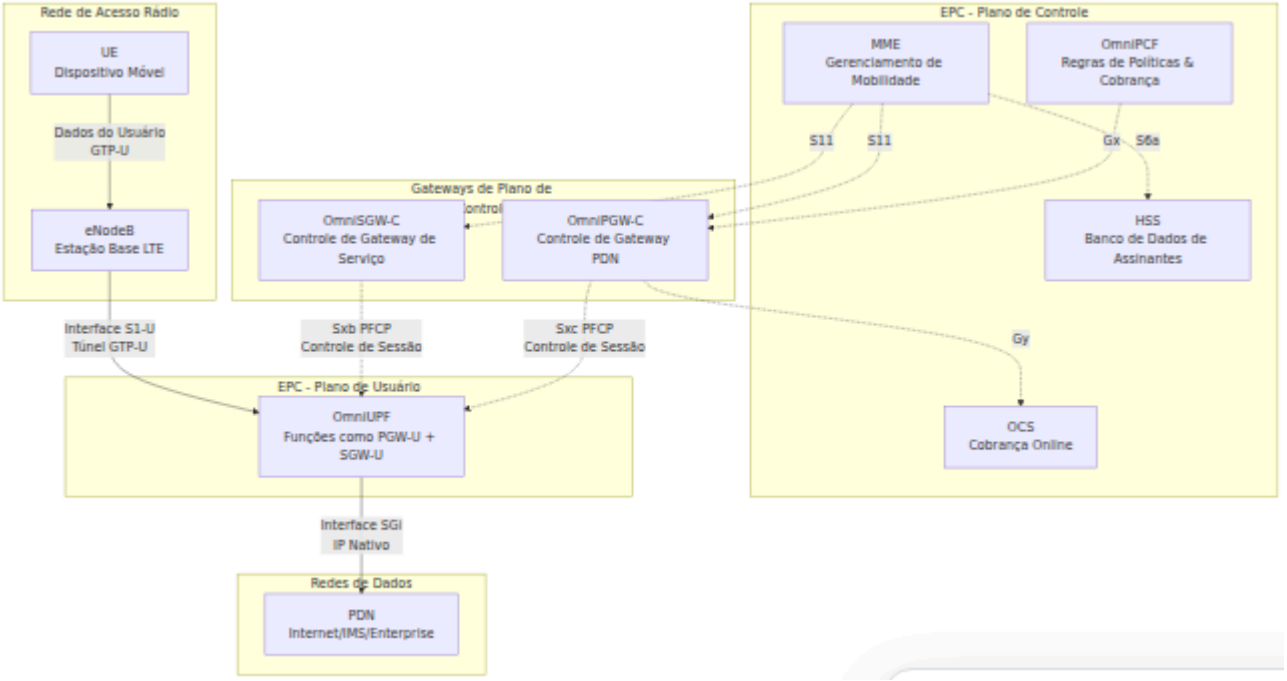
Modo Combinado PGW-U/SGW-U (Implantação Típica de 4G)

Neste modo, o OmniUPF atua como SGW-U e PGW-U, controlado por funções de plano de controle separadas.



Modo Separado SGW-U e PGW-U (Roaming/Múltiplos Locais)

Em implantações de roaming ou múltiplos locais, duas instâncias separadas do OmniUPF podem ser implantadas - uma como SGW-U e outra como PGW-U.



Como as Funções de Plano do Usuário Funcionam na Rede

A função de plano do usuário (OmniUPF, OmniPGW-U ou OmniSGW-U) opera como o plano de encaminhamento controlado pelo respectivo plano de controle:

1. **Estabelecimento de Sessão**
    - **5G**: OmniSMF estabelece associação PFCP via interface N4 com OmniUPF
    - **4G**: OmniPGW-C ou OmniSGW-C estabelece associação PFCP via Sxb/Sxc com OmniPGW-U/OmniSGW-U
    - O plano de controle cria sessões PFCP para cada sessão PDU de UE (5G) ou contexto PDP (4G)
    - O plano do usuário recebe regras PDR, FAR, QER e URR via PFCP
    - Os mapas eBPF são preenchidos com regras de encaminhamento
2. **Processamento de Pacotes de Uplink** (UE → Rede de Dados)
    - **5G**: Pacotes chegam na interface N3 do gNB com encapsulação GTP-U
    - **4G**: Pacotes chegam na interface S1-U (SGW-U) ou interface S5/S8 (PGW-U) do eNodeB com encapsulação GTP-U
    - O plano do usuário compara pacotes com PDRs de uplink com base em TEID
    - O programa eBPF aplica QER (limitação de taxa, marcação)
    - FAR determina a ação de encaminhamento (encaminhar, descartar, armazenar em buffer, duplicar)
    - Túnel GTP-U removido, pacotes encaminhados para a interface N6 (5G) ou SGi (4G)
    - URR rastreia contagens de pacotes e bytes para cobrança
3. **Processamento de Pacotes de Downlink** (Rede de Dados → UE)
    - **5G**: Pacotes chegam na interface N6 como IP nativo
    - **4G**: Pacotes chegam na interface SGi como IP nativo

- O plano do usuário compara pacotes com PDRs de downlink com base no endereço IP do UE
- Filtros SDF podem classificar ainda mais o tráfego por porta, protocolo ou aplicação
- FAR determina o túnel GTP-U e parâmetros de encaminhamento
- Encapsulação GTP-U adicionada com o TEID apropriado
- **5G**: Pacotes encaminhados para a interface N3 em direção ao gNB
- **4G**: Pacotes encaminhados para S1-U (SGW-U) ou S5/S8 (PGW-U) em direção ao eNodeB

4. **Mobilidade e Transferência**

- **5G**: OmniSMF atualiza regras PDR/FAR durante cenários de transferência
- **4G**: OmniSGW-C/OmniPGW-C atualiza regras durante transferência inter-eNodeB ou TAU (Atualização de Área de Rastreamento)
- O plano do usuário pode armazenar pacotes em buffer durante a troca de caminho
- Transição contínua entre estações base sem perda de pacotes

Integração com o Plano de Controle (4G e 5G)

OmniUPF integra-se com funções de plano de controle 5G e 4G via interfaces padrão da 3GPP:

Interfaces 5G

Interface	De → Para	Propósito	Especificação 3GPP
N4	OmniSMF ↔ OmniUPF	Estabelecimento, modificação, exclusão de sessão PFCP	TS 29.244
N3	gNB → OmniUPF	Tráfego de plano do usuário da RAN (GTP-U)	TS 29.281
N6	OmniUPF → Rede de Dados	Tráfego de plano do usuário para DN (IP nativo)	TS 23.501
N9	OmniUPF ↔ OmniUPF	Comunicação Inter-UPF para roaming/borda	TS 23.501

Interfaces 4G/EPC

Interface	De → Para	Propósito	Especificação 3GPP
Sxb	OmniSGW-C ↔ OmniUPF (modo SGW-U)	Controle de sessão PFCP para gateway de serviço	TS 29.244
Sxc	OmniPGW-C ↔ OmniUPF (modo PGW-U)	Controle de sessão PFCP para gateway PDN	TS 29.244
S1-U	eNodeB → OmniUPF (modo SGW-U)	Tráfego de plano do usuário da RAN (GTP-U)	TS 29.281
S5/S8	OmniUPF (SGW-U) ↔ OmniUPF (PGW-U)	Plano de usuário inter-gateway (GTP-U)	TS 29.281
SGi	OmniUPF (modo PGW-U) → PDN	Tráfego de plano do usuário para rede de dados (IP nativo)	TS 23.401

Nota: Todas as interfaces PFCP (N4, Sxb, Sxc) usam o mesmo protocolo PFCP definido na TS 29.244. Os nomes das interfaces diferem, mas o protocolo e os formatos de mensagem são idênticos.

Componentes do UPF

Caminho de Dados eBPF

O **caminho de dados eBPF** é o mecanismo central de processamento de pacotes que roda no kernel do Linux para desempenho máximo.

Funções Principais:

- **Processamento GTP-U**: Encapsulação e desencapsulação de túneis GTP-U
- **Classificação de Pacotes**: Comparação de pacotes com regras PDR usando TEID, IP do UE ou filtros SDF
- **Aplicação de QoS**: Aplicar limitação de taxa e marcação de pacotes por regras QER
- **Decisões de Encaminhamento**: Executar ações FAR (encaminhar, descartar, armazenar em buffer, duplicar, notificar)
- **Rastreamento de Uso**: Incrementar contadores URR para cobrança baseada em volume

Mapas eBPF: O caminho de dados usa mapas eBPF (tabelas hash na memória do kernel) para armazenamento de regras:

Nome do Mapa	Propósito	Chave	Valor
`uplink_pdr_map`	PDRs de uplink	TEID (32 bits)	Informações PDR (ID FAR, ID QER, IDs URR)
`downlink_pdr_map`	PDRs de downlink (IPv4)	Endereço IP do UE	Informações PDR
`downlink_pdr_map_ip6`	PDRs de downlink (IPv6)	Endereço IPv6 do UE	Informações PDR
`far_map`	Regras de encaminhamento	ID FAR	Parâmetros de encaminhamento (ação, informações do túnel)
`qer_map`	Regras de QoS	ID QER	Parâmetros de QoS (MBR, GBR, marcação)
`urr_map`	Rastreamento de uso	ID URR	Contadores de volume (uplink, downlink, total)
`sdf_filter_map`	Filtros SDF	ID PDR	Filtros de aplicação (portas, protocolos)

Características de Desempenho:

- **Zero-copy**: Pacotes processados inteiramente no espaço do kernel
- **Suporte a XDP**: Anexar no nível do driver de rede para latência sub-microsegundo

- **Multi-core**: Escala entre núcleos de CPU com suporte a mapas por CPU
- **Capacidade**: Milhões de PDRs/FARs em mapas eBPF (limitado pela memória do kernel)

Para monitoramento de capacidade, consulte Gerenciamento de Capacidade.

Manipulador de Interface PFCP

A **interface PFCP** implementa a 3GPP TS 29.244 para comunicação com SMF ou PGW-C.

Funções Principais:

- **Gerenciamento de Associação**: Batimento de coração PFCP e configuração/liberação de associação
- **Ciclo de Vida da Sessão**: Criar, modificar e excluir sessões PFCP
- **Instalação de Regras**: Traduzir IEs PFCP em entradas de mapa eBPF
- **Relatório de Eventos**: Notificar SMF sobre limites de uso, erros ou eventos de sessão

Suporte a Mensagens PFCP:

Tipo de Mensagem	Direção	Propósito
Configuração de Associação	SMF → UPF	Estabelecer associação de controle PFCP
Liberação de Associação	SMF → UPF	Encerrar associação PFCP
Batimento de Coração	Bidirecional	Manter associação ativa
Estabelecimento de Sessão	SMF → UPF	Criar nova sessão PDU com PDR/FAR/QER/URR
Modificação de Sessão	SMF ��� UPF	Atualizar regras para mobilidade, mudanças de QoS
Exclusão de Sessão	SMF → UPF	Remover sessão e todas as regras associadas
Relatório de Sessão	UPF → SMF	Relatar uso, erros ou eventos

Elementos de Informação (IE) Suportados:

- Criar PDR, FAR, QER, URR
- Atualizar PDR, FAR, QER, URR
- Remover PDR, FAR, QER, URR
- Informações de Detecção de Pacotes (IP do UE, F-TEID, filtro SDF)
- Parâmetros de Encaminhamento (instância de rede, criação de cabeçalho externo)
- Parâmetros de QoS (MBR, GBR, QFI)
- Gatilhos de Relatório de Uso (limite de volume, limite de tempo)

Servidor API REST

A **API REST** fornece acesso programático ao estado e operações do UPF.

Funções Principais:

- **Monitoramento de Sessões**: Consultar sessões PFCP ativas e associações
- **Inspeção de Regras**: Visualizar configurações de PDR, FAR, QER, URR
- **Estatísticas**: Recuperar contadores de pacotes, estatísticas de rotas, estatísticas de XDP
- **Gerenciamento de Buffers**: Visualizar e controlar buffers de pacotes
- **Informações de Mapas**: Monitorar uso e capacidade de mapas eBPF

Endpoints da API: (34 endpoints no total)

Categoria	Endpoints	Descrição
Saúde	`/health`	Verificação de saúde e status
Configuração	`/config`	Configuração do UPF
Sessões	`/pfcp_sessions`, `/pfcp_associations`	Dados de sessão/associação PFCP
PDRs	`/uplink_pdr_map`, `/downlink_pdr_map`, `/downlink_pdr_map_ip6`, `/uplink_pdr_map_ip6`	Regras de detecção de pacotes
FARs	`/far_map`	Regras de ação de encaminhamento
QERs	`/qer_map`	Regras de aplicação de QoS
URRs	`/urr_map`	Regras de relatório de uso
Buffers	`/buffer`	Status e controle do buffer de pacotes
Estatísticas	`/packet_stats`, `/route_stats`, `/xdp_stats`, `/n3n6_stats`	Métricas de desempenho
Capacidade	`/map_info`	Capacidade e uso de mapas eBPF
Plano de Dados	`/dataplane_config`	Endereços das interfaces N3/N9

Para detalhes e uso da API, consulte **Guia de Monitoramento**.

Painel de Controle Web

O **Painel de Controle Web** fornece um painel em tempo real para monitoramento e gerenciamento do UPF.

Recursos:

- **Visualização de Sessões**: Navegar por sessões PFCP ativas com IP do UE, TEID e contagens de regras
- **Gerenciamento de Regras**: Visualizar e gerenciar PDRs, FARs, QERs e URRs em todas as sessões
- **Monitoramento de Buffers**: Rastrear pacotes armazenados em buffer e controlar o armazenamento por FAR
- **Painel de Estatísticas**: Estatísticas em tempo real de pacotes, rotas, XDP e interfaces N3/N6
- **Monitoramento de Capacidade**: Uso de mapas eBPF com indicadores de capacidade codificados por cores
- **Visualização de Configuração**: Exibir configuração do UPF e endereços do plano de dados
- **Visualizador de Logs**: Streaming de logs ao vivo para solução de problemas

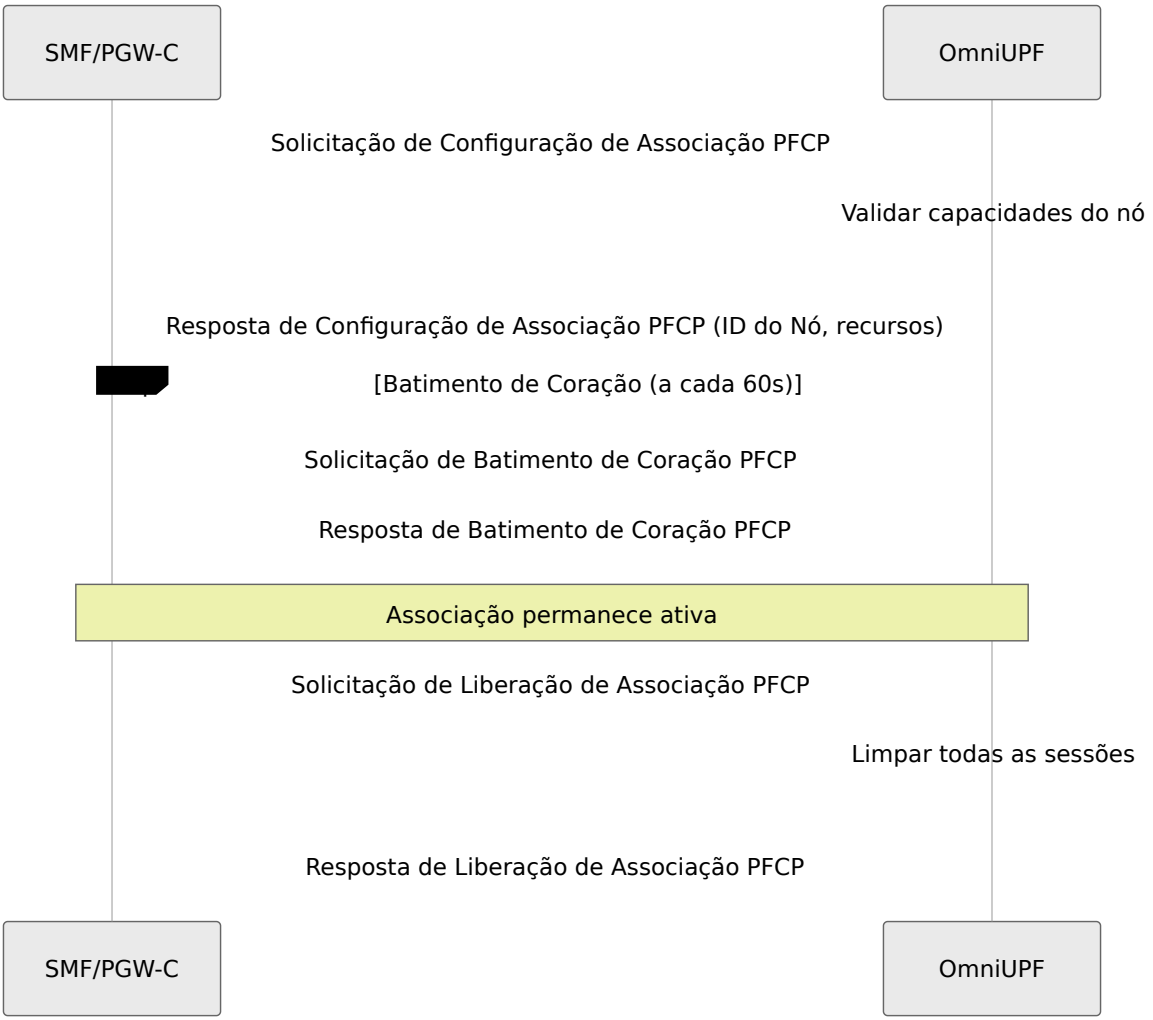
Para operações detalhadas da interface, consulte **Guia de Operações da Interface Web**.

Protocolo PFCP e Integração SMF

Associação PFCP

Antes que as sessões possam ser criadas, o SMF deve estabelecer uma associação PFCP com o UPF.

Ciclo de Vida da Associação:



Pontos Chave:

- Cada SMF estabelece uma associação com o UPF
- O UPF rastreia a associação pelo ID do Nó (FQDN ou endereço IP)
- Mensagens de batimento de coração mantêm a vivacidade da associação
- Todas as sessões sob uma associação são excluídas se a associação for liberada

Para visualizar associações, consulte Visualização de Sessões.

Detecção de Reinício do SMF e Limpeza de Sessões Órfãs

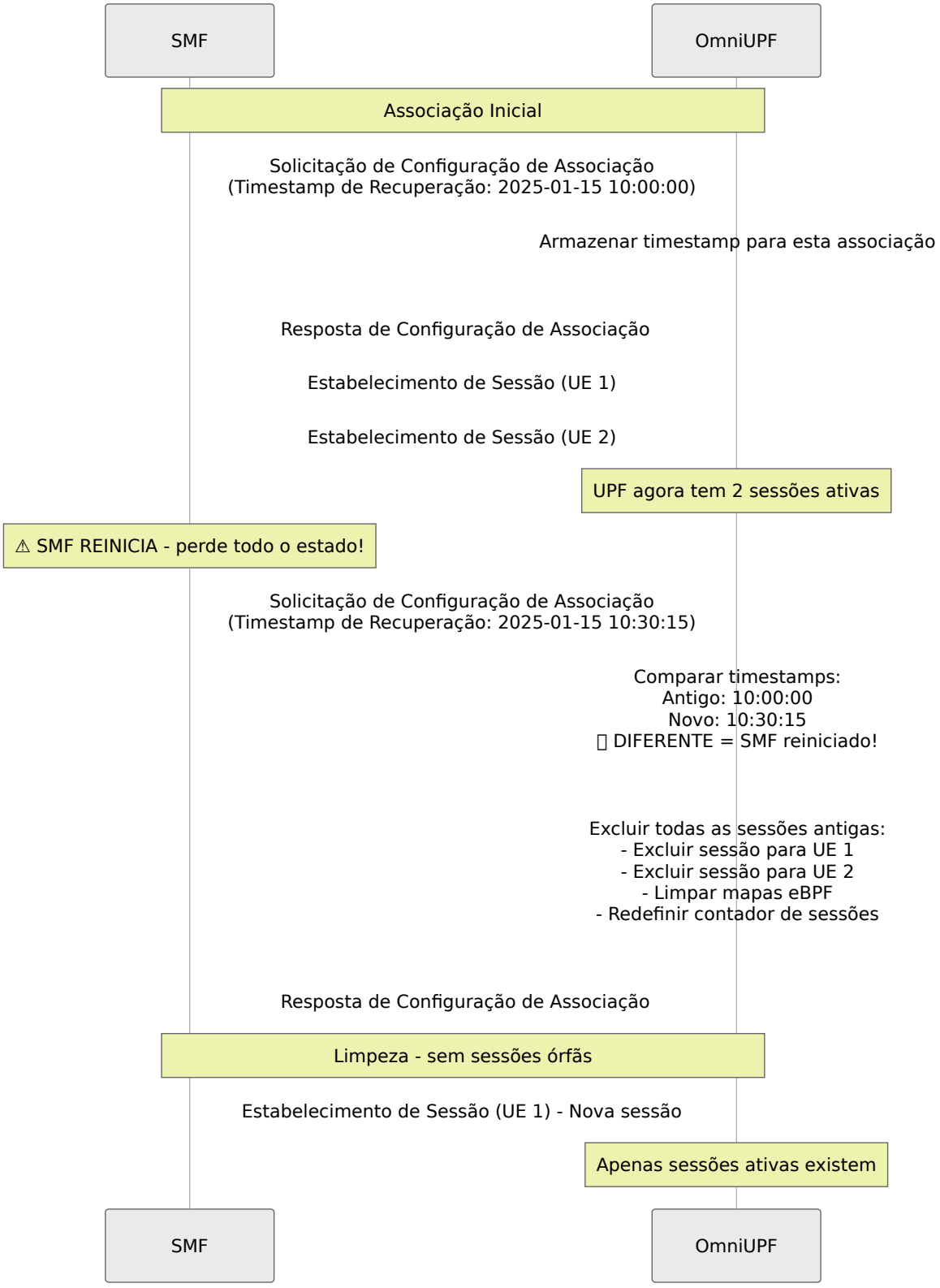
OmniUPF detecta automaticamente quando um SMF reinicia e limpa sessões órfãs de acordo com as especificações da 3GPP TS 29.244.

Como Funciona:

Quando um SMF estabelece uma associação PFCP, ele fornece um **Timestamp de Recuperação** indicando quando começou. OmniUPF armazena esse timestamp para cada associação. Se o SMF reiniciar:

1. O SMF perde todo o estado da sessão na memória
2. O SMF restabelece a associação PFCP com o UPF
3. O SMF envia um **novo Timestamp de Recuperação** (diferente do anterior)
4. O UPF detecta a mudança de timestamp = SMF reiniciado
5. O UPF exclui automaticamente **todas as sessões órfãs** da antiga instância do SMF
6. O SMF cria novas sessões para assinantes ativos

Fluxo de Detecção de Reinício:



Exemplo de Log:

Quando um SMF reinicia, você verá:

```
WARN: Associação com NodeID: smf-1 e endereço: 192.168.1.10 já
existe
WARN: Timestamp de Recuperação do SMF alterado (antigo: 2025-01-
15T10:00:00Z, novo: 2025-01-15T10:30:15Z) - SMF reiniciado,
excluindo 245 sessões órfãs
INFO: Excluindo sessão órfã 2 (LocalSEID) devido ao reinício do
SMF
INFO: Excluindo sessão órfã 3 (LocalSEID) devido ao reinício do
SMF
...
INFO: Excluindo sessão órfã 246 (LocalSEID) devido ao reinício do
SMF
```

Notas Importantes:

1. **Isolamento**: Apenas as sessões do SMF reiniciado são excluídas. Outras associações de SMF e suas sessões **não são afetadas**.
2. **Comparação de Timestamp**: Se o Timestamp de Recuperação for **idêntico**, as sessões são **mantidas** (SMF reconectado sem reiniciar).
3. **Conformidade com 3GPP**: Esse comportamento é exigido pela 3GPP TS 29.244 Seção 5.22.2:

   > "Se o Timestamp de Recuperação da função CP tiver mudado desde a última Configuração de Associação, a função UP deverá considerar que a função CP foi reiniciada e deverá excluir todas as sessões PFCP associadas a essa função CP."

Para solucionar problemas de sessões órfãs, consulte o Guia de Solução de Problemas.

Manipulação de Indicação de Erro GTP-U

OmniUPF manipula mensagens de Indicação de Erro GTP-U de pares a jusante (PGW-U, SGW-U, eNodeB, gNodeB) de acordo com as especificações da 3GPP TS 29.281.

O que são Indicações de Erro:

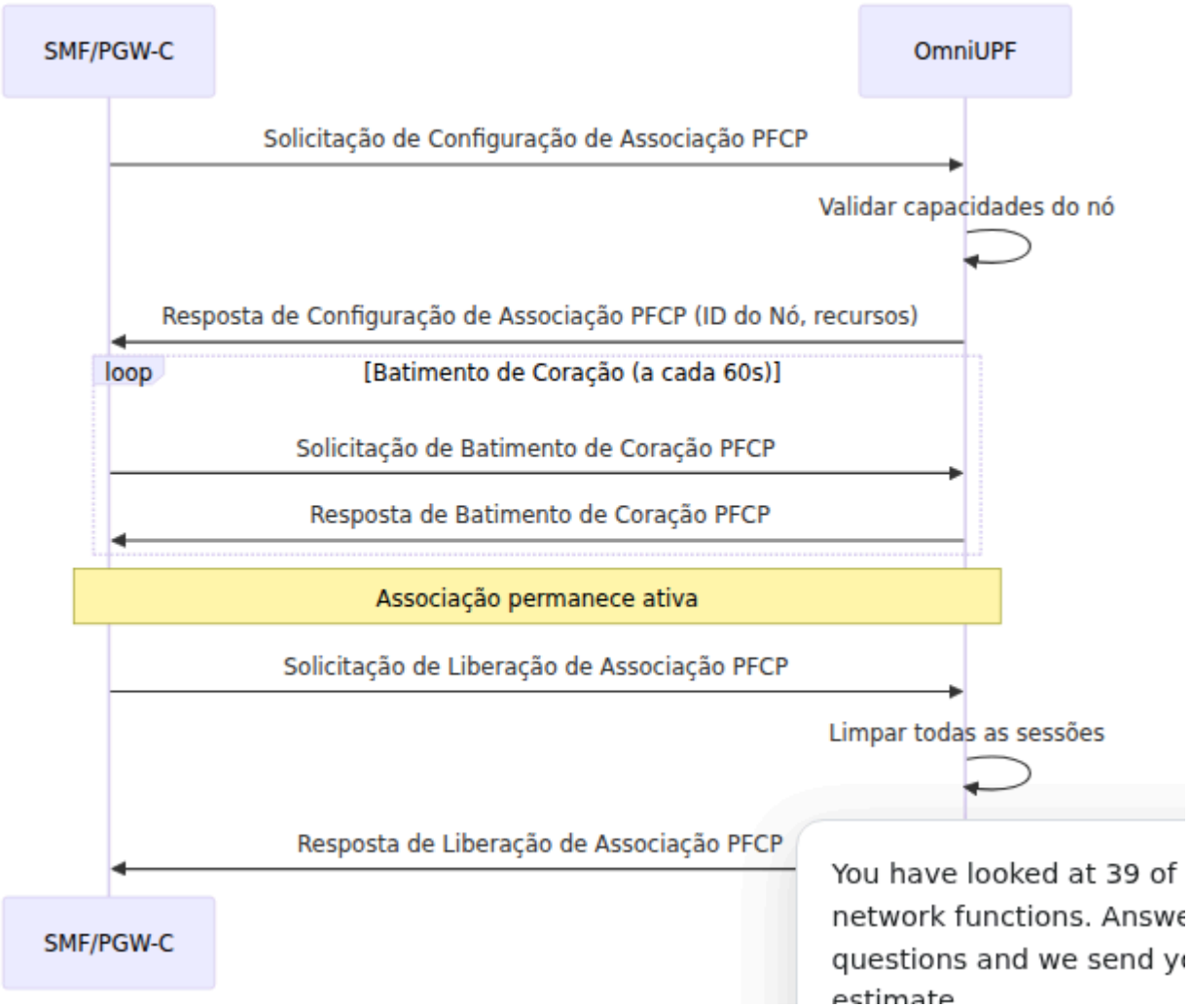
Quando o OmniUPF encaminha um pacote GTP-U para um par remoto (por exemplo, PGW-U em uma implantação SGW-U), o par pode enviar de volta uma **Indicação de Erro** se não reconhecer o TEID (Identificador de Ponto de Extremidade do Túnel). Isso indica:

- O par remoto foi reiniciado e perdeu o estado do túnel
- O túnel nunca foi criado no lado remoto (incompatibilidade de configuração)
- O túnel já foi excluído no lado remoto

Como Funciona:

1. **UPF encaminha pacote** → Envia pacote GTP-U com TEID X para par remoto (porta 2152)
2. **Par remoto não reconhece TEID X** → Procura TEID em sua tabela de túneis, não encontrado
3. **Par remoto envia Indicação de Erro** → Mensagem GTP-U tipo 26 com IE contendo TEID errôneo
4. **UPF recebe Indicação de Erro** → Analisa a mensagem para extrair TEID X
5. **UPF encontra sessões afetadas** → Pesquisa todas as sessões por FARs encaminhando para TEID X
6. **UPF exclui sessões** → Remove sessões dos mapas eBPF e do estado PFCP
7. **UPF atualiza métricas** → Incrementa contadores do Prometheus para monitoramento

Fluxo de Indicação de Erro:



Formato do Pacote (3GPP TS 29.281 Seção 7.3.1):

```
Indicação de Erro GTP-U:
┌──────────────────────────────────────────────┐
│ Cabeçalho GTP-U (12 bytes)                   │
├──────────────────────────────────────────────┤
│ Versão, PT, Flags          │ 0x32            │
│ Tipo de Mensagem           │ 26 (0x1A)       │
│ Comprimento                │ 9 bytes         │
│ TEID                       │ 0 (sempre)      │
│ Número de Sequência        │ varia           │
│ Número N-PDU               │ 0               │
│ Próximo Cabeçalho de Extensão │ 0            │
├──────────────────────────────────────────────┤
│ IE: Dados TEID I (5 bytes)                   │
├──────────────────────────────────────────────┤
│ Tipo                       │ 16 (0x10)       │
│ TEID errôneo               │ 4 bytes         │
└──────────────────────────────────────────────┘
```

Quando Isso Importa:

Cenário 1: Reinício do PGW-U na Arquitetura GTP S5/S8

- SGW-U (OmniUPF) encaminha tráfego S5/S8 para PGW-U
- PGW-U reinicia e perde todo o estado do túnel S5/S8
- SGW-U continua encaminhando para TEIDs antigos
- PGW-U envia Indicações de Erro
- SGW-U **para automaticamente de usar túneis mortos**

Cenário 2: Reinício do UPF Par na Arquitetura N9

- UPF-1 (OmniUPF) encaminha tráfego N9 para UPF-2
- UPF-2 reinicia
- UPF-1 recebe Indicações de Erro
- UPF-1 limpa sessões

Exemplo de Log:

Ao receber uma Indicação de Erro:

```
WARN: Recebida Indicação de Erro GTP-U de 192.168.50.10:2152 para
TEID 0x12345678 - par remoto não reconhece este TEID
WARN: Sessão encontrada LocalSEID=42 com FAR GlobalId=1
encaminhando para TEID errôneo 0x12345678 do par 192.168.50.10
INFO: Excluindo sessão LocalSEID=42 devido à Indicação de Erro
GTP-U para TEID 0x12345678 de 192.168.50.10
WARN: Excluídas 1 sessão(ões) devido à Indicação de Erro GTP-U
para TEID 0x12345678 do par 192.168.50.10
```

Métricas do Prometheus:

Monitore a atividade de Indicação de Erro com granularidade por par e por nó:

```
# Total de Indicações de Erro recebidas de pares
upf_buffer_listener_error_indications_received_total{node_id="pgw-u-
1",peer_address="192.168.50.10"}

# Sessões excluídas devido a Indicações de Erro
upf_buffer_listener_error_indication_sessions_deleted_total{node_id="
u-1",peer_address="192.168.50.10"}

# Indicações de Erro enviadas (para TEIDs de entrada desconhecidos)
upf_buffer_listener_error_indications_sent_total{node_id="enodeb-
1",peer_address="10.60.0.1"}
```

Rótulos de Métrica:

- `node_id`: ID do Nó PFCP da associação (ou "desconhecido" se nenhuma associação existir)
- `peer_address`: Endereço IP do par remoto

Essas métricas ajudam a identificar pares problemáticos e rastrear padrões de Indicação de Erro por nó de plano de controle.

Notas Importantes:

1. **Limpeza Automática**: Nenhuma intervenção do operador necessária - sessões são excluídas automaticamente

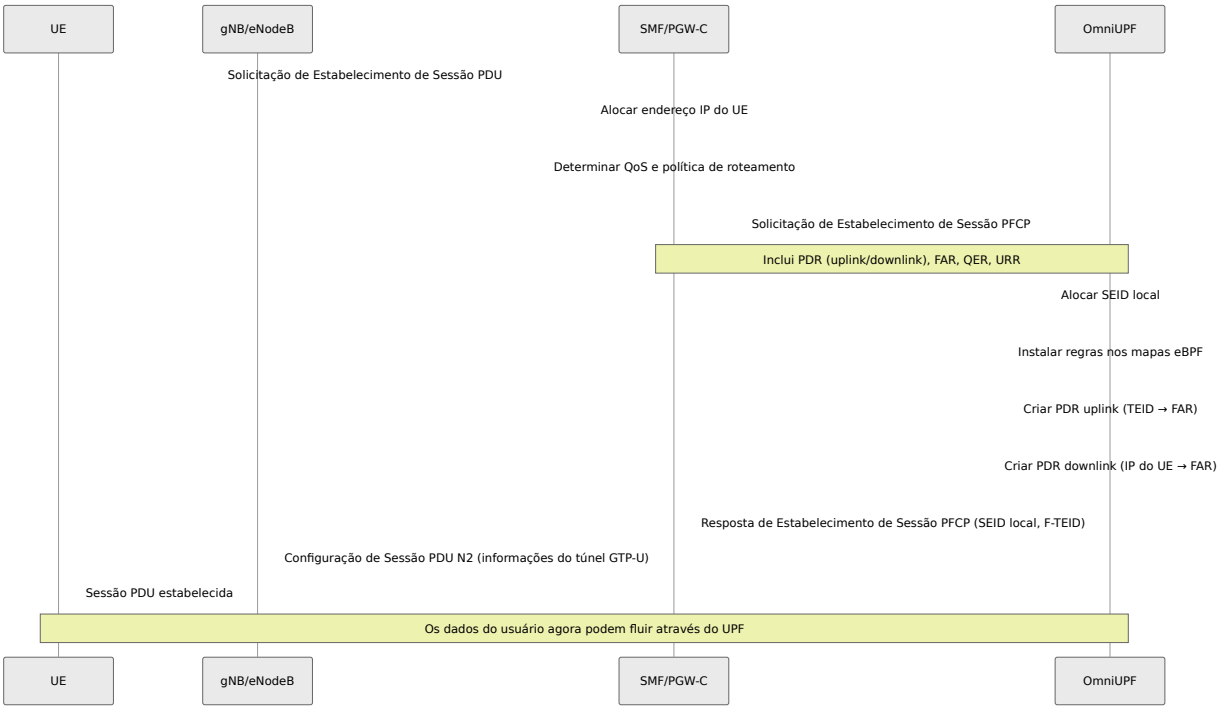
2. **Correspondência de TEID**: Apenas sessões com FARs encaminhando para o TEID errôneo exato são excluídas
3. **Isolamento por Par**: Indicações de Erro de um par afetam apenas sessões que encaminham para esse par
4. **Múltiplas Sessões**: Se várias sessões encaminham para o mesmo TEID morto, **todas são excluídas**
5. **Complementar ao Timestamp de Recuperação**:
    - Detecção de Timestamp de Recuperação = proativa (detecta reinício durante a configuração da associação)
    - Manipulação de Indicação de Erro = reativa (detecta túneis mortos quando o tráfego flui)
6. **Manipulação de Pacotes Malformados**: Indicações de Erro inválidas são registradas e ignoradas (nenhuma sessão excluída)

Para solucionar problemas de Indicações de Erro, consulte o Guia de Solução de Problemas.

Criação de Sessão PFCP

Quando um UE estabelece uma sessão PDU (5G) ou contexto PDP (LTE), o SMF cria uma sessão PFCP no UPF.

Fluxo de Estabelecimento de Sessão:



Conteúdos Típicos da Sessão:

- **PDR Uplink**: Correspondência no TEID N3, encaminhar via FAR para N6
- **PDR Downlink**: Correspondência no endereço IP do UE, encaminhar via FAR para N3 com encapsulação GTP-U
- **FAR**: Parâmetros de encaminhamento (criação de cabeçalho externo, instância de rede)
- **QER**: Limites de QoS (MBR, GBR) e marcação de pacotes (QFI)
- **URR**: Relatório de volume para cobrança (opcional)

Modificação de Sessão PFCP

O SMF pode modificar sessões para eventos de mobilidade (handover), mudanças de QoS ou atualizações de serviço.

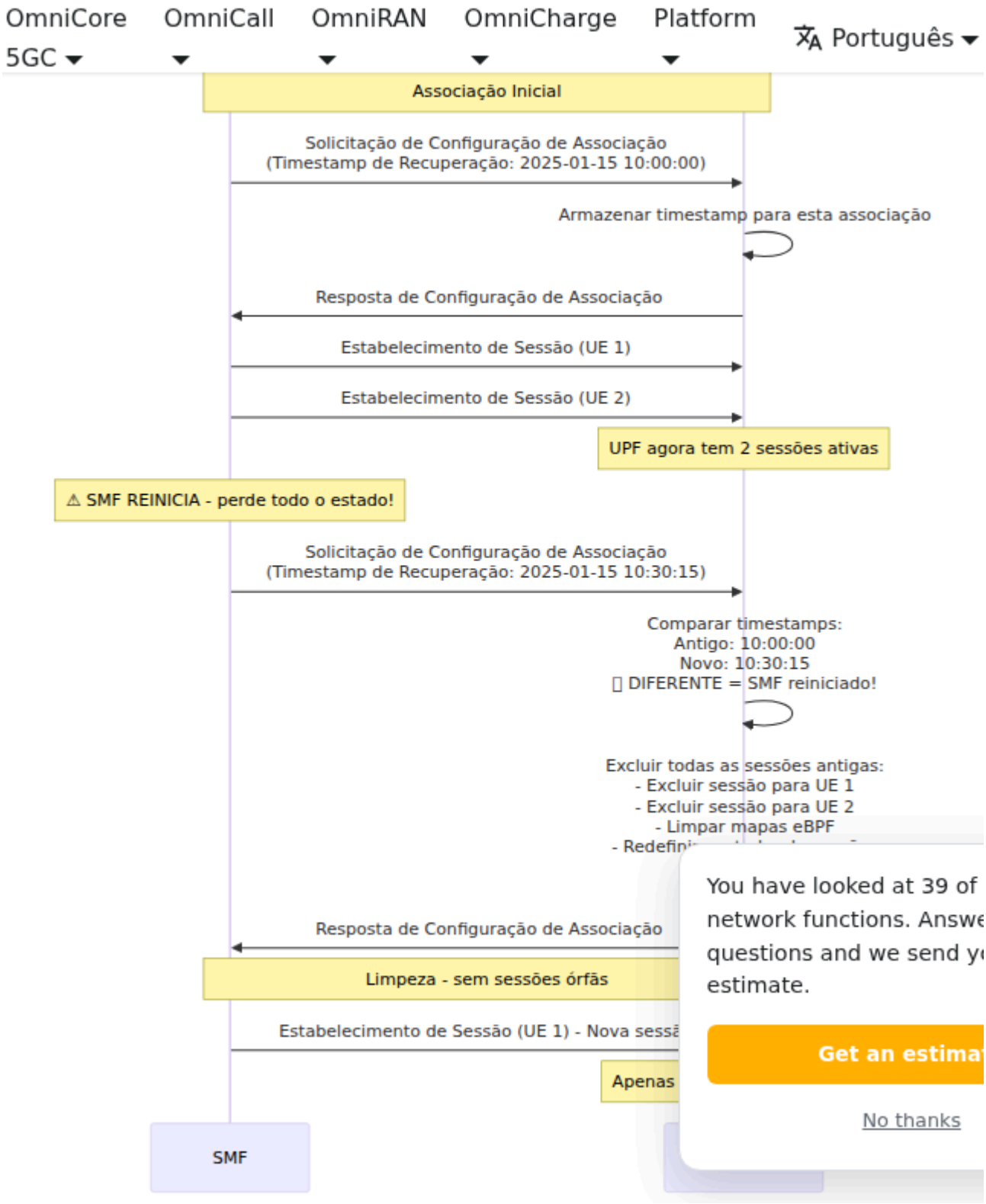
Cenários Comuns de Modificação:

1. **Handover (baseado em N2)**
    - Atualizar FAR uplink com novo ponto de extremidade de túnel gNB (F-TEID)
    - Opcionalmente, armazenar pacotes durante a troca de caminho

- Limpar o buffer para o novo caminho quando estiver pronto

2. **Mudança de QoS**
    - Atualizar QER com novos valores MBR/GBR
    - Pode adicionar/remover filtros SDF no PDR para QoS específica de aplicativo
3. **Atualização de Serviço**
    - Adicionar novos PDRs para fluxos de tráfego adicionais
    - Modificar FARs para mudanças de roteamento

Fluxo de Modificação de Sessão:

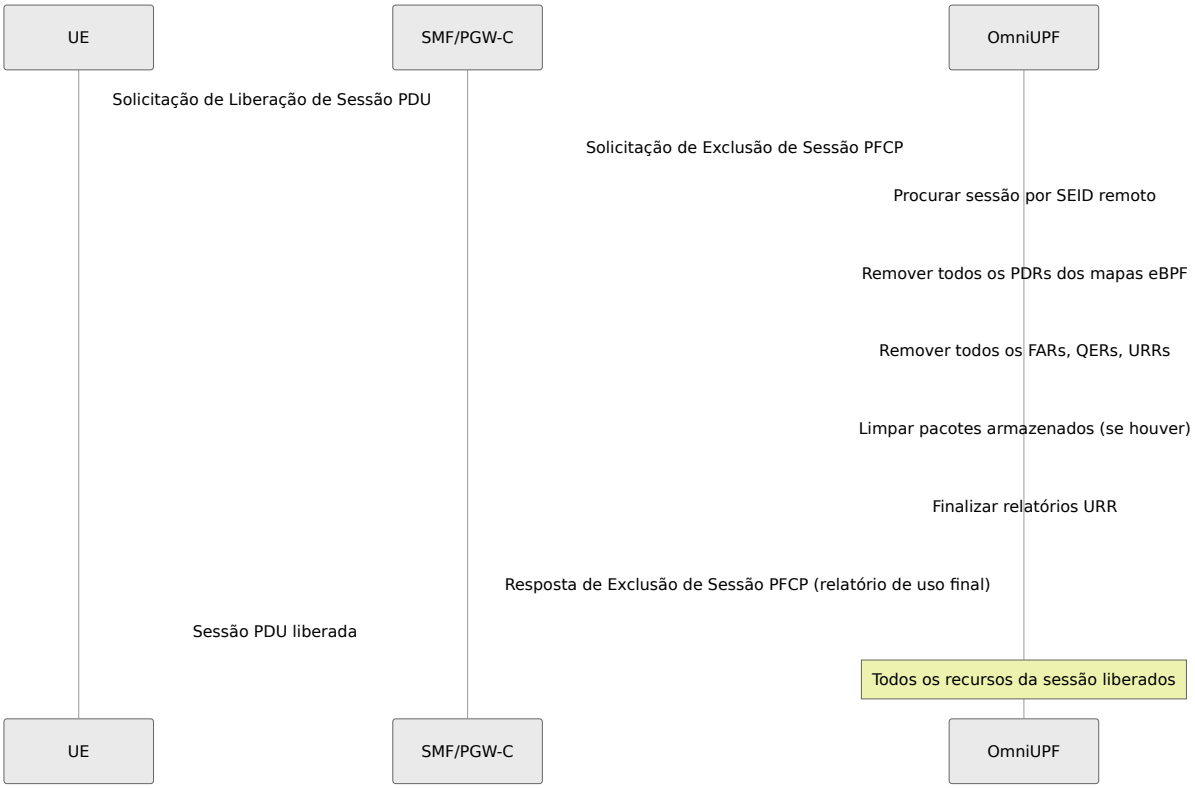


Para gerenciamento de regras, consulte o Guia de Gerenciamento de Regras.

Exclusão de Sessão PFCP

Quando uma sessão PDU é liberada, o SMF exclui a sessão PFCP no UPF.

Fluxo de Exclusão de Sessão:



Limpeza Realizada:

- Todos os PDRs removidos (uplink e downlink)
- Todos os FARs, QERs, URRs removidos
- Buffers de pacotes limpos
- Relatório de uso final enviado ao SMF para cobrança

Operações Comuns

OmniUPF fornece capacidades operacionais abrangentes através de seu painel de controle baseado na web e API REST. Esta seção cobre tarefas operacionais comuns e sua importância.

Monitoramento de Sessão

Entendendo Sessões PFCP:

As sessões PFCP representam sessões PDU ativas do UE (5G) ou contextos PDP (LTE). Cada sessão contém:

- SEIDs local e remoto (Identificadores de Ponto de Extremidade da Sessão)
- PDRs para classificação de pacotes
- FARs para decisões de encaminhamento
- QERs para aplica��ão de QoS (opcional)
- URRs para rastreamento de uso (opcional)

Principais Operações de Sessão:

- **Visualizar todas as sessões** com endereços IP do UE, TEIDs e contagens de regras
- **Filtrar sessões** por endereço IP ou TEID
- **Inspecionar detalhes da sessão** incluindo configurações completas de PDR/FAR/QER/URR
- **Monitorar contagens de sessão** por associação PFCP

Para procedimentos detalhados de sessão, consulte Visualização de Sessões.

Gerenciamento de Regras

Regras de Detecção de Pacotes (PDR):

PDRs determinam quais pacotes correspondem a fluxos de tráfego específicos. Os operadores podem:

- **Visualizar PDRs uplink** indexados por TEID da interface N3
- **Visualizar PDRs downlink** indexados por endereço IP do UE (IPv4 e IPv6)
- **Inspecionar filtros SDF** para classificação específica de aplicativo
- **Monitorar contagens de PDR** e uso de capacidade

Regras de Ação de Encaminhamento (FAR):

FARs definem o que fazer com pacotes correspondentes. Os operadores podem:

- **Visualizar ações FAR** (ENCAMINHAR, DESCARTAR, BUFFER, DUPLICAR, NOTIFICAR)

- **Inspecionar parâmetros de encaminhamento** (criação de cabeçalho externo, destino)
- **Monitorar status de buffer** por FAR
- **Alternar buffer** para FARs específicas durante a solução de problemas

Regras de Aplicação de QoS (QER):

QERs aplicam limites de largura de banda e marcação de pacotes. Os operadores podem:

- **Visualizar parâmetros de QoS** (MBR, GBR, orçamento de atraso de pacotes)
- **Monitorar QERs ativas** por sessão
- **Inspecionar marcações QFI** para fluxos de QoS 5G

Regras de Relatório de Uso (URR):

URRs rastreiam volumes de dados para cobrança. Os operadores podem:

- **Visualizar contadores de volume** (uplink, downlink, total de bytes)
- **Monitorar limites de uso** e gatilhos de relatório
- **Inspecionar URRs ativas** em todas as sessões

Para operações de regras, consulte Guia de Gerenciamento de Regras.

Buffering de Pacotes

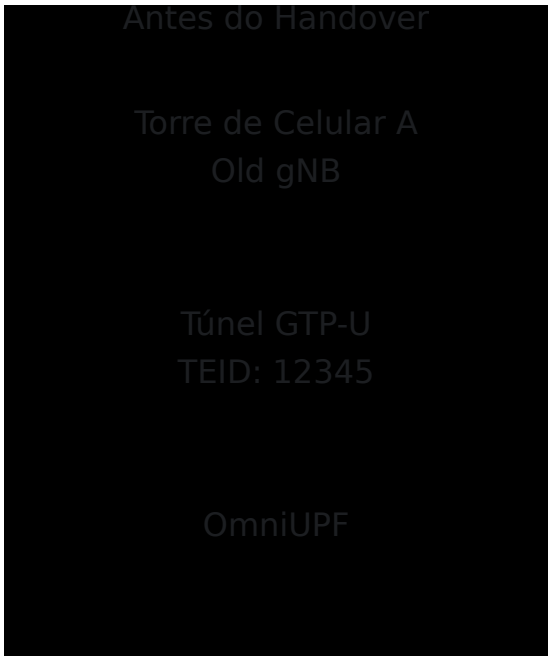
Por que o Buffering é Crítico para o UPF

O buffering de pacotes é uma das funções mais importantes de um UPF porque previne a perda de pacotes durante eventos de mobilidade e reconfigurações de sessão. Sem buffering, usuários móveis experimentariam conexões interrompidas, downloads interrompidos e falhas em comunicações em tempo real toda vez que se movem entre torres de celular ou quando as condições da rede mudam.

O Problema: Perda de Pacotes Durante a Mobilidade

Em redes móveis, os usuários estão constantemente se movendo. Quando um dispositivo se move de uma torre de celular para outra (handover), ou quando a rede precisa reconfigurar o caminho de dados, há uma janela crítica onde pacotes estão em trânsito, mas o novo caminho ainda não está pronto:


Após o Handover
OmniUPF
Túnel GTP-U
TEID: 67890
Torre de Celular B
New gNB
Durante o Handover -
Janela Crítica
Pacotes Downlink
chegando para UE
OmniUPF
Não pode enviar
Para onde enviar?
Caminho antigo
fechando
Novo caminho não está
pronto



Sem buffering: Pacotes que chegam durante esta janela crítica seriam **descartados**, causando:

- **Conexões TCP travadas** ou redefinidas (navegação na web, downloads interrompidos)
- **Chamadas de vídeo congeladas** ou descartadas (Zoom, Teams, chamadas do WhatsApp falham)
- **Sessões de jogos desconectadas** (jogos online, aplicativos em tempo real falham)
- **Chamadas VoIP com lacunas** ou que caem completamente (chamadas telefônicas interrompidas)
- **Downloads falham** e precisam ser reiniciados

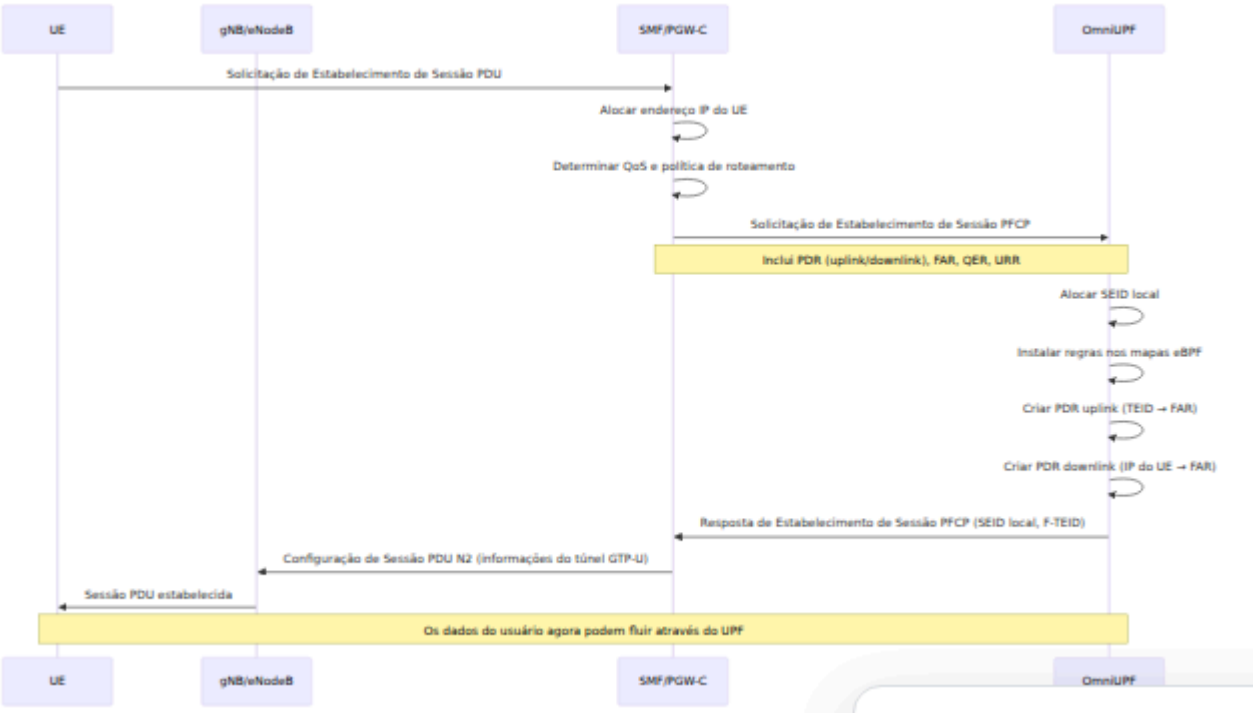
Com buffering: OmniUPF retém temporariamente pacotes até que o novo caminho seja estabelecido, depois os encaminha sem problemas. O usuário experimenta **zero interrupção**.

Quando o Buffering Acontece

OmniUPF bufferiza pacotes nessas situações críticas:

1. Handover Baseado em N2 (5G) / Handover Baseado em X2 (4G)

Quando um UE se move entre torres de celular:



Linha do tempo:

- **T+0ms**: Caminho antigo ainda ativo
- **T+10ms**: SMF diz ao UPF para bufferizar (caminho antigo fechando, novo caminho não pronto)
- **T+10-50ms**: **Janela crítica de buffering** - pacotes chegam, mas não podem ser encaminhados
- **T+50ms**: Novo caminho pronto, SMF diz ao UPF para encaminhar
- **T+50ms+**: UPF limpa pacotes bufferizados para novo caminho, depois encaminha novos pacotes normalmente

Sem buffering: ~40ms de pacotes (potencialmente milhares) seriam **perdidos**. **Com buffering**: Zero perda de pacotes, handover sem interrupções.

---

2. Modificação de Sessão (Mudança de QoS, Atualização de Caminho)

Quando a rede precisa mudar parâmetros de sessão:

- **Atualização/diminuição de QoS**: Usuário se move de cobertura 4G para 5G (modo NSA)

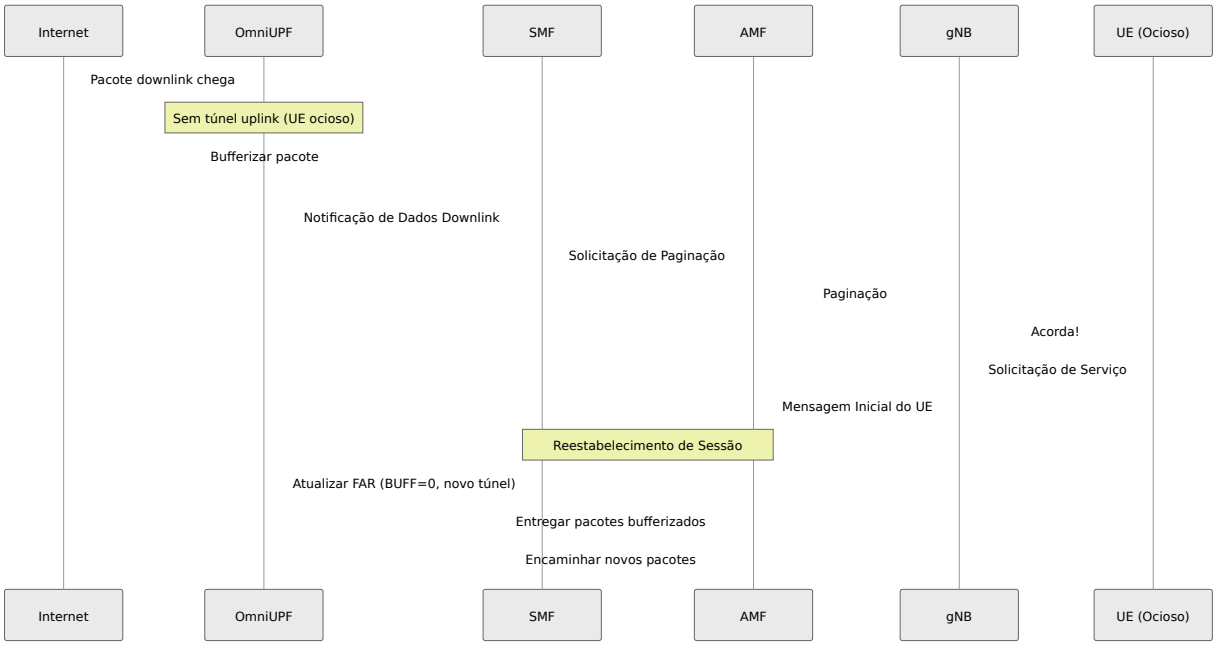
- **Mudança de política**: Usuário corporativo entra no campus corporativo (mudanças de direcionamento de tráfego)
- **Otimização de rede**: Rede central redireciona tráfego para um UPF mais próximo (atualização ULCL)

Durante a modificação, o plano de controle pode precisar atualizar várias regras atomicamente. O buffering garante que pacotes não sejam encaminhados com conjuntos de regras parciais/inconsistentes.

---

3. Notificação de Dados Downlink (Recuperação de Modo Ocioso)

Quando um UE está em modo ocioso (tela desligada, economia de bateria) e dados downlink chegam:



Sem buffering: O pacote inicial que acionou a notificação seria **perdido**, exigindo que o remetente retransmitisse (adiciona latência). **Com buffering**: O pacote que acordou o UE é entregue imediatamente quando o UE se reconecta.

---

4. Handover Inter-RAT (4G ↔ 5G)

Quando um UE se move entre cobertura 4G e 5G:

- Mudanças de arquitetura (eNodeB ↔ gNB)

- Mudanças de pontos de extremidade de túnel (alocação de TEID diferente)
- O buffering garante uma transição suave entre tipos de RAT

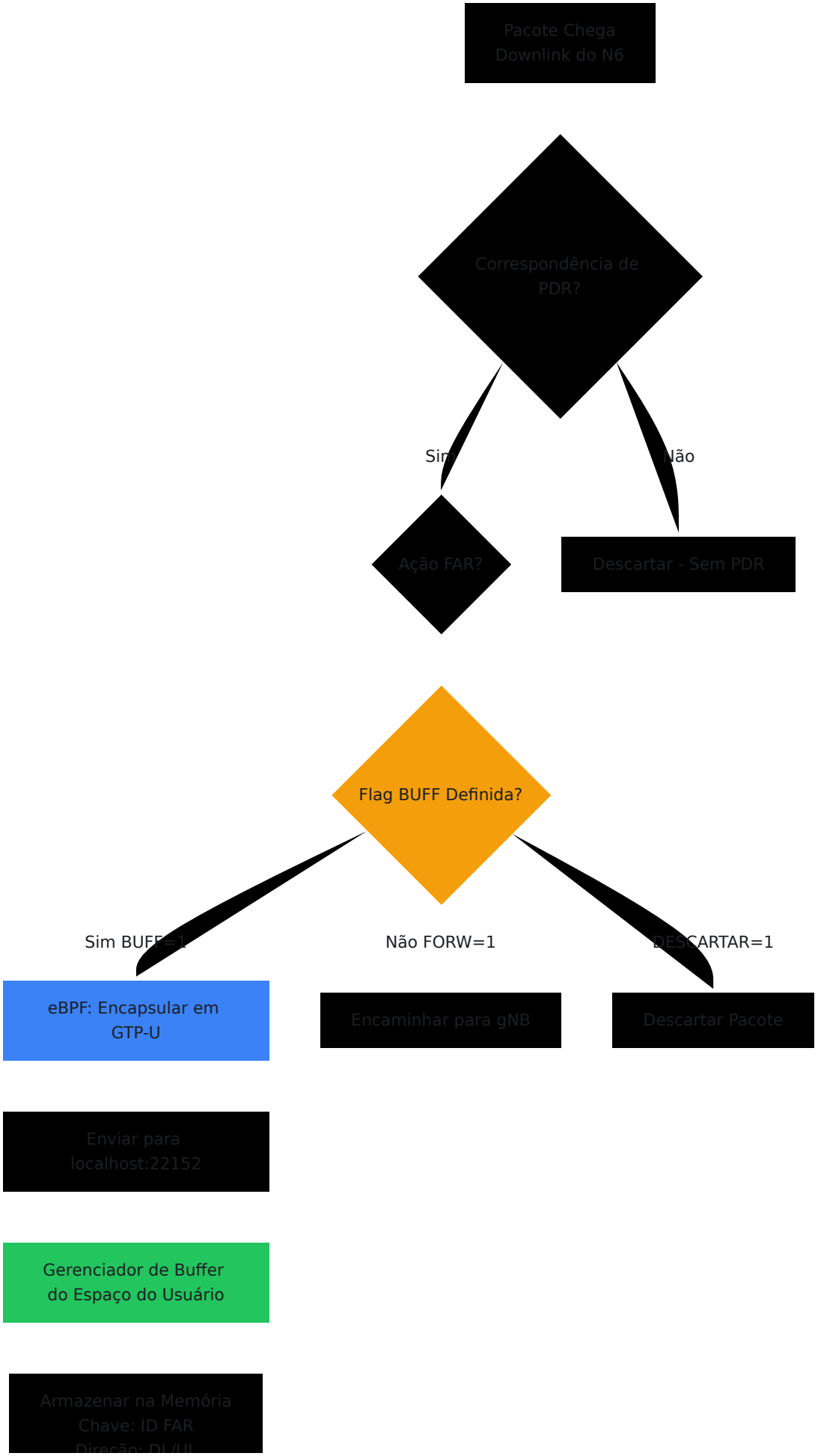
Como o Buffering Funciona no OmniUPF

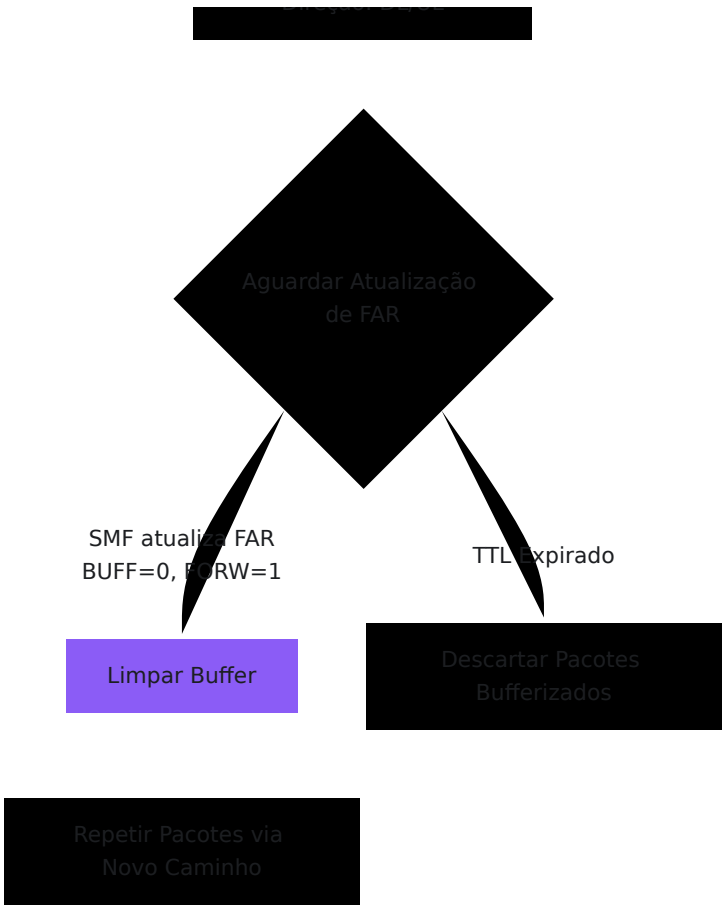
Mecanismo Técnico:

OmniUPF usa uma **arquitetura de buffering em duas etapas**:

1. **Etapa eBPF (Kernel)**: Detecta pacotes que requerem buffering com base em flags de ação FAR
2. **Etapa de Espaço do Usuário**: Armazena e gerencia pacotes bufferizados na memória

Processo de Buffering:


Pacote Chega Downlink do N6
Correspondência de PDR?
Sim
Não
Ação FAR?
Descartar - Sem PDR
Flag BUFF Definida?
Sim BUFF=1
Não FORW=1
DESCARTAR=1
eBPF: Encapsular em GTP-U
Encaminhar para gNB
Descartar Pacote
Enviar para localhost:22152
Gerenciador de Buffer do Espaço do Usuário
Armazenar na Memória
Chave: ID FAR
Direção: DL/UL



Detalhes Chave:

- **Porta de Buffer**: Porta UDP 22152 (pacotes enviados do eBPF para o espaço do usuário)
- **Encapsulação**: Pacotes embrulhados em GTP-U com ID FAR como TEID
- **Armazenamento**: Buffers na memória por FAR com metadados (timestamp, direção, tamanho do pacote)
- **Limites**:
  - Limite por FAR: 10.000 pacotes (padrão)
  - Limite global: 100.000 pacotes em todos os FARs
  - TTL: 30 segundos (padrão) - pacotes mais antigos que o TTL são descartados
- **Limpeza**: Processo em segundo plano remove pacotes expirados a cada 60 segundos

Ciclo de Vida do Buffer:

1. **Buffering Habilitado**: SMF define ação FAR BUFF=1 (bit 2) via Modificação de Sessão PFCP

2. **Pacotes Bufferizados**: eBPF detecta flag BUFF, encapsula pacotes, envia para a porta 22152
3. **Armazenamento no Espaço do Usuário**: Gerenciador de buffer armazena pacotes com ID FAR, timestamp, direção
4. **Buffering Desabilitado**: SMF define ação FAR FORW=1, BUFF=0 com novos parâmetros de encaminhamento
5. **Limpar Buffer**: Espaço do usuário repete pacotes bufferizados usando novas regras FAR (novo ponto de extremidade de túnel)
6. **Retomar Normal**: Novos pacotes encaminhados imediatamente via novo caminho

---

Por que Isso Importa para a Experiência do Usuário

Impacto no Mundo Real:

Cenário	Sem Buffering	Com Buffering
Chamada de Vídeo Durante Handover	Chamada congela por 1-2 segundos, pode cair	Sem interrupção, sem problemas
Download de Arquivo na Borda da Célula	Download falha, deve reiniciar	Download continua sem interrupções
Jogos Online Enquanto se Move	Conexão cai, expulso do jogo	Jogo suave, sem desconexões
Chamada VoIP no Carro	Chamada cai a cada handover	Cristalina, sem quedas
Streaming de Vídeo no Trem	Vídeo bufferiza, qualidade cai	Reprodução suave
Hotspot Móvel para Laptop	Sessão SSH cai, chamada de vídeo falha	Todas as conexões mantidas

Benefícios do Operador de Rede:

- **Taxa de Queda de Chamadas Reduzida (CDR)**: KPI crítico para a qualidade da rede
- **Maior Satisfação do Cliente**: Usuários não percebem as transferências
- **Custos de Suporte Mais Baixos**: Menos reclamações sobre conexões perdidas
- **Vantagem Competitiva**: Marketing de "melhor rede para cobertura"

Operações de Gerenciamento de Buffer

Os operadores podem monitorar e controlar o buffer através da interface Web e API:

Monitoramento:

- **Visualizar pacotes em buffer** por ID FAR (contagem, bytes, idade)
- **Rastrear uso do buffer** em relação aos limites (por FAR, global)
- **Alertar sobre transbordo de buffer** ou duração excessiva de buffering
- **Identificar buffers presos** (pacotes em buffer > limite de TTL)

Operações de Controle:

- **Limpar buffers**: Acionar manualmente a reprodução do buffer (solução de problemas)
- **Descartar buffers**: Descartar pacotes em buffer (limpar buffers presos)
- **Ajustar TTL**: Alterar o tempo de expiração dos pacotes
- **Modificar limites**: Aumentar a capacidade do buffer por FAR ou global

Solução de Problemas:

- **Buffer não esvaziando**: Verifique se o SMF enviou atualização FAR para desabilitar o buffering
- **Transbordo de buffer**: Aumente os limites ou investigue por que a duração do buffering é excessiva
- **Pacotes antigos no buffer**: TTL pode estar muito alto ou atualização FAR atrasada

- **Buffering excessivo**: Pode indicar problemas de mobilidade ou problemas no SMF

Para operações detalhadas de buffer, veja Guia de Gerenciamento de Buffer.

Configuração de Buffer

Configure o comportamento do buffer em `/etc/omniupf/runtime.exs`:

```
# Configurações de buffer
buffer_port = 22152                # Porta UDP para pacotes em
buffer (padrão)
```

Recomendações:

- **Redes de alta mobilidade** (estradas, trens): Aumentar `buffer_max_packets` para 20.000+
- **Áreas urbanas densas** (transferências frequentes): Diminuir `buffer_packet_ttl` para 15s
- **Aplicações de baixa latência**: Definir `buffer_packet_ttl` para 10s para evitar dados obsoletos
- **Redes IoT**: Diminuir limites (dispositivos IoT geram menos tráfego durante a transferência)

Para opções de configuração completas, veja Guia de Configuração.

Estatísticas e Monitoramento

Estatísticas de Pacotes:

Métricas de processamento de pacotes em tempo real, incluindo:

- **Pacotes RX**: Total recebido de todas as interfaces
- **Pacotes TX**: Total transmitido para todas as interfaces
- **Pacotes descartados**: Pacotes descartados devido a erros ou políticas
- **Pacotes GTP-U**: Contagem de pacotes encapsulados

Estatísticas de Rota:

Métricas de encaminhamento por rota:

- **Acertos de rota**: Pacotes correspondidos por cada rota
- **Contagens de encaminhamento**: Sucesso/falha por destino
- **Contadores de erro**: TEIDs inválidos, IPs UE desconhecidos

Estatísticas XDP:

Métricas de desempenho do eXpress Data Path:

- **XDP processados**: Pacotes tratados na camada XDP
- **XDP passados**: Pacotes enviados para a pilha de rede
- **XDP descartados**: Pacotes descartados na camada XDP
- **XDP abortados**: Erros de processamento

Estatísticas da Interface N3/N6:

Contadores de tráfego por interface:

- **N3 RX/TX**: Tráfego para/de RAN (gNB/eNodeB)
- **N6 RX/TX**: Tráfego para/de rede de dados
- **Contagens totais de pacotes**: Estatísticas agregadas da interface

Para detalhes de monitoramento, veja Guia de Monitoramento.

Gerenciamento de Capacidade

Monitoramento de Capacidade do Mapa eBPF:

O desempenho do UPF depende da capacidade do mapa eBPF. Os operadores podem:

- **Monitorar uso do mapa** com indicadores de porcentagem em tempo real
- **Visualizar limites de capacidade** para cada mapa eBPF
- **Alertas codificados por cores**:

    - Verde (<50%): Normal
    - Amarelo (50-70%): Cuidado
    - Âmbar (70-90%): Aviso
    - Vermelho (>90%): Crítico

Mapas Críticos a Monitorar:

- `uplink_pdr_map`: Classificação de tráfego de uplink
- `downlink_pdr_map`: Classificação de tráfego IPv4 de downlink
- `far_map`: Regras de encaminhamento
- `qer_map`: Regras de QoS
- `urr_map`: Rastreamento de uso

Planejamento de Capacidade:

- Cada PDR consome uma entrada de mapa (tamanho da chave + tamanho do valor)
- A capacidade do mapa é configurada na inicialização do UPF (limite de memória do kernel)
- Exceder a capacidade causa falhas na criação de sessões

Para monitoramento de capacidade, veja Gerenciamento de Capacidade.

Gerenciamento de Configuração

Configuração do UPF:

Visualize e verifique os parâmetros operacionais do UPF:

- **Interface N3**: Endereço IP para conectividade RAN (GTP-U)
- **Interface N6**: Endereço IP para conectividade com a rede de dados
- **Interface N9**: Endereço IP para comunicação entre UPFs (opcional)
- **Interface PFCP**: Endereço IP para conectividade SMF
- **Porta API**: Porta de escuta da API REST
- **Endpoint de Métricas**: Porta de métricas do Prometheus

Configuração do Dataplane:

Parâmetros ativos da camada de dados eBPF:

- **Endereço N3 ativo**: Vinculação da interface N3 em tempo de execução
- **Endereço N9 ativo**: Vinculação da interface N9 em tempo de execução (se habilitado)

Para visualização de configuração, veja Visualização de Configuração.

Solução de Problemas

Esta seção cobre problemas operacionais comuns e suas estratégias de resolução.

Falhas na Estabelecimento de Sessão

Sintomas: Sessões PFCP falham ao criar, UE não consegue estabelecer conectividade de dados

Causas Raiz Comuns:

1. **Associação PFCP Não Estabelecida**
    - Verifique se o SMF pode alcançar a interface PFCP do UPF (porta 8805)
    - Verifique o status da associação PFCP na visualização de Sessões
    - Verifique se a configuração do ID do Nó corresponde entre SMF e UPF
2. **Capacidade do Mapa eBPF Esgotada**
    - Verifique a visualização de Capacidade para uso de mapa vermelho (>90%)
    - Aumente os tamanhos do mapa eBPF na configuração do UPF
    - Exclua sessões obsoletas se o mapa estiver cheio
3. **Configuração Inválida de PDR/FAR**
    - Verifique se o endereço IP do UE é único e válido

- Verifique se a alocação de TEID não está em conflito
- Certifique-se de que o FAR referencia instâncias de rede válidas

4. **Problemas de Configuração da Interface**

   - Verifique se o IP da interface N3 é acessível a partir do gNB
   - Verifique as tabelas de roteamento para conectividade N6 com a rede de dados
   - Confirme se o tráfego GTP-U não está bloqueado pelo firewall

Para solução de problemas detalhada, veja Guia de Solução de Problemas.

Perda de Pacotes ou Problemas de Encaminhamento

Sintomas: UE tem conectividade, mas experimenta perda de pacotes ou nenhum fluxo de tráfego

Causas Raiz Comuns:

1. **Misconfiguração de PDR**

   - Verifique se o TEID do PDR de uplink corresponde ao TEID atribuído pelo gNB
   - Verifique se o IP do PDR de downlink do UE corresponde ao IP atribuído
   - Inspecione os filtros SDF para regras excessivamente restritivas

2. **Problemas de Ação FAR**

   - Verifique se a ação FAR é FORWARD (não DROP ou BUFFER)
   - Verifique os parâmetros de criação do cabeçalho externo para GTP-U
   - Certifique-se de que o endpoint de destino está correto

3. **Limites de QoS Excedidos**

   - Verifique as configurações de MBR (Taxa de Bits Máxima) do QER
   - Verifique a alocação de GBR (Taxa de Bits Garantida)
   - Monitore a perda de pacotes devido a limitação de taxa

4. **Problemas de MTU da Interface**
    - Verifique se a sobrecarga do GTP-U (40-50 bytes) não causa fragmentação
    - Verifique a configuração de MTU da interface N3/N6
    - Monitore mensagens de ICMP indicando necessidade de fragmentação

---

Problemas Relacionados ao Buffer

Sintomas: Pacotes em buffer indefinidamente, transbordo de buffer

Causas Raiz Comuns:

1. **Buffering Não Desabilitado Após Transferência**
    - Verifique a flag de buffering do FAR (bit 2)
    - Verifique se o SMF enviou Modificação de Sessão para desabilitar o buffering
    - Desabilite manualmente o buffering via painel de controle se estiver preso
2. **Expiração do TTL do Buffer**
    - Verifique a idade dos pacotes na visualização do buffer
    - Verifique a configuração do TTL do buffer (o padrão pode ser muito longo)
    - Limpe buffers expirados manualmente
3. **Capacidade do Buffer Esgotada**
    - Monitore o uso total do buffer e os limites por FAR
    - Verifique se há regras mal configuradas causando buffering excessivo
    - Ajuste os limites max_per_far e max_total do buffer

Para solução de problemas de buffer, veja Operações de Buffer.

Anomalias de Estatísticas

Sintomas: Contadores de pacotes inesperados, estatísticas ausentes

Causas Raiz Comuns:

1. **Transbordo de Contador**
    - Mapas eBPF usam contadores de 64 bits (não devem transbordar)
    - Verifique eventos de redefinição de contadores nos logs
    - Verifique se o relatório URR está funcionando
2. **Estatísticas de Rota Não Atualizando**
    - Verifique se o programa eBPF está anexado às interfaces
    - Verifique se a versão do kernel suporta os recursos eBPF necessários
    - Revise as estatísticas XDP para erros de processamento
3. **Desajuste nas Estatísticas da Interface**
    - Compare as estatísticas N3/N6 com os contadores de interface do kernel
    - Verifique se há tráfego contornando o eBPF (por exemplo, roteamento local)
    - Verifique se todo o tráfego flui através dos ganchos XDP

Degradação de Desempenho

Sintomas: Alta latência, baixa taxa de transferência, saturação da CPU

Diagnóstico:

1. **Monitore as Estatísticas XDP**: Verifique se há descartes ou abortos no XDP
2. **Verifique o Tempo de Acesso ao Mapa eBPF**: As buscas de hash devem ser sub-microsegundo
3. **Revise a Utilização da CPU**: eBPF deve se distribuir entre os núcleos

4. **Analise a Interface de Rede**: Verifique se o NIC suporta descarregamento XDP

Considerações de Escalabilidade:

- **Desempenho do XDP**: 10M+ pacotes por segundo por núcleo
- **Capacidade do PDR**: Milhões de PDRs limitados apenas pela memória do kernel
- **Contagem de Sessões**: Milhares de sessões simultâneas por instância do UPF
- **Taxa de Transferência**: Taxa de transferência de múltiplos gigabits com descarregamento adequado do NIC

Para ajuste de desempenho, veja Guia de Arquitetura.

Documentação Adicional

Guias de Operações Específicas do Componente

Para operações detalhadas e solução de problemas para cada componente do UPF:

Guia de Configuração

Referência completa de configuração, incluindo:

- Parâmetros de configuração (YAML, variáveis de ambiente, CLI)
- Modos operacionais (UPF/PGW-U/SGW-U)
- Visão geral dos modos de anexo XDP
- Compatibilidade com hipervisores (Proxmox, VMware, KVM, Hyper-V, VirtualBox)
- Compatibilidade de NIC e suporte a drivers XDP
- Exemplos de configuração para diferentes cenários
- Dimensionamento de mapas e planejamento de capacidade

Guia de Modos XDP

Configuração e otimização detalhadas do XDP, incluindo:

- Modos de anexo XDP explicados (genérico/nativo/descarregamento)
- Comparação de desempenho e benchmarks
- Configuração nativa do XDP no Proxmox VE passo a passo
- Configuração de múltiplas filas para desempenho ideal
- Configuração do XDP no VMware ESXi, KVM e Hyper-V
- Verificação e solução de problemas do XDP
- Seleção de hardware para desempenho do XDP

Guia de Arquitetura

Análise técnica profunda, incluindo:

- Fundamentos da tecnologia eBPF e ciclo de vida do programa
- Pipeline de processamento de pacotes XDP com chamadas de cauda
- Implementação do protocolo PFCP
- Arquitetura de buffering (encapsulamento GTP-U para a porta 22152)
- Limitação de taxa de controle de QoS com janela deslizante (janela de 5ms)
- Características de desempenho (latência de 3,5µs, 10 Mpps/núcleo)

Guia de Gerenciamento de Regras

Referência de regras PFCP, incluindo:

- Regras de Detecção de Pacotes (PDR) - Classificação de tráfego
- Regras de Ação de Encaminhamento (FAR) - Decisões de roteamento com flags de ação
- Regras de Aplicação de QoS (QER) - Gerenciamento de largura de banda (MBR/GBR)
- Regras de Relatório de Uso (URR) - Rastreamento e relatório de volume
- Diagramas de fluxo de pacotes de uplink e downlink
- Lógica de processamento de regras e precedência

Guia de Monitoramento

Estatísticas e gerenciamento de capacidade, incluindo:

- Estatísticas da interface N3/N6 e distribuição de tráfego
- Estatísticas de processamento XDP (passar/descarregar/redirecionar/abortar)
- Monitoramento de capacidade do mapa eBPF com zonas codificadas por cores
- Métricas de desempenho (taxa de pacotes, taxa de transferência, taxa de descarte)
- Fórmulas de planejamento de capacidade e estimativa de sessões
- Limiares de alerta e melhores práticas

Guia de Operações da Interface Web

Uso do painel de controle, incluindo:

- Visão geral do painel e navegação
- Monitoramento de sessões (estados saudáveis/não saudáveis)
- Inspeção de regras (detalhes de PDR, FAR, QER, URR)
- Monitoramento de buffer e estado de buffering de pacotes
- Painel de estatísticas em tempo real
- Visualização da capacidade do mapa eBPF
- Visualização de configuração

Documentação da API

Referência completa da API REST, incluindo:

- Documentação interativa OpenAPI/Swagger
- Paginação da API (baseada em página e baseada em offset)
- Endpoints de sessões e associações PFCP
- Regras de Detecção de Pacotes (PDR) - IPv4 e IPv6
- Regras de Ação de Encaminhamento (FAR)
- Regras de Aplicação de QoS (QER)
- Regras de Relatório de Uso (URR)

- Gerenciamento de buffer de pacotes
- Endpoints de estatísticas e monitoramento
- Gerenciamento de rotas e integração FRR
- Informações sobre mapas eBPF
- Gerenciamento de configuração
- Diretrizes de autenticação e segurança
- Fluxos de trabalho e exemplos comuns da API

Referência de Métricas

Documentação de métricas do Prometheus, incluindo:

- Métricas de mensagens PFCP (contadores, latência, erros por par)
- Métricas de ação XDP (veredictos do dataplane)
- Métricas de pacotes (contadores em nível de protocolo com rótulos de tipo de pacote)
- Métricas de sessão e associação PFCP (por nó de plano de controle)
- Métricas URR (volume de tráfego por par PFCP)
- Métricas de buffering de pacotes (estado do buffer, capacidade, taxa de transferência)
- Métricas de notificação de Relatório de Dados de Downlink (DLDR)
- Métricas de capacidade do mapa eBPF (utilização de recursos)
- Exemplos de configuração do Prometheus
- Recomendações de painel do Grafana

Referência de Códigos de Causa PFCP

Documentação de códigos de erro PFCP, incluindo:

- Definições de códigos de causa e conformidade com 3GPP (TS 129.244)
- Quando cada código de causa ocorre (sucesso, erros do cliente, erros do servidor)
- Cenários de falha comuns com resoluções
- Solução de problemas com métricas do Prometheus
- Falhas na configuração de associação e ciclo de vida da sessão
- Etapas de depuração para altas taxas de rejeição

- Recomendações de alerta para códigos de causa

Guia de Gerenciamento de Rotas do UE

Integração de roteamento FRR, incluindo:

- Visão geral e arquitetura do FRR (Free Range Routing)
- Ciclo de vida de sincronização de rotas do UE
- Sincronização automática de rotas para o daemon de roteamento
- Anúncio de rotas via OSPF e BGP
- Monitoramento de vizinhos OSPF
- Verificação do banco de dados LSA externo do OSPF
- Gerenciamento de sessão de pares BGP
- Interface de monitoramento de rotas da Web UI
- Operações manuais de sincronização de rotas
- Diagramas Mermaid para fluxo de rotas e arquitetura

Guia de IPv6 / Dual-Stack

Operação do plano do usuário IPv6 para sessões PDN IPv6 e IPv4v6:

- Pré-requisitos do kernel (habilitar IPv6 e encaminhamento)
- Configuração do pool de endereços IPv6 do UE
- Anúncio OSPFv3 de rotas de host `/128` do UE
- Comportamento do plano de dados de downlink/uplink IPv6
- Solução de problemas de encaminhamento IPv6 e anúncio de rotas

Guia de Solução de Problemas

Diagnóstico abrangente de problemas, incluindo:

- Lista de verificação de diagnóstico rápido e ferramentas
- Problemas de instalação e configuração
- Falhas de associação PFCP
- Problemas de processamento de pacotes
- Erros de XDP e eBPF
- Degradação de desempenho

- Problemas específicos de hipervisor (Proxmox, VMware, VirtualBox)
- Problemas de NIC e driver
- Procedimentos de resolução passo a passo

Documentação por Caso de Uso

Instalando e Configurando o OmniUPF

1. Comece com este guia para uma visão geral
2. Guia de Configuração para parâmetros de configuração
3. Guia da Web UI para acessar o painel de controle

Implantando no Proxmox

1. Guia de Modos XDP - Configuração Nativa do XDP no Proxmox - **Comece aqui para desempenho**
2. Guia de Configuração - Compatibilidade com Hipervisor
3. Guia de Configuração - Configuração SR-IOV do Proxmox
4. Solução de Problemas - Problemas no Proxmox

Otimizando Desempenho

1. Guia de Modos XDP - **Habilite o XDP nativo para um aumento de desempenho de 5-10x**
2. Guia de Arquitetura - Otimização de Desempenho
3. Guia de Configuração - Modos XDP
4. Guia de Monitoramento - Métricas de Desempenho
5. Solução de Problemas - Problemas de Desempenho

Compreendendo o Processamento de Pacotes

1. Guia de Arquitetura - Pipeline de Processamento de Pacotes
2. Guia de Gerenciamento de Regras
3. Guia de Monitoramento - Estatísticas

Planejando Capacidade

1. Guia de Configuração - Dimensionamento de Mapas
2. Guia de Monitoramento - Planejamento de Capacidade
3. Guia de Monitoramento - Estimativa de Capacidade de Sessão

Gerenciando Rotas do UE e Integração FRR

1. Guia de Gerenciamento de Rotas do UE - Guia completo de integração de roteamento
2. Documentação da API - Gerenciamento de Rotas - Endpoints da API de rotas
3. Guia da Web UI - Operações da página de rotas
4. Gerenciamento de Rotas do UE - Verificação FRR - Verificação do LSA OSPF

Usando a API REST

1. Documentação da API - Referência completa da API
2. Documentação da API - Swagger UI - Explorador interativo da API
3. Documentação da API - Recursos da API - Exemplos de uso da API
4. Guia da Web UI - Interface da Web como exemplo de cliente da API

Solucionando Problemas

1. Guia de Solução de Problemas - Comece aqui
2. Guia de Monitoramento - Verifique estatísticas e capacidade
3. Guia da Web UI - Use diagnósticos do painel de controle

Referência Rápida

Endpoints Comuns da API

OmniUPF fornece uma API REST para monitoramento e gerenciamento:

```
# Status e saúde
GET http://localhost:8080/api/v1/upf_status

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

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

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

# Monitoramento de capacidade
GET http://localhost:8080/api/v1/map_info

# Estatísticas de buffer
GET http://localhost:8080/api/v1/upf_buffer_info
```

Para documentação completa da API, acesse a interface Swagger em `http://<upf-ip>:8080/swagger/index.html`

Parâmetros de Configuração Essenciais

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"              # Interfaces para tráfego
N3/N6/N9
xdp_attach_mode = "native"           # "generic" | "native" |
"offload"
n3_address = "10.100.50.233"         # IP da interface N3
pfcp_address = "10.100.50.241"       # Endereço de escuta PFCP
pfcp_port = 8805                     # Porta PFCP
node_id = "10.100.50.241"            # ID do Nó PFCP

# Capacidade
max_sessions = 100_000               # Máximo de sessões
simultâneas

# API
api_port = 8080                      # Porta da API REST
```

Limiares de Monitoramento Importantes

- **Capacidade do Mapa eBPF < 70%**: Operação normal
- **Capacidade do Mapa eBPF 70-90%**: Planeje aumento de capacidade dentro de 1 semana
- **Capacidade do Mapa eBPF > 90%**: Crítico - ação imediata necessária
- **Taxa de Descarte de Pacotes < 0.1%**: Excelente
- **Taxa de Descarte de Pacotes 0.1-1%**: Boa - problemas menores
- **Taxa de Descarte de Pacotes > 5%**: Crítico - investigar imediatamente
- **XDP Abortado > 0**: Problema crítico com o programa eBPF

Referência de Normas 3GPP

OmniUPF implementa as seguintes especificações 3GPP:

Especificação	Título	Relevância
TS 23.501	Arquitetura do sistema para o Sistema 5G (5GS)	Arquitetura e interfaces do UPF 5G
TS 23.401	Melhorias do Serviço de Pacote Geral (GPRS) para acesso E-UTRAN	Arquitetura do UPF LTE (PGW-U)
TS 29.244	Interface entre os nós do Plano de Controle e do Plano do Usuário (PFCP)	Protocolo PFCP N4
TS 29.281	Protocolo de Tunelamento do Sistema de Pacote Geral (GTPv1-U)	Encapsulamento GTP-U
TS 23.503	Estrutura de controle de políticas e cobrança para o Sistema 5G (5GS)	QoS e cobrança
TS 29.212	Controle de Políticas e Cobrança (PCC)	Aplicação de QoS

Glossário

Termos da Arquitetura 5G

- **3GPP**: 3rd Generation Partnership Project - Órgão de normas para telecomunicações móveis
- **AMF**: Função de Gerenciamento de Acesso e Mobilidade - Elemento da rede central 5G para controle de acesso
- **CHF**: Função de Cobrança - Sistema de cobrança 5G
- **DN**: Rede de Dados - Rede externa (Internet, IMS, empresa)
- **eNodeB**: Evolved Node B - Estação base LTE

- **F-TEID**: Identificador de Endpoint de Tunelamento Totalmente Qualificado - ID de túnel GTP-U com endereço IP
- **gNB**: Next Generation Node B - Estação base 5G
- **GTP-U**: Protocolo de Tunelamento GPRS do Plano do Usuário - Protocolo de tunelamento para dados do usuário
- **MBR**: Taxa de Bits Máxima - Parâmetro de QoS para largura de banda máxima permitida
- **GBR**: Taxa de Bits Garantida - Parâmetro de QoS para largura de banda mínima garantida
- **N3**: Interface entre RAN e UPF (tráfego do plano do usuário)
- **N4**: Interface entre SMF e UPF (controle PFCP)
- **N6**: Interface entre UPF e Rede de Dados (tráfego do plano do usuário)
- **N9**: Interface entre dois UPFs (tráfego do plano do usuário inter-UPF)
- **PCF**: Função de Controle de Políticas - Servidor de políticas 5G
- **PDU**: Unidade de Dados de Protocolo - Sessão de dados em 5G
- **PGW-C**: PDN Gateway Control Plane - Equivalente ao plano de controle LTE do SMF
- **PGW-U**: PDN Gateway User Plane - Plano do usuário LTE (equivalente ao UPF)
- **QFI**: Identificador de Fluxo de QoS - Marcação de fluxo de QoS 5G
- **QoS**: Qualidade de Serviço - Priorização de tráfego e gerenciamento de largura de banda
- **RAN**: Rede de Acesso Rádio - Rede de estação base (gNB/eNodeB)
- **SEID**: Identificador de Endpoint de Sessão - ID de sessão PFCP
- **SMF**: Função de Gerenciamento de Sessão - Elemento da rede central 5G para controle de sessão
- **TEID**: Identificador de Endpoint de Tunelamento - ID de túnel GTP-U
- **UE**: Equipamento do Usuário - Dispositivo móvel
- **UPF**: Função do Plano do Usuário - Elemento de rede de encaminhamento de pacotes 5G

Termos do Protocolo PFCP

- **Associação**: Relação de controle entre SMF e UPF

- **FAR**: Regra de Ação de Encaminhamento - Determina o comportamento de encaminhamento de pacotes
- **IE**: Elemento de Informação - Componente da mensagem PFCP
- **Node ID**: Identificador do UPF ou SMF (FQDN ou endereço IP)
- **PDR**: Regra de Detecção de Pacotes - Classifica pacotes em fluxos
- **PFCP**: Protocolo de Controle de Encaminhamento de Pacotes - Protocolo de controle N4
- **QER**: Regra de Aplicação de QoS - Aplica limites de largura de banda e marcação
- **SDF**: Fluxo de Dados de Serviço - Filtro de tráfego específico da aplicação
- **Sessão**: Sessão PFCP representando a sessão PDU do UE ou contexto PDP
- **URR**: Regra de Relatório de Uso - Acompanha volumes de dados para cobrança

Termos do eBPF e do Kernel Linux

- **BPF**: Filtro de Pacotes Berkeley - Tecnologia de filtragem de pacotes do kernel
- **eBPF**: BPF Estendido - Caminho de dados programável do kernel
- **Hash Map**: Armazenamento de chave-valor eBPF para buscas rápidas
- **XDP**: Caminho de Dados eXpress - Processamento de pacotes do kernel em nível de driver
- **Verificador**: Componente do kernel que valida programas eBPF para segurança
- **Mapa**: Estrutura de dados eBPF compartilhada entre o kernel e o espaço do usuário
- **Zero-copy**: Processamento de pacotes sem cópia para o espaço do usuário

Termos do Produto OmniUPF

- **OmniUPF**: Função de Plano de Usuário baseada em eBPF (este produto)
- **Caminho de Dados**: Motor de processamento de pacotes (programas eBPF)
- **Plano de Controle**: Manipulador PFCP e gerenciamento de sessões
- **REST API**: API HTTP para monitoramento e gerenciamento

- **Web UI**: Painel de controle baseado em navegador

Guia de Arquitetura do OmniUPF

Índice

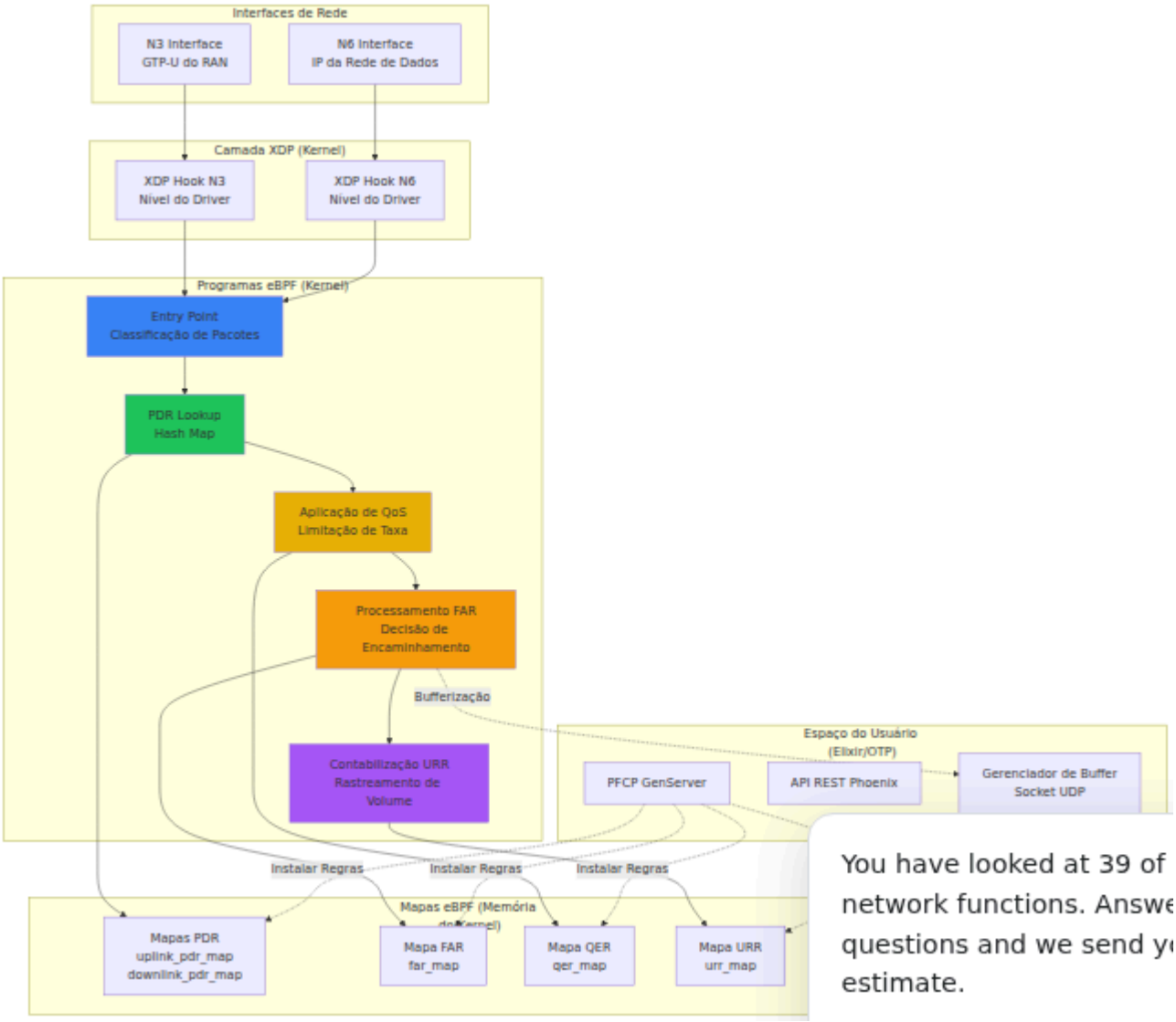
1. Visão Geral
2. Fundação da Tecnologia eBPF
3. Caminho de Dados XDP
4. Pipeline de Processamento de Pacotes
5. Arquitetura de Mapas eBPF
6. Mecanismo de Bufferização
7. Aplicação de QoS
8. Características de Desempenho
9. Escalabilidade e Ajustes

Visão Geral

OmniUPF aproveita eBPF (Extended Berkeley Packet Filter) e XDP (eXpress Data Path) para alcançar desempenho de nível de operadora para processamento de pacotes 5G/LTE. O plano de controle é implementado em Elixir/OTP usando gerenciamento de sessão PFCP baseado em GenServer, enquanto o plano de dados executa programas eBPF diretamente no kernel Linux. Essa separação elimina a sobrecarga do processamento de pacotes em espaço de usuário e alcança uma taxa de transferência de múltiplos gigabits com latência em microssegundos.

O arquivo objeto eBPF (`ipentrypoint_bpf.o`) é compilado a partir do código-fonte C (no diretório `ebpf/xdp/`) e carregado em tempo de execução pelo plano de controle Elixir via bindings NIF para libbpf.

Camadas da Arquitetura



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

Princípios de Design Chave

Processamento Zero-Copy:

- Pacotes processados inteiramente no espaço do kernel
- Sem cópia de dados entre kernel e espaço do usuário
- Manipulação direta de pacotes usando XDP

Estruturas de Dados Sem Bloqueio:

- Mapas eBPF usam tabelas hash por CPU
- Operações atômicas para acesso concorrente

- Sem sobrecarga de mutex/spinlock

Pronto para Offload de Hardware:

- O modo de offload XDP suporta execução em SmartNIC
- Compatível com placas de rede que suportam XDP
- Retorno para modos nativos do driver ou genéricos

Fundação da Tecnologia eBPF

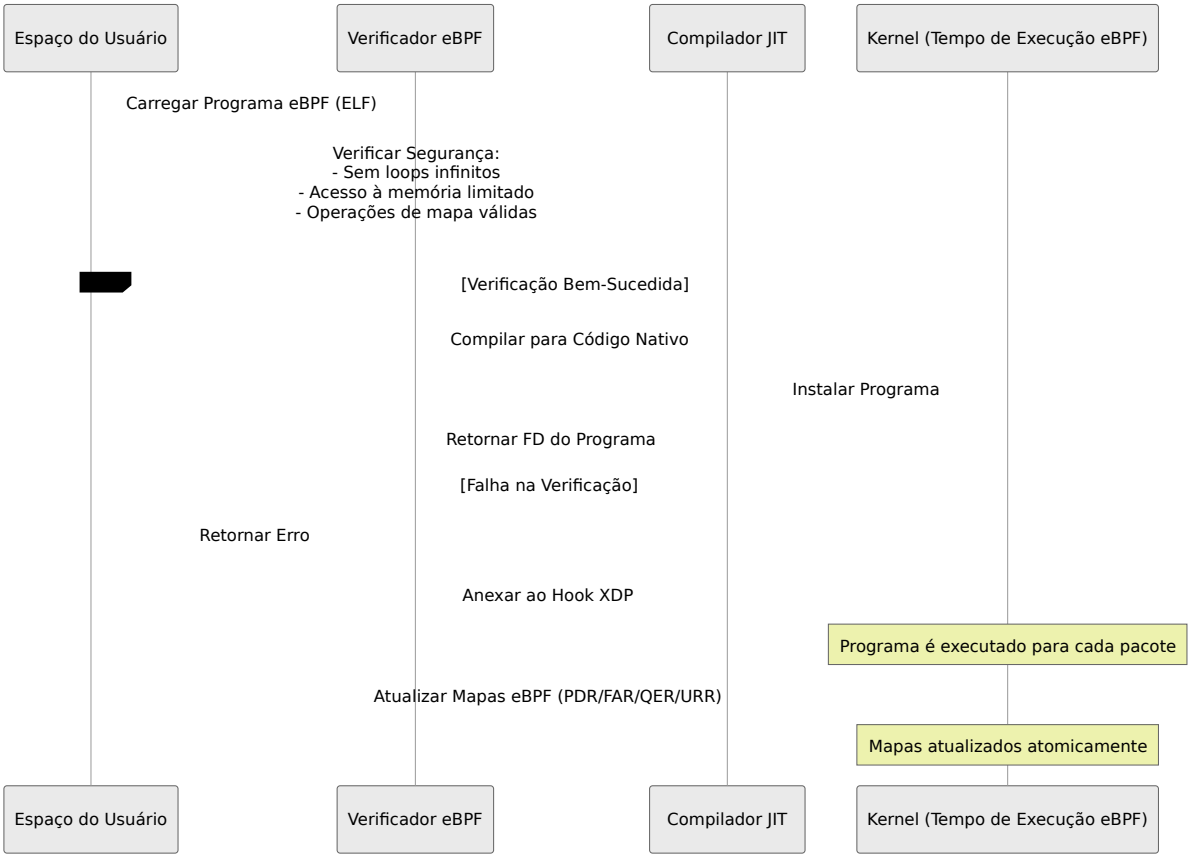
O que é eBPF?

eBPF (Extended Berkeley Packet Filter) é uma tecnologia revolucionária do kernel Linux que permite que programas seguros e isolados sejam executados no espaço do kernel sem alterar o código-fonte do kernel ou carregar módulos do kernel.

Características Principais:

- **Segurança**: O verificador eBPF garante que os programas não possam travar o kernel
- **Desempenho**: Executa na velocidade nativa do kernel (sem sobrecarga de interpretação)
- **Flexibilidade**: Pode ser atualizado em tempo de execução sem reiniciar o kernel
- **Observabilidade**: Rastreio e estatísticas integrados

Ciclo de Vida do Programa eBPF



Mapas eBPF

Os mapas eBPF são estruturas de dados do kernel compartilhadas entre programas eBPF e espaço do usuário.

Tipos de Mapa Usados no OmniUPF:

Tipo de Mapa	Descrição	Caso de Uso
`BPF_MAP_TYPE_HASH`	Tabela hash com pares chave-valor	Busca PDR por TEID ou IP do UE
`BPF_MAP_TYPE_ARRAY`	Array indexado por inteiro	Busca QER, FAR, URR por ID
`BPF_MAP_TYPE_PERCPU_HASH`	Tabela hash por CPU (sem bloqueio)	Buscas PDR de alto desempenho
`BPF_MAP_TYPE_LRU_HASH`	Hash LRU (Least Recently Used)	Evicção automática de entradas antigas

Operações de Mapa:

- **Lookup**: O(1) busca hash (sub-microsegundo)
- **Update**: Atualizações atômicas do espaço do usuário
- **Delete**: Remoção imediata de entradas
- **Iterate**: Operações em lote para despejos de mapa

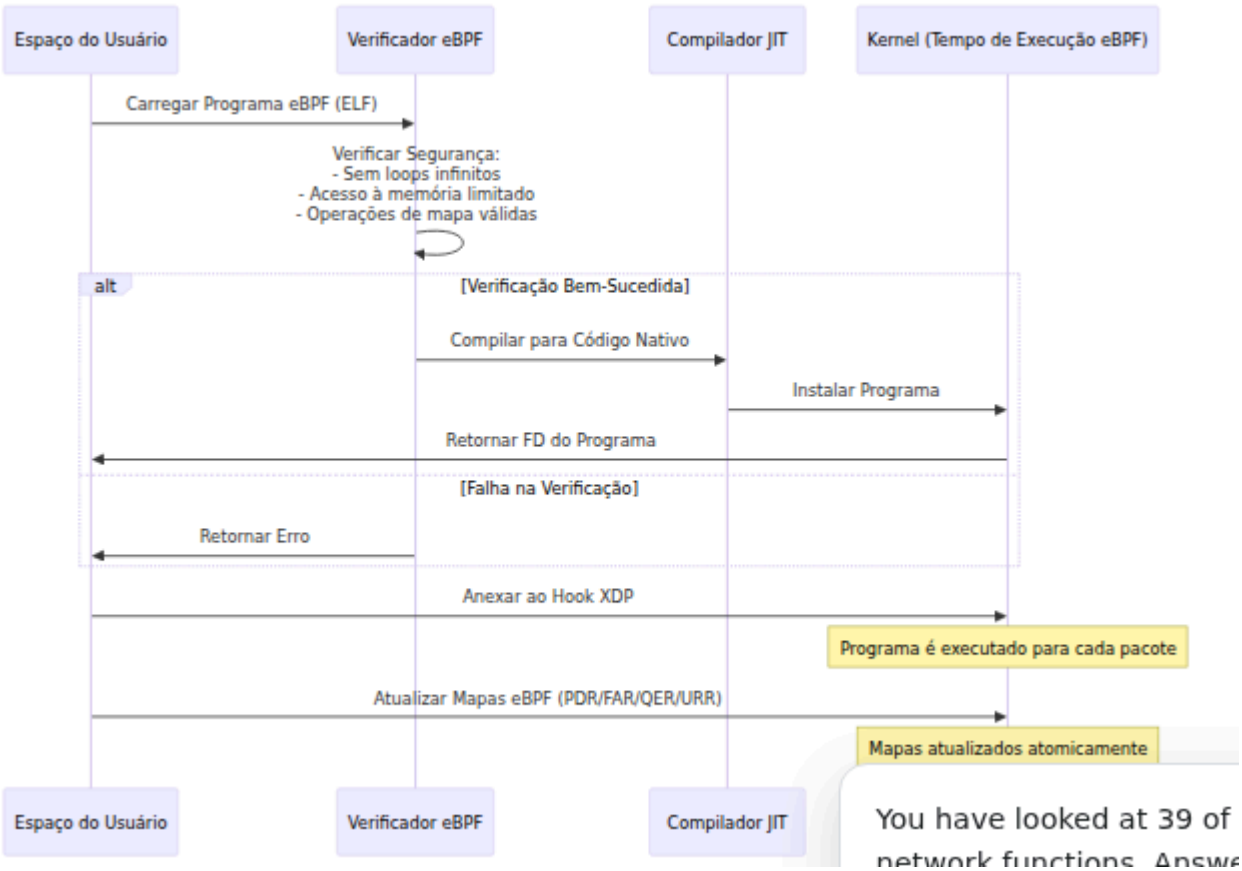
Caminho de Dados XDP

Visão Geral do XDP

XDP (eXpress Data Path) é um hook do kernel Linux que permite que programas eBPF processem pacotes no ponto mais cedo possível - logo após o driver de rede recebê-los, antes da pilha de rede do kernel.

Modos de Anexação XDP

OmniUPF suporta três modos de anexação XDP, cada um com diferentes características de desempenho e compatibilidade.



1. Modo de Offload XDP

Execução em Hardware (Melhor Desempenho):

- Programa eBPF é executado diretamente no hardware SmartNIC
- Processamento de pacotes no NIC sem tocar na CPU
- Alcança taxa de transferência de 100 Gbps+
- Requer SmartNIC compatível (Netronome, Mellanox ConnectX-6)

Configuração (em `runtime.exs`):

```
xdp_attach_mode = "offload"
```

Limitações:

- Requer hardware SmartNIC caro
- Complexidade limitada do programa eBPF
- Nem todos os recursos eBPF são suportados em hardware

---

2. Modo Nativo XDP (Padrão para Produção)

Execução em Nível de Driver (Alto Desempenho):

- Programa eBPF é executado no contexto do driver de rede
- Pacotes processados antes da alocação de SKB (socket buffer)
- Alcança 10-40 Gbps por núcleo
- Requer driver com suporte a XDP (a maioria dos drivers modernos)

Configuração (em `runtime.exs`):

```
xdp_attach_mode = "native"
```

Vantagens:

- Desempenho muito alto (milhões de pps)
- Ampla compatibilidade de hardware
- Conjunto completo de recursos eBPF

Drivers Suportados:

- Intel: i40e, ice, ixgbe, igb
- Mellanox: mlx4, mlx5
- Broadcom: bnxt
- Amazon: ena
- A maioria das placas de rede 10G+

---

3. Modo Genérico XDP

Emulação de Software (Compatibilidade):

- Programa eBPF é executado após o kernel alocar SKB
- Emulação de software do comportamento XDP
- Funciona em qualquer interface de rede
- Útil para testes e desenvolvimento

Configuração (em `runtime.exs`):

```
xdp_attach_mode = "generic"
```

Casos de Uso:

- Desenvolvimento e testes
- Ambientes virtualizados (VMs sem SR-IOV)
- Hardware de rede mais antigo
- Testes de interface de loopback

Desempenho: 1-5 Gbps (significativamente mais lento que nativo/offload)

Códigos de Retorno XDP

Programas eBPF retornam códigos de ação XDP para informar ao kernel o que fazer com os pacotes:

Código de Retorno	Significado	Uso no OmniUPF
`XDP_PASS`	Enviar pacote para a pilha de rede do kernel	Bufferização (entrega local), ICMP, tráfego desconhecido
`XDP_DROP`	Descartar pacote imediatamente	Pacotes inválidos, limitação de taxa, descartes de política
`XDP_TX`	Transmitir pacote de volta pela mesma interface	Não utilizado atualmente
`XDP_REDIRECT`	Enviar pacote para uma interface diferente	Caminho principal de encaminhamento (N3 ↔ N6)
`XDP_ABORTED`	Erro de processamento, descartar pacote e registrar	Erros de programa eBPF

Pipeline de Processamento de Pacotes

Estrutura do Programa

OmniUPF usa chamadas de cauda eBPF para criar um pipeline modular de processamento de pacotes.



Chamadas de Cauda:

- Permitem que programas eBPF chamem outros programas eBPF
- Reutiliza o mesmo quadro de pilha (profundidade de pilha limitada)

- Habilita o design modular do pipeline
- Profundidade máxima de 33 chamadas de cauda

Processamento de Pacotes de Uplink

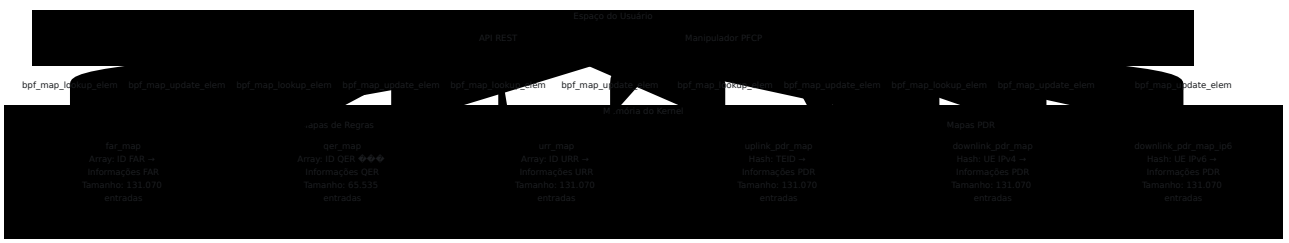


Processamento de Pacotes de Downlink



Arquitetura de Mapas eBPF

Layout da Memória do Mapa



Dimensionamento do Mapa

OmniUPF calcula automaticamente os tamanhos dos mapas com base na configuração `max_sessions`:

```
Mapas PDR = 2 × max_sessions  (uplink + downlink)
Mapas FAR = 2 × max_sessions  (uplink + downlink)
Mapas QER = 1 × max_sessions  (compartilhado por sessão)
Mapas URR = 3 × max_sessions  (múltiplos URRs por sessão)
```

Exemplo (max_sessions = 65.535):

- Mapas PDR: 131.070 entradas cada
- Mapa FAR: 131.070 entradas
- Mapa QER: 65.535 entradas
- Mapa URR: 131.070 entradas

Memória Total:

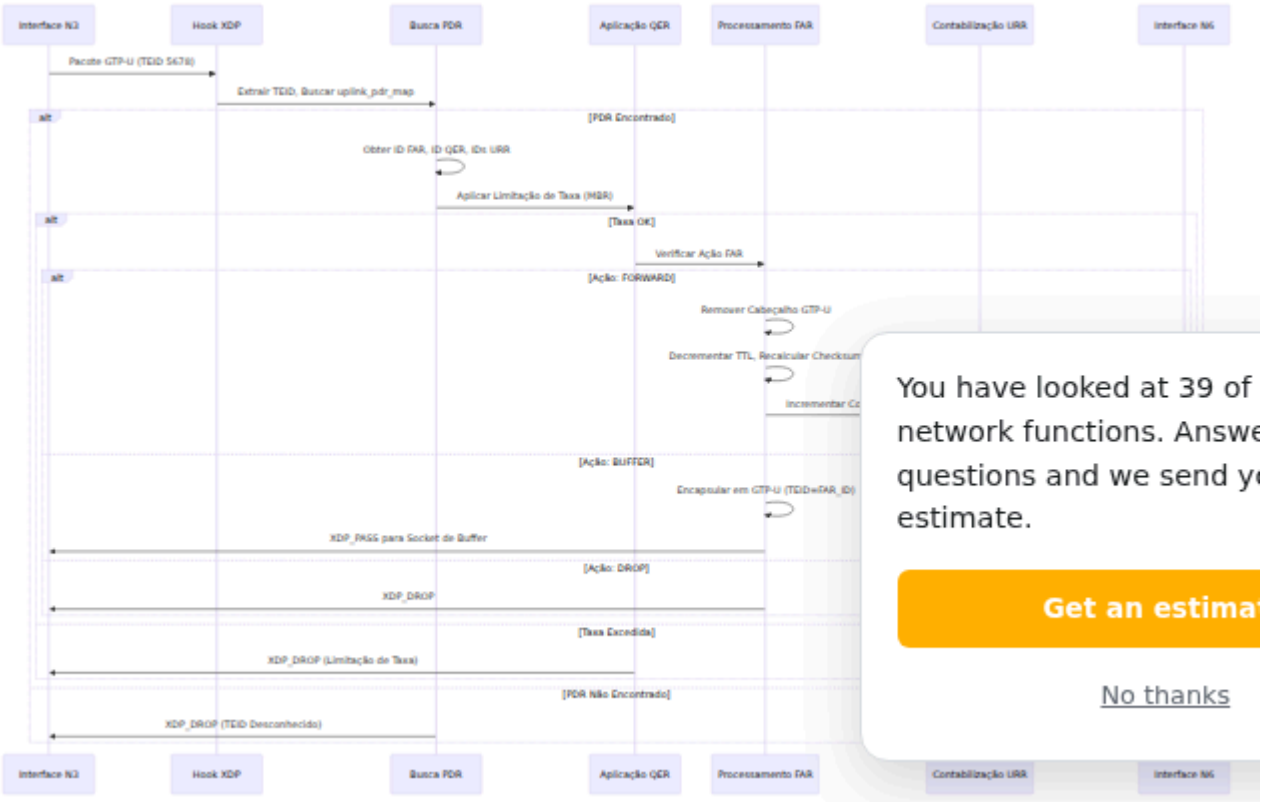
```
Mapas PDR: 3 × 131.070 × 212 B = ~83 MB
Mapa FAR:  131.070 × 20 B = ~2.6 MB
Mapa QER:  65.535 × 36 B = ~2.3 MB
Mapa URR:  131.070 × 20 B = ~2.6 MB
Total: ~91 MB de memória do kernel
```

Mecanismo de Bufferização

Visão Geral da Bufferização

OmniUPF implementa bufferização de pacotes para cenários de transferência, encapsulando pacotes em GTP-U e enviando-os para um processo de espaço de usuário via socket UDP.

Arquitetura de Bufferização



Detalhes da Encapsulação do Buffer

Quando a bufferização está habilitada (bit de ação FAR 2 definido), o programa eBPF:

1. **Calcula o Tamanho do Pacote Original**:

```
orig_packet_len = ntohs(ip->tot_len);  // Do cabeçalho IP
```

2. **Expande o Cabeçalho do Pacote**:

```
// Adicionar espaço para: IP Externo + UDP + GTP-U
gtp_encap_size = sizeof(struct iphdr) + sizeof(struct udphdr) +
sizeof(struct gtpuhdr);
bpf_xdp_adjust_head(ctx, -gtp_encap_size);
```

3. **Constrói o Cabeçalho IP Externo**:

```
ip->saddr = original_sender_ip;  // Preservar origem para
evitar filtragem martiana
ip->daddr = local_upf_ip;        // IP local onde o listener de
espaço de usuário se vincula
ip->protocol = IPPROTO_UDP;
ip->ttl = 64;
```

4. **Constrói o Cabeçalho UDP**:

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

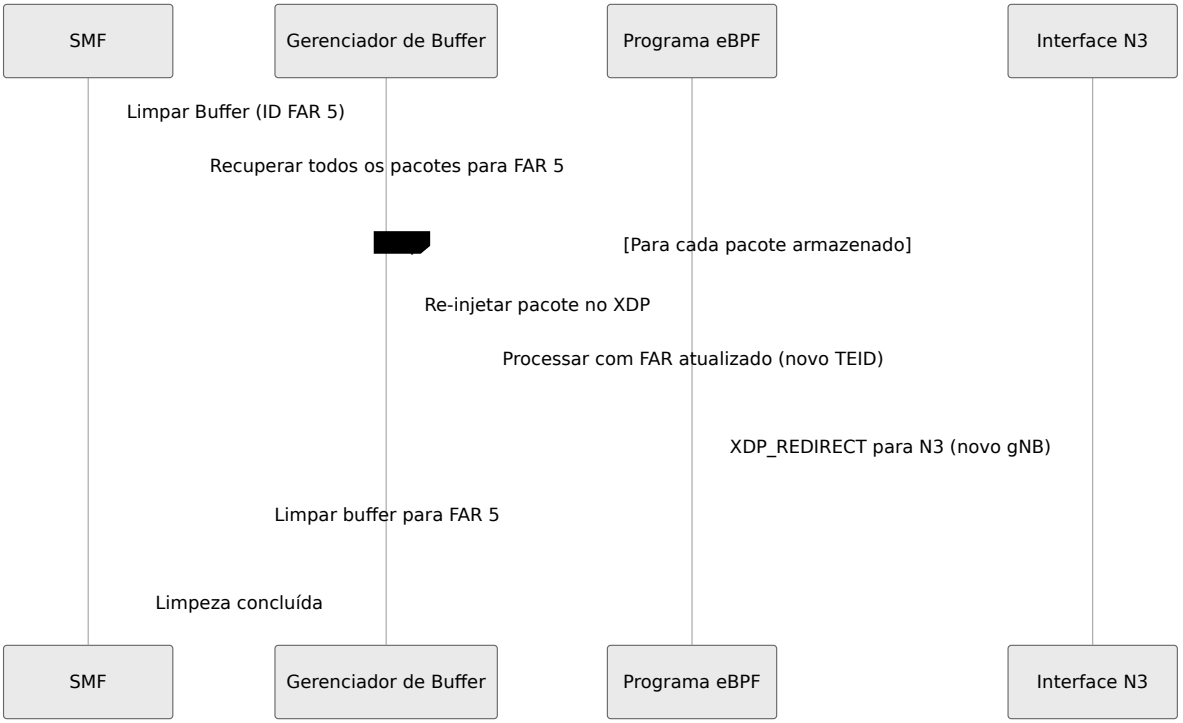
5. **Constrói o Cabeçalho GTP-U**:

```
gtp->version = 1;
gtp->message_type = GTPU_G_PDU;
gtp->teid = htonl(far_id | (direction << 24));  // Codificar ID
FAR e direção
gtp->message_length = htons(orig_packet_len);
```

6. **Retorna XDP_PASS**:
    - O kernel entrega o pacote ao socket UDP local na porta 22152
    - O gerenciador de buffer do espaço de usuário recebe e armazena o pacote

Operação de Limpeza do Buffer

Quando a transferência é concluída, o SMF atualiza o FAR para limpar a flag BUFFER. Os pacotes armazenados são reproduzidos:



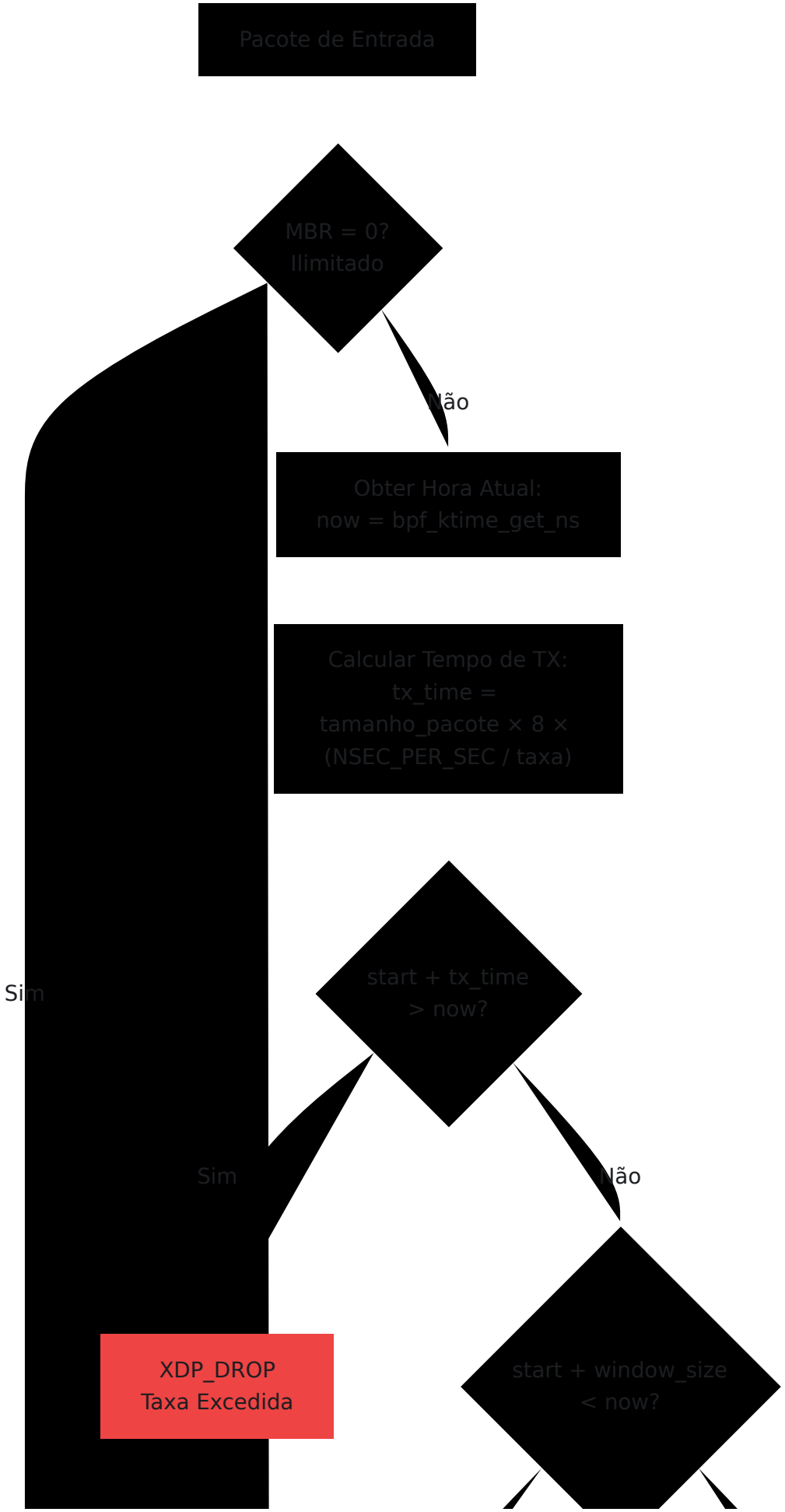
Parâmetros de Gerenciamento de Buffer

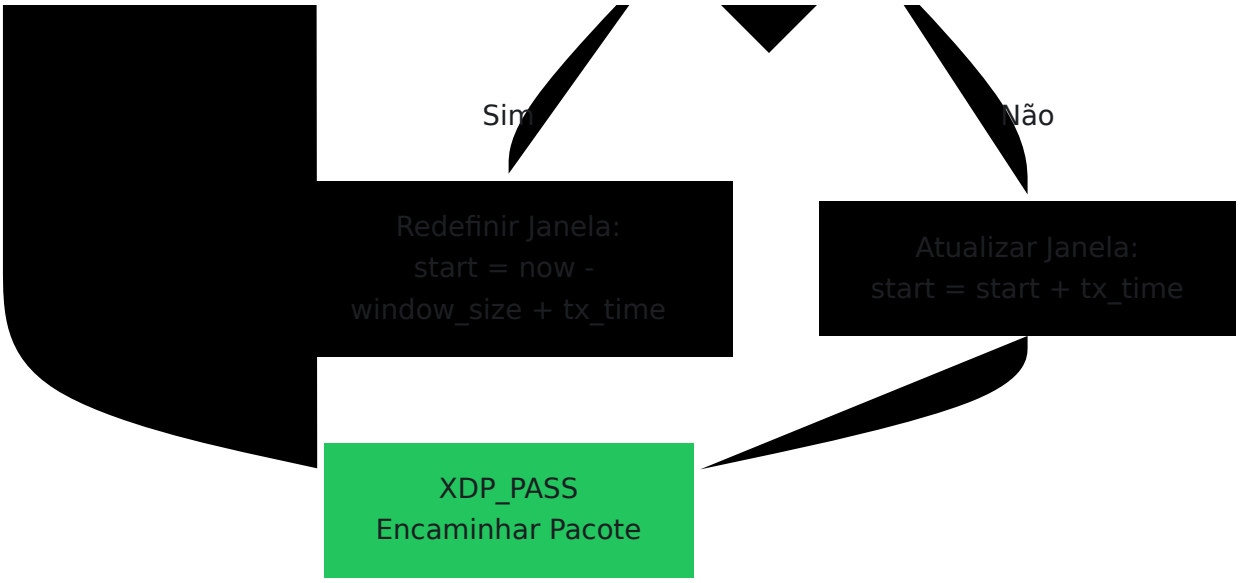
Parâmetro	Padrão	Descrição
Máx por FAR	10.000 pacotes	Máximo de pacotes armazenados por FAR
Máx Total	100.000 pacotes	Máximo total de pacotes armazenados
TTL do Pacote	30 segundos	Tempo antes dos pacotes armazenados expirarem
Porta do Buffer	22152	Porta UDP para entrega do buffer
Intervalo de Limpeza do Buffer	60 segundos	Com que frequência verificar pacotes expirados

Aplicação de QoS

Algoritmo de Limitação de Taxa

OmniUPF implementa um **limitador de taxa de janela deslizante** para aplicação de QoS.


Pacote de Entrada
MBR = 0?
Ilimitado
Não
Obter Hora Atual:
now = bpf_ktime_get_ns
Calcular Tempo de TX:
tx_time =
tamanho_pacote × 8 ×
(NSEC_PER_SEC / taxa)
Sim
start + tx_time
> now?
Sim
Não
XDP_DROP
Taxa Excedida
start + window_size
< now?



Implementação da Janela Deslizante

Algoritmo (do `qer.h`):

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

    // Taxa = 0 significa ilimitado
    if (rate == 0)
        return XDP_PASS;

    // Calcular tempo de transmissão para este pacote
    __u64 tx_time = packet_size * 8 * (NSEC_PER_SEC / rate);
    __u64 now = bpf_ktime_get_ns();

    // Verificar se estamos à frente da janela (o pacote seria
transmitido no futuro)
    __u64 start = *window_start;
    if (start + tx_time > now)
        return XDP_DROP;  // Limite de taxa excedido

    // Se a janela passou, redefini-la
    if (start + window_size < now) {
        *window_start = now - window_size + tx_time;
        return XDP_PASS;
    }

    // Atualizar a janela para contabilizar este pacote
    *window_start = start + tx_time;
    return XDP_PASS;
}
```

Parâmetros Chave:

- **Tamanho da Janela**: 5ms (5.000.000 nanosegundos)
- **Por Direção**: Janelas separadas para uplink e downlink
- **Atualizações Atômicas**: Usa ponteiros voláteis para acesso concorrente
- **MBR = 0**: Tratado como largura de banda ilimitada

Cálculo de Exemplo de QoS

Cenário: MBR = 100 Mbps, Tamanho do Pacote = 1500 bytes

1. **Tempo de Transmissão**:

```
tx_time = 1500 bytes × 8 bits/byte × (1.000.000.000 ns/sec ÷
100.000.000 bps)
tx_time = 1500 × 8 × 10 = 120.000 ns = 120 µs
```

2. **Verificação de Taxa**:

   - Se o último pacote foi transmitido em `t=0`, o próximo pacote pode ser transmitido em `t=120µs`
   - Se o pacote chega em `t=100µs`, ele é descartado (muito cedo)
   - Se o pacote chega em `t=150µs`, ele é encaminhado (janela avançada)

3. **Taxa Máxima de Pacotes**:

```
Max PPS = (100 Mbps ÷ 8) ÷ 1500 bytes = 8.333 pacotes/segundo
Gap entre pacotes = 120 µs
```

Características de Desempenho

Taxa de Transferência

Configuração	Taxa de Transferência	Pacotes/Segundo	Latência
XDP Offload (SmartNIC)	100 Gbps	148 Mpps	< 1 µs
XDP Nativo (NIC 10G, núcleo único)	10 Gbps	8 Mpps	2-5 µs
XDP Nativo (NIC 10G, 4 núcleos)	40 Gbps	32 Mpps	2-5 µs
XDP Genérico	1-5 Gbps	0.8-4 Mpps	50-100 µs

Quebra de Latência

Latência Total de Processamento de Pacotes (XDP Nativo):

Estágio	Latência	Cumulativa
RX NIC	0.5 µs	0.5 µs
Invocação do Hook XDP	0.1 µs	0.6 µs
Busca PDR (Hash)	0.3 µs	0.9 µs
Verificação de Taxa QER	0.1 µs	1.0 µs
Processamento FAR	0.5 µs	1.5 µs
Atualização URR	0.2 µs	1.7 µs
Encapsulação/Decapsulação GTP-U	0.8 µs	2.5 µs
XDP_REDIRECT	0.5 µs	3.0 µs
TX NIC	0.5 µs	3.5 µs

Total: ~3.5 µs por pacote (XDP Nativo, NIC 10G)

Utilização da CPU

Capacidade de Processamento por Núcleo:

- Núcleo único: 8-10 Mpps (XDP Nativo)
- Com hyper-threading: 12-15 Mpps
- Escalonamento multi-núcleo: quase linear até 8 núcleos

Uso da CPU pela Taxa de Pacotes:

```
CPU % ≈ (Taxa de Pacotes / 10.000.000) × 100% por núcleo
```

Exemplo: Tráfego de 2 Mpps utiliza ~20% de um núcleo

Largura de Banda de Memória

Acesso a Mapas eBPF:

- Busca hash: ~100 ns (acerto de cache)
- Busca hash: ~300 ns (erro de cache)
- Busca de array: ~50 ns (sempre acerto de cache)

Largura de Banda de Memória Necessária:

```
Largura de Banda = Taxa de Pacotes × (Tamanho Médio do Pacote + Acessos ao Mapa × 64 bytes)
```

Exemplo: 10 Mpps × (1500 B + 3 buscas × 64 B) ≈ 160 Gbps de largura de banda de memória

Escalabilidade e Ajustes

Escalonamento Horizontal

Múltiplas Instâncias UPF:

Setting SMF as parent of SMF would create a cycle

Tentar novamente

Distribuição de Sessões:

- O SMF distribui sessões entre as instâncias UPF
- Cada UPF lida com um subconjunto de sessões de UE
- Nenhuma comunicação inter-UPF necessária (sem estado)

Escalonamento Vertical

Ajustes de CPU:

1. Habilitar afinidade de CPU para processamento XDP

2. Usar RSS (Receive Side Scaling) para distribuir filas RX
3. Fixar programas eBPF em núcleos específicos

Ajustes de NIC:

1. Aumentar o tamanho do buffer RX
2. Habilitar NICs de múltiplas filas (RSS)
3. Usar diretor de fluxo para direcionamento de tráfego

Ajustes de Kernel:

```
# Aumentar limite de memória bloqueada para mapas eBPF
ulimit -l unlimited

# Desativar balanceamento de IRQ para núcleos XDP
systemctl stop irqbalance

# Definir governador da CPU para desempenho
cpupower frequency-set -g performance

# Aumentar tamanhos de buffer de rede
sysctl -w net.core.rmem_max=134217728
sysctl -w net.core.wmem_max=134217728
```

Planejamento de Capacidade

Fórmula:

```
Núcleos de CPU Necessários = (PPS Esperado ÷ 10.000.000) × 1.5
(50% de margem)
Memória Necessária = (Sessões Máximas × 212 B × 3) + 100 MB (mapas
eBPF + sobrecarga)
Rede Necessária = (Taxa de Transferência de Pico × 2) + 10 Gbps
(margem)
```

Exemplo (1 milhão de sessões, 20 Gbps de pico):

- CPU: (20 Gbps ÷ 10 Gbps por núcleo) × 1.5 = 3-4 núcleos
- Memória: (1M × 212 B × 3) + 100 MB ≈ 750 MB

- Rede: (20 Gbps × 2) + 10 Gbps = 50 Gbps de interfaces

Documentação Relacionada

- **Guia de Operações do UPF** - Operações gerais do UPF e implantação
- **Guia de Gerenciamento de Regras** - Detalhes sobre PDR, FAR, QER, URR
- **Guia de Monitoramento** - Monitoramento de desempenho e métricas
- **Guia de Operações da Interface Web** - Uso do painel de controle
- **Guia de Solução de Problemas** - Problemas comuns e diagnósticos
- **Guia de Jardim Murado** - Redirecionamento fora de crédito usando o mecanismo BUFF para interceptação de pacotes em espaço de usuário

Guia de Configuração do OmniUPF

Índice

1. Visão Geral
2. Modos de Operação
3. Modos de Anexação XDP
4. Parâmetros de Configuração
5. Arquivo de Configuração
6. Compatibilidade com Hypervisor
7. Compatibilidade com NIC
8. Exemplos de Configuração
9. Dimensionamento de Mapas e Planejamento de Capacidade

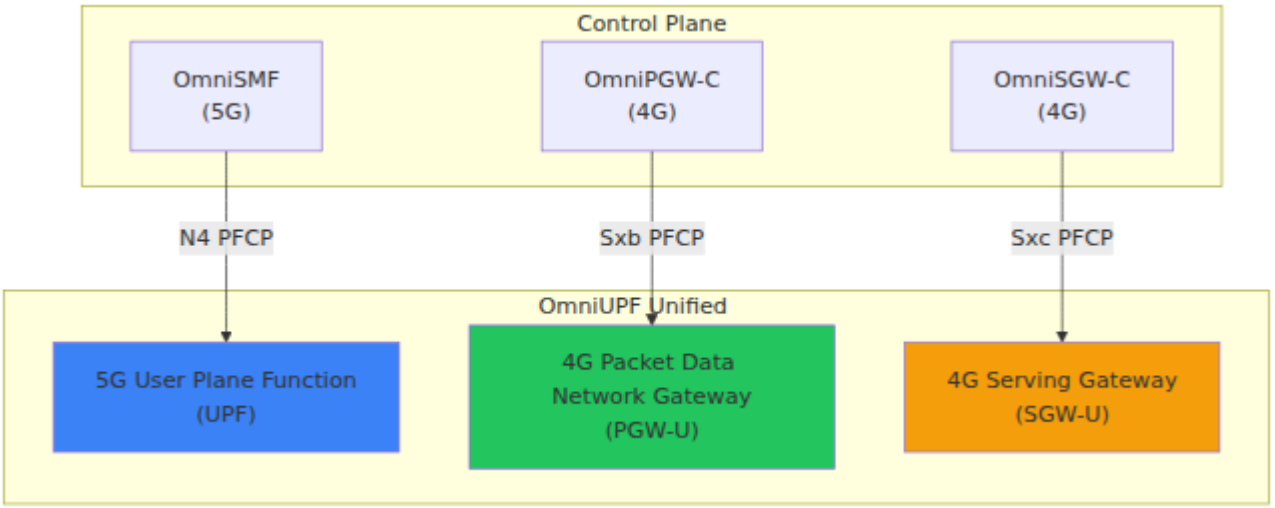
Visão Geral

OmniUPF é uma função de plano de usuário versátil que pode operar em múltiplos modos para suportar redes centrais 4G (EPC) e 5G. O plano de controle é implementado em Elixir/OTP, e a configuração é gerenciada através de um arquivo de configuração Elixir (`runtime.exs`).

Na instalação do pacote, o arquivo de configuração é colocado em `/etc/omniupf/runtime.exs` e preservado durante as atualizações. O UPF deve ser reiniciado após mudanças na configuração.

Modos de Operação

OmniUPF é uma **plataforma unificada** que pode operar simultaneamente como:



Configuração do Modo

O modo de operação é **determinado pelo plano de controle** (SMF, PGW-C ou SGW-C) que estabelece associações PFCP com o OmniUPF. Nenhuma configuração específica do OmniUPF é necessária para alternar entre modos.

Operação Simultânea:

- OmniUPF pode aceitar associações PFCP de múltiplos planos de controle simultaneamente
- Uma única instância do OmniUPF pode atuar como UPF, PGW-U e SGW-U **ao mesmo tempo**
- Sessões de diferentes planos de controle são isoladas e gerenciadas de forma independente

---

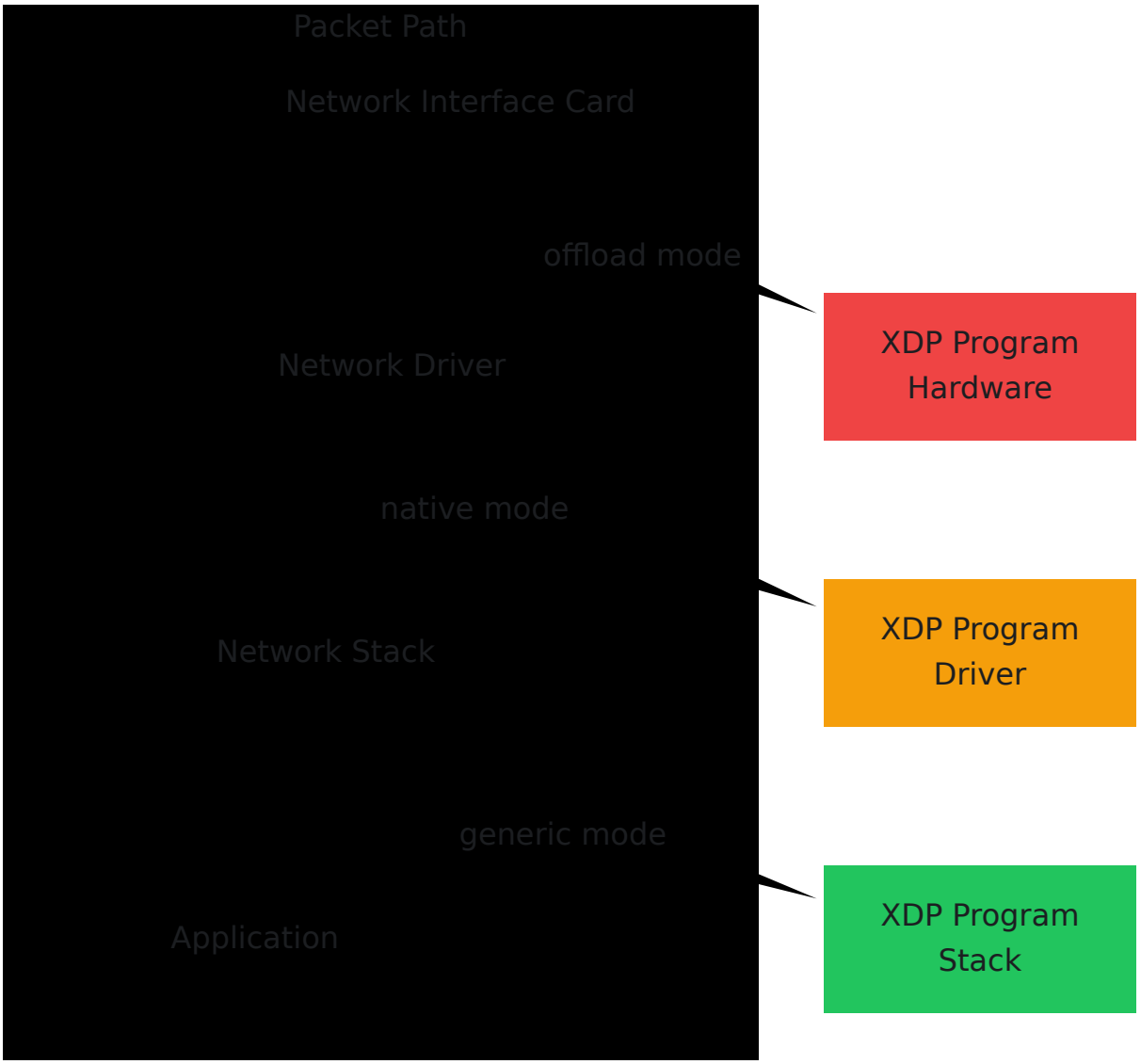
Modos de Anexação XDP

OmniUPF utiliza XDP (eXpress Data Path) para processamento de pacotes de alto desempenho. Três modos de anexação são suportados.

Para instruções detalhadas de configuração do XDP, especialmente para Proxmox e outros hypervisores, veja o Guia de Modos XDP.

Comparação de Modos

Modo	Ponto de Anexação	Desempenho	Caso de Uso	Requisitos de NIC
Genérico	Pilha de rede (kernel)	~1-2 Mpps	Testes, desenvolvimento, compatibilidade	Qualquer NIC
Nativo	Driver de rede (kernel)	~5-10 Mpps	Produção (bare metal, VM com SR-IOV)	Driver compatível com XDP
Offload	Hardware NIC (SmartNIC)	~10-40 Mpps	Produção de alto throughput	SmartNIC com offload XDP



Modo Genérico (Padrão)

Descrição: O programa XDP é executado na pilha de rede do kernel

Vantagens:

- Funciona com **qualquer** interface de rede
- Sem requisitos especiais de driver ou hardware
- Ideal para testes e desenvolvimento
- Compatível com todos os hypervisores e plataformas de virtualização

Desvantagens:

- Desempenho mais baixo (~1-2 Mpps por núcleo)

- Pacotes já passaram pelo driver antes do processamento XDP

Configuração (em `runtime.exs`):

```
xdp_attach_mode = "generic"
```

Melhor para:

- Máquinas virtuais sem SR-IOV
- Ambientes de testes e validação
- NICs sem suporte a driver XDP
- Hypervisores como Proxmox, VMware, VirtualBox

Modo Nativo (Recomendado)

Descrição: O programa XDP é executado no nível do driver de rede

Vantagens:

- Alto desempenho (~5-10 Mpps por núcleo)
- Pacotes processados antes de entrar na pilha de rede
- Latência significativamente menor do que o modo genérico
- Funciona em bare metal e VMs com SR-IOV

Desvantagens:

- Requer driver de rede com suporte a XDP
- Nem todas as NICs/drivers suportam XDP nativo

Configuração (em `runtime.exs`):

```
xdp_attach_mode = "native"
```

Melhor para:

- Implantações de produção em bare metal

- VMs com passthrough SR-IOV
- NICs com drivers compatíveis com XDP (Intel, Mellanox, etc.)

Requisitos:

- Driver de rede compatível com XDP (veja Compatibilidade com NIC)
- Kernel Linux 5.15+ com suporte a XDP habilitado

Modo Offload (Desempenho Máximo)

Descrição: O programa XDP é executado diretamente no hardware SmartNIC

Vantagens:

- Desempenho máximo (~10-40 Mpps)
- Zero sobrecarga de CPU para processamento de pacotes
- Latência sub-microsegundo
- Libera CPU para processamento do plano de controle

Desvantagens:

- Requer hardware SmartNIC caro
- Disponibilidade limitada de SmartNIC
- Configuração e instalação complexas

Configuração (em `runtime.exs`):

```
xdp_attach_mode = "offload"
```

Melhor para:

- Implantações de produção ultra-alto throughput
- Computação de borda com requisitos rigorosos de latência
- Ambientes onde os recursos de CPU são limitados

Requisitos:

- SmartNIC com suporte a offload XDP (Netronome Agilio CX, Mellanox BlueField)
- Firmware e drivers especializados

Parâmetros de Configuração

Interfaces de Rede

Parâmetro	Descrição	Tipo	Padrão
`xdp_interfaces`	Interfaces de rede para tráfego N3/N6/N9 (separadas por vírgula, ou `"auto"` para todas não-loopback)	String	`"auto"`
`gtpu_interface`	NIC que transporta todo GTP-U (N3/N9/S5-S8). Defina junto com `sgi_interface` para anexar um programa XDP apenas GTP-U a esta NIC; deixe não definido para o programa combinado	String	`nil`
`sgi_interface`	NIC que transporta SGi/N6 em IP puro. Defina junto com `gtpu_interface` para anexar um programa XDP apenas de downlink a esta NIC	String	`nil`
`n3_address`	Endereço IPv4 para a interface N3 (GTP-U do RAN)	IP	`"127.0.0.1"`
`n9_address`	Endereço IPv4 para a interface N9 (UPF para UPF para ULCL)	IP	Mesmo que `n3_address`

Exemplo (em `runtime.exs`):

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

Divisão de datapath por NIC. Quando GTP-U e SGi utilizam NICs separadas, nomeie cada uma para que o carregador anexe um programa XDP de propósito único por NIC (apenas GTP-U na `gtpu_interface`, apenas downlink na `sgi_interface`) em vez do programa combinado "on-a-stick". Ambas também devem aparecer em `xdp_interfaces`. Deixe ambas não definidas (o padrão) para anexar o programa combinado a cada interface.

```
xdp_interfaces = "eth1,eth2"
gtpu_interface = "eth1"   # todo GTP-U (N3/N9/S5-S8)
sgi_interface  = "eth2"   # SGi/N6 em IP puro
```

Configuração PFCP

Parâmetro	Descrição	Tipo	Padrão
`pfcp_address`	Endereço local para o servidor PFCP (interface N4/Sxb/Sxc)	IP String	`"127.0.0.1"`
`pfcp_port`	Porta PFCP	Inteiro	`8805`
`node_id`	ID do Nó Local para o protocolo PFCP	IP String	`"127.0.0.1"`
`heartbeat_interval_ms`	Intervalo de heartbeat PFCP (milissegundos)	Inteiro	`5000`
`heartbeat_timeout_ms`	Timeout de heartbeat PFCP (milissegundos)	Inteiro	`5000`
`heartbeat_retries`	Número de tentativas de heartbeat antes de declarar o par como inativo	Inteiro	`3`
`association_setup_timeout`	Timeout aguardando uma Resposta de Configuração de Associação PFCP quando o UPF inicia uma	Inteiro	`5000`

Parâmetro	Descrição	Tipo	Padrão
	associação (milissegundos)		
`pfcp_remote_nodes`	Lista de strings de endereços IP do plano de controle com os quais o UPF ativa associações na inicialização (iniciadas pelo UPF). Vazio significa que o UPF aceita apenas associações iniciadas pelo plano de controle.	Lista de String	`[]`

Exemplo (em `runtime.exs`):

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

API e Monitoramento

Parâmetro	Descrição	Tipo	Padrão
`api_port`	Porta local para o servidor REST API (Phoenix)	Inteiro	`8080`
`log_level`	Nível de log (`:debug`, `:info`, `:warning`, `:error`)	Átomo	`:info`

Exemplo (em `runtime.exs`):

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

Gerenciamento de Caminho GTP

Parâmetro	Descrição	Tipo	Padrão
`gtpu_port`	Porta GTP-U	Inteiro	`2152`
`gtp_echo_interval`	Intervalo de eco padrão em segundos	Inteiro	`10`
`gtp_echo_retries`	Máximo de ecos perdidos antes da falha do caminho	Inteiro	`3`
`gtp_peers`	Lista de pares GTP para keepalives de Echo Request	Lista	`[]`

Exemplo (em `runtime.exs`):

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

GTP-U de Downlink Oversize

GTP-U adiciona 36 bytes a cada pacote de downlink. Um pacote de tamanho total de 1500 bytes da internet, portanto, se torna um pacote GTP-U de 1536 bytes, que não se encaixa em um link N3 de 1500 bytes. O remetente de GTP-U é obrigado a fragmentar tais pacotes (TS 29.281 §4.2.4), e muitos UEs e roteadores ignoram o MTU do link oferecido no PCO, então isso acontece na operação normal.

O caminho rápido do XDP não pode fragmentar. Quando um pacote GTP-U de downlink é maior que o MTU de saída, ele é entregue ao caminho lento do espaço do usuário, que o envia de um socket UDP comum para que o kernel fragmenta o pacote externo. Isso se aplica tanto quando o UPF encapsula (N6 para N3) quanto quando ele retransmite entre túneis GTP-U (SGW-U).

Parâmetro	Descrição	Tipo	Padrão
`gtpu_oversize_punt_enabled`	`true`: GTP-U de downlink oversize é enviado para o espaço do usuário e fragmentado. `false`: é descartado	Booleano	nenhum, **obrigatório**

O parâmetro é obrigatório: o UPF não iniciará sem ele. Cada pacote punido deixa o caminho rápido, então um assinante enviando tráfego sustentado de

tamanho total através de um MTU de túnel menor (por exemplo, um download em massa para um dispositivo que ignora o MTU do PCO) move essa carga para o espaço do usuário. Observe `upf_gtpu_oversize_punted_total` (veja METRICS.md) e defina o parâmetro como `false` para aliviar a carga, à custa de descartar esses pacotes.

Exemplo (em `runtime.exs`):

```
gtpu_oversize_punt_enabled = true
```

Os pacotes GTP-U fragmentados recebidos do RAN são reassemblados pelo kernel e reinjetados no datapath. Isso não requer configuração.

Limitação de TCP MSS

Um dispositivo que ignora o MTU do link no PCO (ou Anúncio de Roteador) anuncia um MSS de 1460 bytes em seu handshake TCP. Ambos os lados então enviam pacotes de 1500 bytes, e cada um deles passa pelo caminho lento de oversize descrito acima. Limitar o MSS no handshake mantém esse tráfego no caminho rápido.

Parâmetro	Descrição	Tipo	Padrão
`tcp_mss_clamp`	Maior TCP MSS (IPv4) que um SYN ou SYN-ACK pode anunciar através deste UPF. Valores maiores são reduzidos a ele. IPv6 usa um valor 20 menor. `false` deixa o MSS inalterado	`false` ou inteiro 536-1460	nenhum, **obrigatório**

Somente pacotes SYN e SYN-ACK são alterados, um ou dois por conexão; pacotes de dados não são tocados. Ambas as direções são limitadas: o SYN do UE limita o que o servidor envia para o UE, e o SYN-ACK do servidor limita o que o UE envia. O checksum TCP é atualizado incrementalmente. A limitação se aplica onde o UPF decapsula ou encapsula o pacote do usuário (PGW-U / UPF), e não em uma retransmissão pura de SGW-U.

Defina-o para o MTU do link IPv4 oferecido no PCO menos 40 bytes: `1310` para um MTU de 1350.

Exemplo (em `runtime.exs`):

```
tcp_mss_clamp = 1310
```

A limitação cobre apenas TCP. Tráfego UDP e outros maiores que o MTU do túnel ainda seguem o caminho lento de oversize, assim como TCP cujo handshake o UPF não viu.

Capacidade do Mapa eBPF

Parâmetro	Descrição	Tipo	Padrão
`max_sessions`	Número máximo de sessões concorrentes	Inteiro	`1_000_000`
`far_map_size`	Tamanho do mapa eBPF FAR	Inteiro	`131_070`
`qer_map_size`	Tamanho do mapa eBPF QER	Inteiro	`65_535`
`urr_map_size`	Tamanho do mapa eBPF URR	Inteiro	`65_535`

Exemplo (em `runtime.exs`):

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

Configuração de Buffer

Pacotes de downlink são armazenados em buffer enquanto um UE está ocioso, e depois são descarregados quando o UE é chamado. Esses parâmetros limitam quanto é mantido e por quanto tempo.

Parâmetro	Descrição	Tipo	Padrão
`buffer_port`	Porta UDP para pacotes armazenados em buffer do eBPF	Inteiro	`22152`
`buffer_max_total`	Máximo de pacotes armazenados em buffer em todos os FARs (limite global)	Inteiro	`10_000`
`buffer_max_per_far`	Máximo de pacotes armazenados em buffer por FAR	Inteiro	`100`
`buffer_ttl_ms`	Tempo que um pacote armazenado em buffer é mantido antes de ser descartado (milissegundos)	Inteiro	`30_000`
`buffer_cleanup_interval_ms`	Intervalo em que pacotes armazenados em buffer expirados são varridos (milissegundos)	Inteiro	`5_000`

Exemplo (em `runtime.exs`):

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

Descartes causados por esses limites são contados por `upf_buffer_packets_dropped_total` (rótulos `global_limit`, `far_limit`, `expired`). Veja a Referência de Métricas.

Flags de Recursos

Parâmetro	Descrição	Tipo	Padrão
`feature_ueip`	Habilitar alocação de IP do UE pelo OmniUPF	Booleano	`false`
`ueip_pool`	Pool de IP para alocação de IP do UE (requer `feature_ueip`)	String CIDR	`"10.60.0.0/24"`
`feature_ftup`	Habilitar alocação de F-TEID pelo OmniUPF	Booleano	`true`
`teid_pool_start`	Início do intervalo de alocação de TEID (inclusivo)	Inteiro	`1`
`teid_pool_end`	Fim do intervalo de alocação de TEID (inclusivo). TEIDs são escolhidos aleatoriamente do intervalo e não são reutilizados imediatamente, então mantenha o intervalo amplo	Inteiro	`0xFFFFFFFF`

Exemplo (alocação de IP do UE) (em `runtime.exs`):

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

Exemplo (alocação de F-TEID) (em `runtime.exs`):

```
feature_ftup = true
teid_pool_start = 1
teid_pool_end = 0xFFFFFFFF
```

Configuração do Gerenciador de Roteamento

Para sincronização de rotas do UE com o daemon FRR (Free Range Routing). Veja o Guia de Gerenciamento de Rotas para detalhes.

Parâmetro	Descrição	Tipo	
`route_manager_enabled`	Habilitar sincronização automática de rotas do UE	Booleano	`tru`
`route_manager_type`	Tipo de daemon de roteamento (`"frr"` ou `"static"`)	String	`"fr`
`route_manager_nexthop`	Endereço do próximo salto para rotas do UE injetadas (vazio = diretamente conectado)	String	`""`
`route_manager_vtysh_path`	Caminho para o binário FRR `vtysh` (usado quando `route_manager_type` é `"frr"`)	String	`"/u`
`route_manager_sync_interval_ms`	Intervalo de reconciliação total entre rotas rastreadas e o daemon de roteamento (milissegundos)	Inteiro	`600`
`route_batch_interval_ms`	Janela de debounce para agrupar operações incrementais de adição/remover	Inteiro	`1_0`

Parâmetro	Descrição	Tipo	
	rotas (milissegundos)		

Exemplo (em `runtime.exs`):

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

Quando Habilitar:

- Implantações Multi-UPF que requerem anúncio de rotas
- Integração com protocolos de roteamento OSPF ou BGP
- Requer daemon FRRouting instalado e configurado

Configuração eBPF / XDP

Parâmetro	Descrição	Tipo	Padrão
`xdp_interfaces`	Lista separada por vírgulas de interfaces para XDP, ou `"auto"` para todas não-loopback	String	`"auto"`
`xdp_attach_mode`	Modo XDP: `"generic"`, `"native"` ou `"offload"`	String	`"generic"`
`ebpf_pin_path`	Caminho do sistema de arquivos BPF para pinagem	String	`"/sys/fs/bpf/upf_pipeline"`
`xdp_obj_path`	Caminho para o objeto eBPF compilado	String	`"/etc/omniupf/ipentrypoint_bp`
`ebpf_backend`	Backend do mapa do	Átomo	`:nif`

Parâmetro	Descrição	Tipo	Padrão
	dataplane: `:nif` (NIF direto, padrão), `:port` (processo de porta externo), ou `:mock` (em memória, para testes sem eBPF)		

Exemplo (em `runtime.exs`):

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

Arquivo de Configuração

Arquivo de Configuração Elixir (runtime.exs)

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

O arquivo de configuração utiliza a sintaxe Elixir com atribuições de variáveis simples para legibilidade. As variáveis são aplicadas à configuração da aplicação na parte inferior do arquivo via `config :upf_ex, ...`.

```
import Config

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

#
============================================================================
# Configuração PFCP (N4/Sx)
#
============================================================================
pfcp_address = "10.100.50.241"       # Endereço IP para escutar mensa
PFCP
pfcp_port = 8805                      # Porta PFCP (padrão: 8805)
node_id = "10.100.50.241"            # ID do Nó anunciado na Configur
Associação

#
============================================================================
# Configuração do Plano de Dados GTP-U
#
============================================================================
n3_address = "10.100.50.233"         # IP da interface N3 (GTP-U em c
ao RAN)
n9_address = n3_address               # IP da interface N9 (padrão pa
gtpu_port = 2152                      # Porta GTP-U (padrão: 2152)
buffer_port = 22152                   # Porta do listener de buffer

#
============================================================================
# Configuração eBPF / XDP
#
============================================================================
xdp_interfaces = "auto"              # "auto" para todas não-loopback
lista separada por vírgulas
xdp_attach_mode = "native"           # Modo XDP: "generic", "native"
"offload"
ebpf_pin_path = "/sys/fs/bpf/upf_pipeline"
xdp_obj_path = "/etc/omniupf/ipentrypoint_bpf.o"

#
============================================================================
```

```
# Configuração de Sessão e Pool de Recursos
#
============================================================================
max_sessions = 100_000
teid_pool_start = 1
teid_pool_end = 0xFFFFFFFF
far_map_size = 131_070
qer_map_size = 65_535
urr_map_size = 65_535

#
============================================================================
# Flags de Recursos
#
============================================================================
feature_ueip = true
feature_ftup = true
ueip_pool = "10.45.0.0/16"

#
============================================================================
# Gerenciamento de Rotas
#
============================================================================
route_manager_enabled = true
route_manager_type = "frr"

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

#
============================================================================
# Monitoramento de Echo de Pares GTP
#
============================================================================
gtp_echo_interval = 10
gtp_echo_retries = 3
gtp_peers = [
```

```
  %{address: parse_ip.("10.100.50.50"), echo: true, echo_interval: 10
]

#
# ============================================================================
# API HTTP e Logging
#
# ============================================================================
api_port = 8080
log_level = :info

#
# ============================================================================
# Aplicar Configuração (não edite abaixo desta linha)
#
# ============================================================================
config :upf_ex,
  pfcp_address: parse_ip.(pfcp_address),
  pfcp_port: pfcp_port,
  n3_address: parse_ip.(n3_address),
  n9_address: parse_ip.(n9_address),
  node_id: parse_ip.(node_id),
  api_port: api_port,
  ebpf_pin_path: ebpf_pin_path,
  xdp_obj_path: xdp_obj_path,
  interface_names: String.split(xdp_interfaces, ","),
  xdp_attach_mode: xdp_attach_mode,
  feature_ueip: feature_ueip,
  feature_ftup: feature_ftup,
  ueip_pool: ueip_pool,
  teid_pool_start: teid_pool_start,
  teid_pool_end: teid_pool_end,
  far_map_size: far_map_size,
  qer_map_size: qer_map_size,
  urr_map_size: urr_map_size,
  max_sessions: max_sessions,
  route_manager_enabled: route_manager_enabled,
  route_manager_type: route_manager_type,
  buffer_port: buffer_port,
  gtpu_port: gtpu_port,
  heartbeat_interval_ms: heartbeat_interval_ms,
  heartbeat_timeout_ms: heartbeat_timeout_ms,
  heartbeat_retries: heartbeat_retries,
  gtp_echo_interval: gtp_echo_interval,
```

```
  gtp_echo_retries: gtp_echo_retries,
  gtp_peers: gtp_peers

config :logger, level: log_level
```

Gerenciando o Serviço

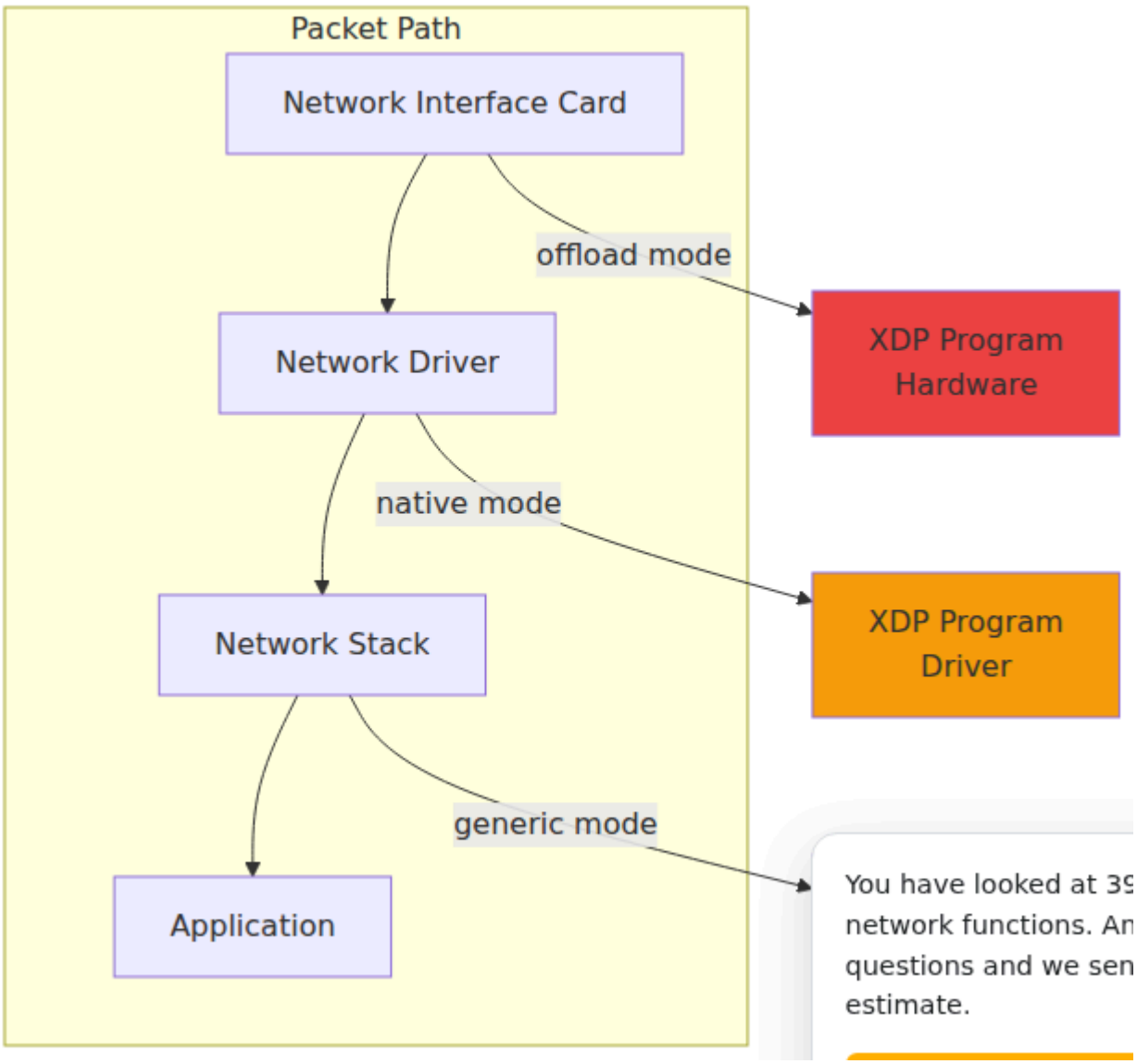
```
# Iniciar o UPF
sudo systemctl start omniupf

# Parar o UPF
sudo systemctl stop omniupf

# Reiniciar após alterações de configuração
sudo systemctl restart omniupf

# Verificar status
sudo systemctl status omniupf

# Visualizar logs
sudo journalctl -u omniupf -f

# Ou use o binário de lançamento diretamente
sudo /opt/omniupf/bin/upf_ex start
sudo /opt/omniupf/bin/upf_ex stop
sudo /opt/omniupf/bin/upf_ex daemon   # iniciar como daemon em
segundo plano
```

Compatibilidade com Hypervisor

Visão Geral

OmniUPF é compatível com todos os principais hypervisores e plataformas de virtualização. O modo de anexação XDP e a configuração de rede dependem das capacidades de rede do hypervisor.

Para instruções passo a passo sobre como habilitar o XDP nativo no Proxmox e outros hypervisores, consulte o Guia de Modos XDP.



Proxmox VE

Configurações Suportadas:

1. Modo Bridge (XDP Genérico)

Caso de uso: Rede padrão de VM

Configuração:

- Dispositivo de Rede: VirtIO ou E1000

- Modo XDP: `genérico`
- Desempenho: ~1-2 Mpps

Configurações da VM Proxmox:

```
Dispositivo de Rede: net0
Modelo: VirtIO (paravirtualizado)
Bridge: vmbr0
```

Configuração do OmniUPF (em `runtime.exs`):

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

---

2. SR-IOV Passthrough (XDP Nativo)

Caso de uso: Produção de alto desempenho

Configuração:

- Dispositivo de Rede: Função Virtual SR-IOV
- Modo XDP: `nativo`
- Desempenho: ~5-10 Mpps

Requisitos:

- NIC física com suporte a SR-IOV (Intel X710, Mellanox ConnectX-5)
- SR-IOV habilitado na BIOS
- IOMMU habilitado (`intel_iommu=on` ou `amd_iommu=on` no GRUB)

Habilitar SR-IOV no Proxmox:

```
# Editar configuração do GRUB
nano /etc/default/grub

# Adicionar ao GRUB_CMDLINE_LINUX_DEFAULT:
intel_iommu=on iommu=pt

# Atualizar GRUB e reiniciar
update-grub
reboot

# Habilitar VFs na NIC (exemplo: 4 funções virtuais em eth0)
echo 4 > /sys/class/net/eth0/device/sriov_numvfs

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

Configurações da VM Proxmox:

```
Hardware -> Adicionar -> Dispositivo PCI
Selecionar: Função Virtual SR-IOV
Todas as Funções: Não
GPU Primária: Não
PCI-Express: Sim (opcional)
```

Configuração do OmniUPF (em `runtime.exs`):

```
xdp_interfaces = "ens1f0"    # nome da VF SR-IOV
xdp_attach_mode = "native"
```

---

3. PCI Passthrough (XDP Nativo)

Caso de uso: NIC dedicada para uma única VM

Configuração:

- Toda a NIC física passada para a VM

- Modo XDP: `nativo` ou `offload` (se SmartNIC)
- Desempenho: ~5-40 Mpps (depende da NIC)

Configurações da VM Proxmox:

```
Hardware -> Adicionar -> Dispositivo PCI
Selecionar: NIC Física (ex: 0000:01:00.0)
Todas as Funções: Sim
GPU Primária: Não
PCI-Express: Sim
```

Configuração do OmniUPF (em `runtime.exs`):

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

KVM/QEMU

Modo Bridge:

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

SR-IOV Passthrough:

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

VMware ESXi

vSwitch Padrão (XDP Genérico):

- Adaptador de Rede: VMXNET3
- Modo XDP: `genérico`

SR-IOV (XDP Nativo):

- Habilitar SR-IOV nas configurações do host ESXi
- Adicionar adaptador de rede SR-IOV à VM
- Modo XDP: `nativo`

Microsoft Hyper-V

Switch Virtual (XDP Genérico):

- Adaptador de Rede: Sintético
- Modo XDP: `genérico`

SR-IOV (XDP Nativo):

- Habilitar SR-IOV no Gerenciador Hyper-V
- Configurar SR-IOV no adaptador de rede virtual
- Modo XDP: `nativo`

VirtualBox

Modo NAT/Bridge (apenas XDP Genérico):

- Adaptador de Rede: VirtIO-Net ou Intel PRO/1000
- Modo XDP: `genérico`
- Nota: O VirtualBox **não** suporta SR-IOV

Compatibilidade de NIC

Entendendo Mpps vs Throughput

Pacotes por segundo (Mpps) e throughput (Gbps) não são diretamente equivalentes - a relação depende inteiramente do tamanho do pacote. O tráfego de rede móvel varia dramaticamente em tamanho de pacote, desde pequenos pacotes de VoIP até grandes quadros de streaming de vídeo.

Impacto do Tamanho do Pacote no Throughput

Em redes móveis, o UPF processa pacotes encapsulados GTP-U na interface N3 e pacotes IP nativos na interface N6.

Sobrecarga de Encapsulação GTP-U (Interface N3):

- **Cabeçalho IPv4 externo**: 20 bytes
- **Cabeçalho UDP externo**: 8 bytes
- **Cabeçalho GTP-U**: 8 bytes
- **Total de sobrecarga GTP-U**: 36 bytes

Pacote GTP-U Mínimo (N3):

- **Cabeçalho IP interno**: 20 bytes (IPv4)
- **Cabeçalho UDP interno**: 8 bytes
- **Carga útil mínima**: 1 byte
- **Total do pacote interno**: 29 bytes
- **Mais sobrecarga GTP-U**: 36 bytes
- **Tamanho total do pacote**: 65 bytes

Throughput a 1 Mpps com pacotes GTP-U mínimos:

```
65 bytes x 1,000,000 pps x 8 bits/byte = 520 Mbps
```

Pacote GTP-U Máximo (N3 com 1500 MTU):

- **MTU IP interno**: 1500 bytes (pacote IP interno completo)

- **Mais sobrecarga GTP-U**: 36 bytes
- **Tamanho total do pacote**: 1536 bytes

Throughput a 1 Mpps com pacotes GTP-U máximos:

```
1536 bytes x 1,000,000 pps x 8 bits/byte = 12,288 Mbps ~ 12.3 Gbps
```

Pacotes IP Nativos (Interface N6):

Na N6 (em direção à Internet), os pacotes são IP nativos sem GTP-U:

Pacote N6 Mínimo:

- **Cabeçalho IP**: 20 bytes
- **Cabeçalho UDP**: 8 bytes
- **Carga útil mínima**: 1 byte
- **Total**: 29 bytes

Throughput a 1 Mpps com pacotes N6 mínimos:

```
29 bytes x 1,000,000 pps x 8 bits/byte = 232 Mbps
```

Pacote N6 Máximo (1500 MTU):

- **MTU IP**: 1500 bytes
- **Total**: 1500 bytes

Throughput a 1 Mpps com pacotes N6 máximos:

```
1500 bytes x 1,000,000 pps x 8 bits/byte = 12,000 Mbps = 12 Gbps
```

Exemplos de Desempenho do Mundo Real

NIC Intel X710 (capacidade de 10 Mpps na interface N3 com GTP-U):

Padrão de Tráfego	Tamanho do Pacote Interno	Total GTP-U	Throughput a 10 Mpps	Caso de Uso Típico
Chamadas VoIP (N3)	65-150 bytes	101-186 bytes	**0.8-1.5 Gbps**	Voz AMR-WB, G.711
Web leve (N3)	400-600 bytes	436-636 bytes	**3.5-5.1 Gbps**	HTTP/HTTPS, mensagens
Móvel moderno (N3)	**1200 bytes**	**1236 bytes**	**9.9 Gbps**	**Mistura de tráfego típica de 2024**
Streaming de vídeo (N3)	1400-1450 bytes	1436-1486 bytes	**11.5-11.9 Gbps**	Fragmentos de vídeo HD/4K
MTU Máxima (N3)	1500 bytes	1536 bytes	**12.3 Gbps**	Grandes downloads de TCP

Na interface N6 (IP nativo, sem GTP-U):

Padrão de Tráfego	Tamanho do Pacote	Throughput a 10 Mpps	Caso de Uso Típico
Pacotes VoIP	65-150 bytes	**0.5-1.2 Gbps**	Fluxos de voz RTP
Web leve	400-600 bytes	**3.2-4.8 Gbps**	Requisições HTTP
Móvel moderno	**1200 bytes**	**9.6 Gbps**	**Tráfego típico de 2024**
Streaming de vídeo	1400-1450 bytes	**11.2-11.6 Gbps**	Downloads de vídeo
MTU Máxima	1500 bytes	**12.0 Gbps**	Transferências de arquivos grandes

A 10 Mpps com tráfego móvel moderno (média de 1200 bytes), espere **~10 Gbps de throughput** nas interfaces N3 e N6.

Planejamento de Capacidade Prática:

Com tamanho médio de pacote de 1200 bytes (típico para redes móveis modernas com streaming de vídeo):

Capacidade Mpps da NIC	Throughput N3 (GTP-U)	Throughput N6 (IP Nativo)	Implantação Realista
1 Mpps	~1.0 Gbps	~1.0 Gbps	Site de pequena célula, gateway IoT
5 Mpps	~4.9 Gbps	~4.8 Gbps	Site de célula média, empresa
10 Mpps	~9.9 Gbps	~9.6 Gbps	Site de célula grande, pequena cidade
20 Mpps	~19.7 Gbps	~19.2 Gbps	Área metropolitana, cidade média
40 Mpps	~39.4 Gbps	~38.4 Gbps	Grande metropolitana, hub regional

---

Drivers de Rede Compatíveis com XDP

OmniUPF requer drivers de rede com suporte a XDP para modos **nativo** e **offload**. O modo genérico funciona com **qualquer** NIC.

NICs Intel

Modelo	Driver	Suporte a XDP	Modo	Desempenho
Intel X710	i40e	Sim	Nativo	~10 Mpps
Intel XL710	i40e	Sim	Nativo	~10 Mpps
Intel E810	ice	Sim	Nativo	~15 Mpps
Intel 82599ES	ixgbe	Sim	Nativo	~8 Mpps
Intel I350	igb	Limitado	Genérico	~1 Mpps
Intel E1000	e1000	Não	Apenas Genérico	~1 Mpps

NICs Mellanox/NVIDIA

Modelo	Driver	Suporte a XDP	Modo	Desempenho
Mellanox ConnectX-5	mlx5	Sim	Nativo	~12 Mpps
Mellanox ConnectX-6	mlx5	Sim	Nativo	~20 Mpps
Mellanox BlueField	mlx5	Sim	Nativo + Offload	~40 Mpps
Mellanox ConnectX-4	mlx4	Limitado	Genérico	~2 Mpps

NICs Broadcom

Modelo	Driver	Suporte a XDP	Modo	Desempenho
Broadcom BCM57xxx	bnxt_en	Sim	Nativo	~8 Mpps
Broadcom NetXtreme II	bnx2x	Não	Apenas Genérico	~1 Mpps

Outros Fornecedores

Modelo	Driver	Suporte a XDP	Modo	Desempenho
Netronome Agilio CX	nfp	Sim	Offload	~30 Mpps
Amazon ENA	ena	Sim	Nativo	~5 Mpps
Solarflare SFC9xxx	sfc	Sim	Nativo	~8 Mpps
VirtIO	virtio_net	Limitado	Genérico	~2 Mpps

Verificando o Suporte a XDP da NIC

Verifique se o driver suporta XDP:

```
# Encontrar driver da NIC
ethtool -i eth0 | grep driver

# Verificar suporte a XDP no driver
modinfo <driver_name> | grep -i xdp

# Exemplo para Intel i40e
modinfo i40e | grep -i xdp
```

Verifique a anexação do programa XDP:

```
# Verifique se o programa XDP está anexado
ip link show eth0 | grep -i xdp

# Exemplo de saída (XDP anexado):
# 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 xdp qdisc mq
```

NICs Recomendadas por Caso de Uso

Com tamanho médio de pacote de 1200 bytes (tráfego móvel moderno):

Caso de Uso	NIC Recomendada	Modo	Capacidade Mpps	Th
Teste/Desenvolvimento	Qualquer NIC (VirtIO, E1000)	Genérico	1-2 Mpps	1-
Site de Pequena Célula	Intel X710, Mellanox CX-5	Nativo	5-10 Mpps	5-
Célula Média/Metrô	Intel E810, Mellanox CX-6	Nativo	10-20 Mpps	10
Grande Metrô	Mellanox CX-6, Intel E810 (duplo)	Nativo	20-40 Mpps	20
Hub Regional	Mellanox BlueField, Netronome Agilio	Offload	40+ Mpps	40
VM Proxmox (Bridge)	VirtIO	Genérico	1-2 Mpps	1-
VM Proxmox (SR-IOV)	Intel X710/E810 VF, Mellanox CX-5 VF	Nativo	5-10 Mpps	5-

---

Recursos Adicionais

Documentação Oficial do XDP:

- Projeto XDP
- Documentação do Kernel XDP

Listas de Compatibilidade de NIC:

- Suporte a Hardware XDP do Cilium
- Drivers XDP do IO Visor

Exemplos de Configuração

Exemplo 1: Ambiente de Desenvolvimento (Modo Genérico)

Cenário: Testando o OmniUPF em laptop ou VM sem SR-IOV

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

Exemplo 2: Bare Metal de Produção (Modo Nativo)

Cenário: UPF de produção em servidor bare metal com NIC Intel X710

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

Exemplo 3: VM Proxmox com SR-IOV (Modo Nativo)

Cenário: UPF de produção em VM Proxmox com passthrough SR-IOV

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

Exemplo 4: Modo PGW-U (EPC 4G)

Cenário: OmniUPF atuando como PGW-U em rede EPC 4G

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

Exemplo 5: Múltiplos Modos (UPF + PGW-U Simultaneamente)

Cenário: OmniUPF atendendo simultaneamente redes 5G e 4G

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

Exemplo 6: Modo Offload de SmartNIC

Cenário: Implantação de ultra-alto throughput com Netronome Agilio CX SmartNIC

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

Dimensionamento de Mapas e Planejamento de Capacidade

Definindo Tamanhos de Mapas

Defina `max_sessions` e configure tamanhos de mapa para sua implantação:

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

Uso de memória: ~91 MB para 100K sessões

Estimativa de Capacidade

Calcule o n��mero máximo de sessões:

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

Exemplo:

- Mapa PDR: 200,000
- Mapa FAR: 200,000
- Mapa QER: 100,000

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

Requisitos de Memória

Uso de memória por sessão:

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

Para 100K sessões: ~52 MB de memória do kernel

Recomendação: Certifique-se de que o limite de memória bloqueada permita 2x o uso estimado:

```
# Verifique o limite atual
ulimit -l

# Defina como ilimitado (necessário para eBPF)
ulimit -l unlimited
```

Documentação Relacionada

- **Guia de Arquitetura** - detalhes técnicos do eBPF/XDP e otimização de desempenho
- **Guia de Gerenciamento de Regras** - configuração de PDR, FAR, QER, URR
- **Guia de Monitoramento** - estatísticas, monitoramento de capacidade e alertas
- **Referência de Métricas** - referência completa de métricas do Prometheus
- **Guia da Interface Web** - operações do painel de controle
- **Guia de Operações** - visão geral da arquitetura e implantação do UPF
- **Guia de Jardim Murado** - redirecionamento fora de crédito, portal cativo e configuração de lista branca

IPv6 / Dual-Stack no Plano do Usuário

Como o OmniUPF lida com o tráfego de usuário IPv6 para sessões PDN IPv6 e IPv4v6: os pré-requisitos do kernel, o pool de endereços do UE, a programação do downlink IPv6 a partir do PFCP e a publicidade de rotas de host do UE para a rede via OSPFv3.

Esta página cobre as responsabilidades do **plano do usuário**. O plano de controle (negociação do tipo PDN, PAA, Gx e sinalização do endereço IPv6 do UE para o UPF via PFCP) é tratado pelo SMF/PGW-C. Veja também Gerenciamento de Rotas e o Guia de Configuração.

Índice

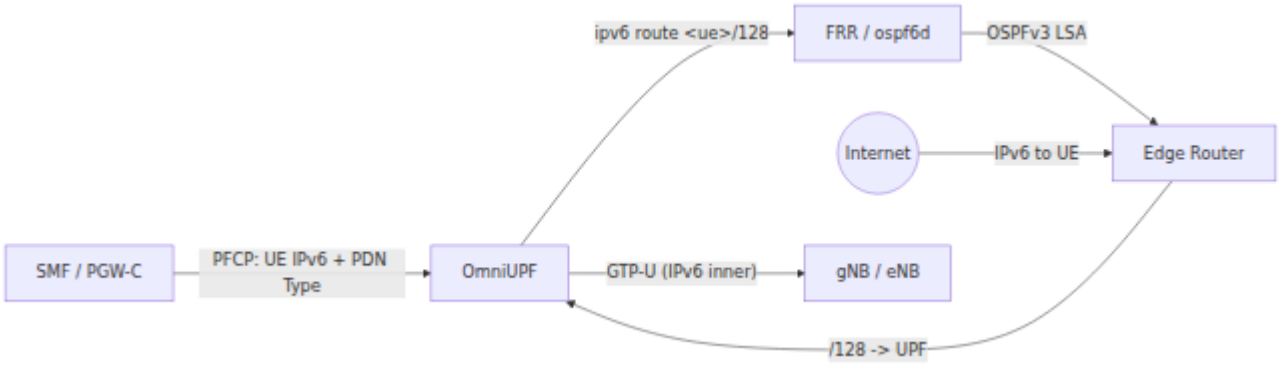
- Visão Geral
- Pré-requisitos do Kernel
- Pool de Endereços IPv6 do UE
- Publicidade de Rotas OSPFv3
- Comportamento do Plano de Dados
- Solução de Problemas

Visão Geral

Quando uma sessão PDN IPv6 (ou IPv4v6) é estabelecida, o SMF/PGW-C envia o endereço IPv6 do UE e o tipo de PDN para o UPF via PFCP. O UPF então:

1. Programa a **busca de downlink IPv6** para que pacotes destinados ao UE sejam correspondidos e encapsulados.
2. Adiciona uma **rota de host** `/128` para o endereço do UE e a publica para o roteador de borda via **OSPFv3**, para que o tráfego de downlink seja direcionado a este UPF.

3. Encaminha o tráfego do usuário em ambas as direções (decapsulação de uplink, encapsulação GTP-U de downlink) para pacotes internos IPv6.



Pré-requisitos do Kernel

IPv6 deve estar habilitado e o encaminhamento ativado no nível do host. Aplique via um drop-in do sysctl (por exemplo, sob `/etc/sysctl.d/`).

Configuração	Valor necessário	Descrição
`net.ipv6.conf.all.disable_ipv6`	`0`	IPv6 deve estar habilitado. A interface do plano de dados do UPF precisa de um endereço IPv6 global para trânsito e roteamento.
`net.ipv6.conf.default.disable_ipv6`	`0`	Garante que o IPv6 permaneça habilitado em interfaces que aparecem recentemente.
`net.ipv6.conf.all.forwarding`	`1`	Habilita o encaminhamento IPv6 para que o tráfego de uplink do UE saia e o tráfego de downlink seja roteado para o plano de dados.

A interface do plano de dados também precisa de um endereço IPv6 global no segmento de trânsito (em direção ao roteador de borda) para que o OSPFv3 possa formar uma adjacência e trocar rotas.

Pool de Endereços IPv6 do UE

Quando o UPF aloca endereços de UE (em vez de recebê-los do SMF), defina um pool IPv6 em `runtime.exs`. Quando o SMF/PGW-C fornece o endereço do UE via PFCP, esse endereço é usado diretamente e este pool não é consultado para a sessão.

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

Parâmetro	Tipo	Necessário	Padrão	Descrição
`ueip_pool`	String (CIDR)	Não	-	Pool de IP do UE usado quando o UPF realiza a alocação de endereços. Aceita um CIDR IPv4 ou IPv6; a família de endereços é detectada a partir da string.

Publicidade de Rotas OSPFv3

Cada endereço IPv6 ativo do UE é instalado como uma rota de host `/128` e publicada para o roteador de borda, espelhando o comportamento do IPv4 `/32`. As rotas são gerenciadas pelo gerenciador de rotas e redistribuídas no OSPF pelo FRR.

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

Parâmetro	Tipo	Necessário	Padrão	Descriçã
`route_manager_enabled`	Booleano	Não	`false`	Habilita o gerenciame de rotas de host do UE. Deve ser `tr` para publica rotas do UE.
`route_manager_type`	String	Não	-	Backend de rotas. `"frr"` injeta rotas FRR/`vtysh`; backend estático usa diretamente os comando de rota do kernel `ip`/`ipv6`. Us `"frr"` para publicidade OSPFv3.

A instância do FRR deve executar **OSPFv3** (`ospf6d`) na interface de trânsito e redistribuir rotas estáticas, para que as rotas de host `/128` do UE sejam inundadas para o roteador de borda. As rotas de host IPv6 são adicionadas com o comando `ipv6 route <addr>/128`

- o comando IPv4 `ip route` não aceita prefixos IPv6. Veja Gerenciamento de Rotas para o ciclo de vida do gerenciador de rotas.

Comportamento do Plano de Dados

Direção	Comportamento
Downlink (rede → UE)	O destino interno IPv6 é correspondido com a tabela de downlink IPv6 programada a partir da sessão PFCP; pacotes correspondentes são encapsulados em GTP-U em direção ao gNB/eNB (ou armazenados em buffer se a ação da regra for BUFF).
Uplink (UE → rede)	GTP-U é decapsulado e o pacote interno IPv6 é encaminhado para fora da interface N6 usando a FIB IPv6 do kernel (daí o sysctl de encaminhamento).
Bufferização (BUFF)	Pacotes de downlink IPv6 armazenados em buffer são entregues ao ouvinte de buffer do espaço do usuário e esvaziados na transição para o encaminhamento, da mesma forma que o IPv4.

Nenhuma relaxação de `accept_local` / filtro de caminho reverso é necessária nas versões atuais; o caminho do buffer usa uma fonte externa não local, então os pacotes não são descartados como martianos.

Solução de Problemas

O tráfego de downlink IPv6 não chega ao UE

Sintomas: A sessão está estabelecida e o uplink funciona, mas o tráfego da rede para o UE é descartado.

Causas possíveis:

- A rota de host `/128` do UE não está sendo publicada (gerenciador de rotas desativado, ou OSPFv3 não está em execução no UPF).
- A adjacência OSPFv3 entre o UPF e o roteador de borda não está estabelecida.

- O roteador de borda ainda não instalou o `/128` (atraso de convergência).

Resolução:

1. Confirme `route_manager_enabled: true` e `route_manager_type: "frr"`.
2. Verifique se a instância do FRR do UPF executa OSPFv3 na interface de trânsito e redistribui rotas estáticas.
3. Confirme se a adjacência OSPFv3 está completa e o roteador de borda mostra o `/128` do UE aprendido via OSPF.

O tráfego de usuário IPv6 não é encaminhado de forma alguma

Sintomas: Nem o uplink nem o downlink IPv6 funcionam, mesmo com rotas presentes.

Causas possíveis:

- IPv6 está desabilitado no host, ou o encaminhamento está desligado.
- A interface do plano de dados não tem um endereço IPv6 global.

Resolução:

1. Verifique os pré-requisitos do kernel (`disable_ipv6=0`, `forwarding=1`).
2. Confirme se a interface do plano de dados tem um endereço IPv6 global no segmento de trânsito.

Referência de Métricas

Este documento descreve todas as métricas do Prometheus expostas pelo OmniUPF no endpoint `/api/metrics` (a API HTTP roda na `api_port`, padrão `8080`). Note o prefixo `/api` - cada rota da API é servida sob ele.

Os nomes, tipos e rótulos das métricas abaixo refletem as métricas que o OmniUPF expõe.

Categorias de Métricas

1. **Métricas PFCP** - contadores de mensagens de controle, associações, sessões e erros
2. **Métricas do ciclo de vida da sessão** - contadores de criação/exclusão/modificação de sessão e latência de estabelecimento
3. **Métricas de caminho lento e buffer de espaço do usuário** - contadores de pacotes para buffer de downlink, jardim murado e indicações de erro de TEID desconhecido
4. **Métricas de Relatório de Dados de Downlink (DLDR)** - notificações de solicitação de relatório de sessão PFCP e rastreamento de índice FAR
5. **Métricas GTP-U** - respostas de eco e contadores de indicações de erro
6. **Métricas URR** - contadores de relatórios de uso e temporizadores URR ativos
7. **Métricas de mapa eBPF** - medidores de entradas do mapa do plano de dados
8. **Throughput do plano de dados** - contadores `_total` por direção (N3/N6), controlados por `expose_xdp_metrics` (#57); também sob demanda via `/api/v1/xdp_stats`
9. **Métricas de recursos / roteamento** - medidores de pool de TEID/FAR-ID e rotas de UE
10. **Métricas de Jardim Murado** - contadores de interceptação de portal cativo, encaminhamento e spoofing de DNS

Métricas PFCP

Contadores de protocolo de plano de controle entre o UPF e seus pares PFCP (SMF / PGW-C / SGW-C).

Nome da Métrica	Tipo	Rótulos	De
`upf_pfcp_associations`	Gauge	none	Associações atualmente pares)
`upf_pfcp_sessions`	Gauge	none	Sessões PFC ativas (todos
`upf_pfcp_peer_restarts_total`	Counter	`peer`	Detecções d pares (muda timestamp d
`upf_pfcp_duplicate_requests_total`	Counter	none	Solicitações detectadas e
`upf_orphaned_sessions_deleted_total`	Counter	none	Sessões exc resultado de par
`upf_pfcp_messages_rx_total`	Counter	`type`, `peer`	Mensagens por tipo de r
`upf_pfcp_messages_tx_total`	Counter	`type`, `peer`	Mensagens por tipo de r
`upf_pfcp_errors_total`	Counter	`type`	Rejeições de PFCP, rotula da rejeição (`session_co` `mandatory_`

Métricas do ciclo de vida da sessão

Nome da Métrica	Tipo	Rótulos
`upf_sessions_created_total`	Counter	none
`upf_sessions_deleted_total`	Counter	none
`upf_sessions_modified_total`	Counter	none
`upf_session_establishment_duration_microseconds`	Histogram	none

Métricas de caminho lento e buffer de espaço do usuário

O plano de dados eBPF envia pacotes para o caminho lento do espaço do usuário (porta UDP `22152`, padrão) para três trabalhos distintos: **buffer de downlink** quando um UE está ocioso (mantido até que o UE seja chamado), **redirecionamento de jardim murado**, e respondendo **uplink de TEID desconhecido** G-PDUs com uma Indicação de Erro GTP-U. Apenas as métricas `buffered`/`flushed`/`current` abaixo descrevem o buffer propriamente dito; o

jardim murado tem suas próprias métricas `upf_walled_garden_*` e TEID desconhecido é contado como `upf_gtpu_error_indications_sent_total`.

Nome da Métrica	Tipo	Rótulos	De
`upf_userspace_punt_received_total`	Counter	none	Todos os pac para o camir espaço do u + jardim mu desconhecid
`upf_gtpu_oversize_punted_total`	Counter	none	Pacotes GTP oversized (ta encapsulado de saída) en espaço do u kernel fragm externo. Cre sustentado s assinante es tráfego de ta caminho len `gtpu_oversi`
`upf_buffer_packets_buffered_total`	Counter	none	Pacotes adic sucesso ao b
`upf_buffer_packets_flushed_total`	Counter	none	Pacotes des
`upf_buffer_packets_dropped_total`	Counter	`reason`	Pacotes des admitidos no
`upf_buffer_packets_current`	Gauge	none	Pacotes atua no buffer
`upf_buffer_flush_operations_total`	Counter	none	Operações d buffer bem-s descarte de

Nome da Métrica	Tipo	Rótulos	De
`upf_buffer_flush_packets_sent_total`	Counter	none	Pacotes env operações d
`upf_buffer_flush_errors_total`	Counter	`reason`	Falhas nas o descarte de

Valores de `reason` de `upf_buffer_packets_dropped_total` (de descartes de adição ao buffer e erros de ouvinte de buffer):

- `global_limit` - capacidade total do buffer (`buffer_max_total`) atingida
- `far_limit` - limite de buffer por FAR (`buffer_max_per_far`) atingido
- `expired` - TTL (`buffer_ttl_ms`) excedido antes do descarte
- `read_error` - erro ao ler do socket do buffer
- `too_small` - pacote muito pequeno para um cabeçalho GTP
- `invalid_gtp_type` - tipo de mensagem GTP não-G-PDU
- `not_buffering_far` - FAR não tem uma ação BUFF
- `error_indication_failed` - não foi possível enviar a Indicação de Erro GTP-U para um uplink de TEID desconhecido

> O uplink de TEID desconhecido **não** é um descarte de buffer: é respondido com uma Indicação de Erro GTP-U e contado por `upf_gtpu_error_indications_sent_total` (uma resposta correta do caminho lento, não perda de pacote).

Valores de `reason` de `upf_buffer_flush_errors_total`:

- `far_lookup_failed` - não foi possível procurar informações do FAR no mapa eBPF
- `no_forw_action` - FAR não tem uma ação FORW definida
- `connection_failed` - falha ao abrir um socket UDP para descarte

Métricas de Relatório de Dados de Downlink (DLDR)

Métricas para notificações de Solicitação de Relatório de Sessão PFCP enviadas ao plano de controle quando pacotes de downlink são armazenados em buffer. Essas notificações acionam o SMF/PGW-C para chamar o UE.

Nome da Métrica	Tipo	Rótulos	Descrição
`upf_dldr_sent_total`	Counter	`peer`	Relatórios de Dados de Downlink enviados, rotulados pelo endereço SMF/PGW-C
`upf_dldr_errors_total`	Counter	none	Erros de codificação / envio de DLDR
`upf_dldr_skipped_total`	Counter	none	DLDRs ignorados porque o FAR já foi notificado
`upf_far_index_size`	Gauge	none	FARs registrados no FarIndex para notificação DLDR

Métricas GTP-U

Contadores de gerenciamento de caminho GTP-U e Indicação de Erro. Indicações de Erro são trocadas quando um par recebe pacotes para um TEID desconhecido (normalmente após um reinício).

Nome da Métrica	Tipo	Rótulos	D
`upf_gtpu_echo_responses_tx_total`	Counter	none	Res Ecc env
`upf_gtpu_error_indications_sent_total`	Counter	none	Ind Err env TEI ent des
`upf_gtpu_error_indications_rx_total`	Counter	none	Ind Err rec par
`upf_error_indication_sessions_deleted_total`	Counter	none	Ses exc res um de rec

Quando as Indicações de Erro são enviadas: o UPF recebe um pacote GTP-U para um TEID que não existe (por exemplo, após um reinício, ou a sessão já foi excluída), e informa o remetente para parar.

Quando as Indicações de Erro são recebidas: um par a montante não reconhece um TEID que o UPF encaminhou; o UPF exclui a sessão afetada para parar de encaminhar para um túnel morto.

Métricas URR (Regra de Relatório de Uso)

Nome da Métrica	Tipo	Rótulos	Descrição
`upf_urr_reports_sent_total`	Counter	`trigger`	Relatórios de uso URR enviados, rotulados pelo gatilho de relatório
`upf_urr_report_errors_total`	Counter	none	Erros de codificação / envio de relatório URR
`upf_urr_active_timers`	Gauge	none	Temporizadores URR ativos (periódicos / limite de tempo / validade de cota)

`trigger` reflete o gatilho de relatório URR que foi acionado - por exemplo, `periodic` (PERIO), `volume_threshold` (VOLTH), `time_threshold` (TIMTH), `quota_validity` (QUVTI).

Contadores de bytes por URR não são exportados como métricas do Prometheus (para evitar alta cardinalidade). Leia estatísticas individuais de URR via a API REST em `/api/v1/urr_map`.

Métricas de mapa eBPF

Medidores para utilização do mapa do plano de dados, atualizados em cada coleta.

Nome da Métrica	Tipo	Rótulos	Descrição
`upf_ebpf_far_entries`	Gauge	none	Entradas FAR no mapa eBPF
`upf_ebpf_pdr_uplink_entries`	Gauge	none	Entradas PDR de uplink no mapa eBPF (PDRs com um TEID)
`upf_ebpf_pdr_downlink_entries`	Gauge	none	Entradas PDR de downlink no mapa eBPF (PDRs indexados por IP de UE)

`upf_ebpf_far_entries` espelha o pool de IDs de FAR alocados (`upf_far_ids_allocated`), uma vez que o far_map mantém uma entrada por ID de FAR alocado. Os medidores de PDR são agregados a partir do estado da sessão ativa em cada coleta (custo é O(total de sessões)).

Throughput do plano de dados (N3/N6)

Contadores de pacotes e bytes por direção para tráfego de plano de usuário (**N3** = lado GTP-U em direção ao gNB/eNB; **N6** = lado SGi em direção à rede de dados) vêm do mapa percpu `upf_ext_stat` eBPF.

Nome da Métrica	Tipo	Rótulos	Descrição
`upf_n3_rx_packets_total` / `upf_n3_tx_packets_total`	Counter	none	Pacotes N3 (GTP-U) entrando / saindo do caminho de dados XDP
`upf_n6_rx_packets_total` / `upf_n6_tx_packets_total`	Counter	none	Pacotes N6 (SGi) entrando / saindo do caminho de dados XDP
`upf_n3_rx_bytes_total` / `upf_n3_tx_bytes_total`	Counter	none	Bytes N3 (GTP-U) entrando / saindo
`upf_n6_rx_bytes_total` / `upf_n6_tx_bytes_total`	Counter	none	Bytes N6 (SGi) entrando / saindo
`upf_xdp_ttl_exceeded_total`	Counter	none	Pacotes descartados porque seu TTL IPv4 / Limite de Salto IPv6 era 1 ou menos quando o UPF estava prestes a encaminhá-los. Qualquer taxa sustentada indica um loop de encaminhamento

Controlado por `expose_xdp_metrics` (issue #57). Ler `upf_ext_stat` no caminho de requisição (por coleta do Prometheus) provou desencadear um **wedge** de TX-ring genérico-XDP `virtio_net`. Duas mitig ações se aplicam: a leitura é feita fora do caminho de requisição pelo `Sampler` em segundo plano (em um intervalo fixo, nunca em uma coleta), e é controlada por `config`

`:upf_ex, expose_xdp_metrics: true`. Quando desativada, esses contadores não são declarados ou exportados.

Estes são contadores cumulativos `_total` **Counters**. O mapa subjacente `upf_ext_stat` é fixado, então sobrevive a um reinício do omniupf, mas é zerado em uma nova implantação de eBPF `.o`; o coletor traduz seu valor absoluto em incrementos de contador monótonos para que `rate()` permaneça correto em ambos. Não leia isso como medidores. Use `rate()`.

JSON sob demanda: `GET /api/v1/xdp_stats` retorna os mesmos campos e toca no eBPF apenas quando você o chama explicitamente. Campos: `rx_n3` / `tx_n3` / `rx_n6` / `tx_n6` (pacotes) e `rx_n3_bytes` / `tx_n3_bytes` / `rx_n6_bytes` / `tx_n6_bytes`, além de `ttl_exceeded`.

TTL / Limite de Salto. O caminho de dados envelhece cada pacote que encaminha por um salto: o TTL IPv4 do pacote do usuário / Limite de Salto IPv6 na desencapsulação (uplink em direção ao N6) e na encapsulação (downlink em direção ao RAN), e o TTL do cabeçalho GTP-U externo em um relé de GTP-U para GTP-U (SGW-U). Um pacote que chega sem nada para gastar é descartado e contado em `ttl_exceeded`, então um loop de encaminhamento entre UPFs se extingue e aparece como uma taxa crescente de `upf_xdp_ttl_exceeded_total` em vez de como throughput descontrolado.

Métricas de recursos / roteamento

Nome da Métrica	Tipo	Rótulos	Descrição
`upf_routes_total`	Gauge	none	Rotas de IP de UE atualmente rastreadas pelo gerenciador de rotas
`upf_teid_allocated`	Gauge	none	TEIDs atualmente alocados do pool
`upf_far_ids_allocated`	Gauge	none	IDs de FAR atualmente alocados do pool

Métricas de Jardim Murado

Contadores para o jardim murado de crédito esgotado / portal cativo. Veja o Guia do Jardim Murado para comportamento.

Nome da Métrica	Tipo	Rótulos	
`upf_walled_garden_active_redirects`	Gauge	none	Ses red do mu
`upf_walled_garden_packets_intercepted_total`	Counter	none	Pac inte pel mu
`upf_walled_garden_packets_dropped_total`	Counter	none	Pac des jar
`upf_walled_garden_packets_forwarded_total`	Counter	`dst_ip`	Pac bra enc por
`upf_walled_garden_bytes_forwarded_total`	Counter	`dst_ip`	Byt bra enc por
`upf_walled_garden_dns_spoofed_total`	Counter	`domain`	Cor fals o p dor
`upf_walled_garden_dns_forwarded_total`	Counter	`domain`	Cor enc (do list dor

> **Nota de cardinalidade.** Os rótulos `dst_ip` e `domain` são ilimitados. Mantenha a lista branca restrita e agregue esses rótulos em regras de gravação se você coletar um jardim murado movimentado.

Usando Métricas do Prometheus

Acessando Métricas

As métricas são expostas no endpoint `/api/metrics` do servidor da API HTTP, que escuta na `api_port` (padrão `8080`). Note o prefixo `/api` - cada rota da API é servida sob ele:

```
# Ver métricas brutas
curl http://localhost:8080/api/metrics

# Exemplo de saída
upf_pfcp_sessions 42
upf_pfcp_associations 2
upf_dldr_sent_total{peer="10.100.50.241"} 17
```

Configuração do Prometheus

Adicione o alvo OmniUPF ao seu `prometheus.yml`:

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

Painéis do Grafana

Painéis úteis para construir a partir dessas métricas:

- Contagens de sessões e associações (`upf_pfcp_sessions`, `upf_pfcp_associations`)
- Throughput N3/N6 por direção está disponível sob demanda via `/api/v1/xdp_stats` (removido da coleta, veja #57)
- Pressão do buffer (`upf_buffer_packets_current`, taxa de descarte por `reason`)
- Atividade DLDR / chamada (`rate(upf_dldr_sent_total[5m])`)
- Taxas de Indicação de Erro e reinício de pares para alerta de saúde do túnel

Documentação Relacionada

- **Guia de Monitoramento** - Monitoramento de estatísticas, planejamento de capacidade e alerta
- **Guia de Configuração** - Configurar `api_port`, ajuste de buffer e outras opções do UPF
- **Guia da Interface Web** - Ver métricas na página de Estatísticas
- **Guia de Arquitetura** - Caminho de dados eBPF e otimização de desempenho
- **Guia de Gerenciamento de Regras** - Compreendendo métricas PDR, FAR, QER, URR
- **Guia do Jardim Murado** - Redirecionamento de portal cativo e comportamento da lista branca
- **Guia de Solução de Problemas** - Usando métricas para diagnósticos

Guia de Monitoramento

Índice

1. Visão Geral
2. Monitoramento de Estatísticas
3. Monitoramento de Capacidade
4. Métricas de Desempenho
5. Alertas e Limiares
6. Planejamento de Capacidade
7. Solução de Problemas de Desempenho

Visão Geral

O monitoramento eficaz do OmniUPF é crítico para manter a qualidade do serviço, prevenir a exaustão da capacidade e solucionar problemas de desempenho. O OmniUPF fornece métricas abrangentes em tempo real através de sua interface Web e API REST.

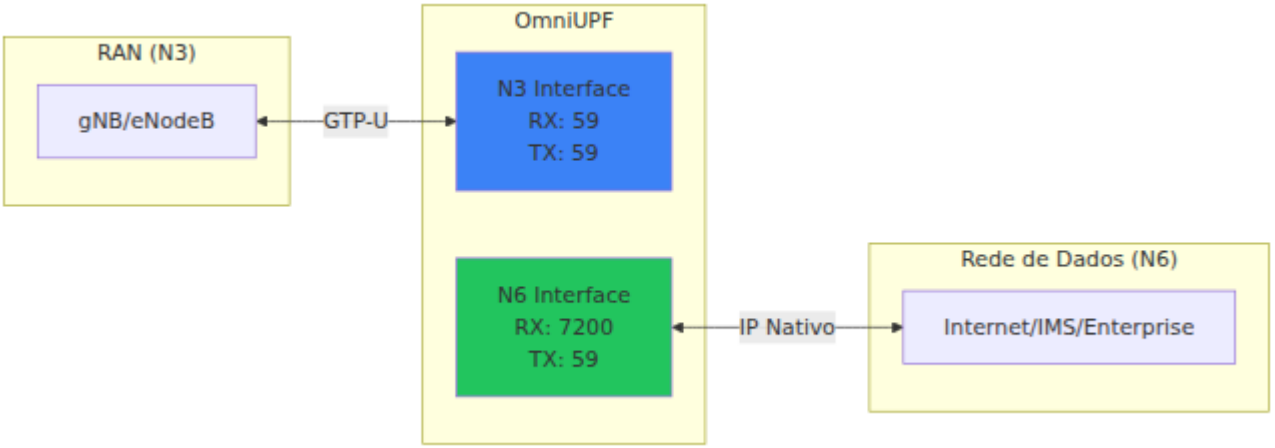
Categorias de Monitoramento

Categoria	Propósito	Frequência de Atualização	Métricas Chave
Estatísticas de Pacotes	Acompanhar taxas de processamento de pacotes e erros	Em tempo real	Pacotes RX/TX, descartes, divisão de protocolos
Estatísticas de Interface	Monitorar a distribuição de tráfego N3/N6	Em tempo real	N3 RX/TX, N6 RX/TX
Estatísticas de XDP	Acompanhar o desempenho do caminho de dados do kernel	Em tempo real	XDP processados, passados, descartados, abortados
Estatísticas de Roteamento	Monitorar decisões de roteamento de pacotes	Em tempo real	Olhadas FIB, acertos/miss de cache
Capacidade do Mapa eBPF	Prevenir a exaustão de recursos	A cada 10s	Porcentagens de uso do mapa, usado vs. capacidade
Estatísticas de Buffer	Acompanhar o buffer de pacotes durante a mobilidade	A cada 5s	Pacotes em buffer, idade do buffer, contagem de FAR

Monitoramento de Estatísticas

Estatísticas da Interface N3/N6

As estatísticas da interface N3/N6 fornecem visibilidade sobre a distribuição de tráfego entre a RAN (N3) e a Rede de Dados (N6).



Métricas:

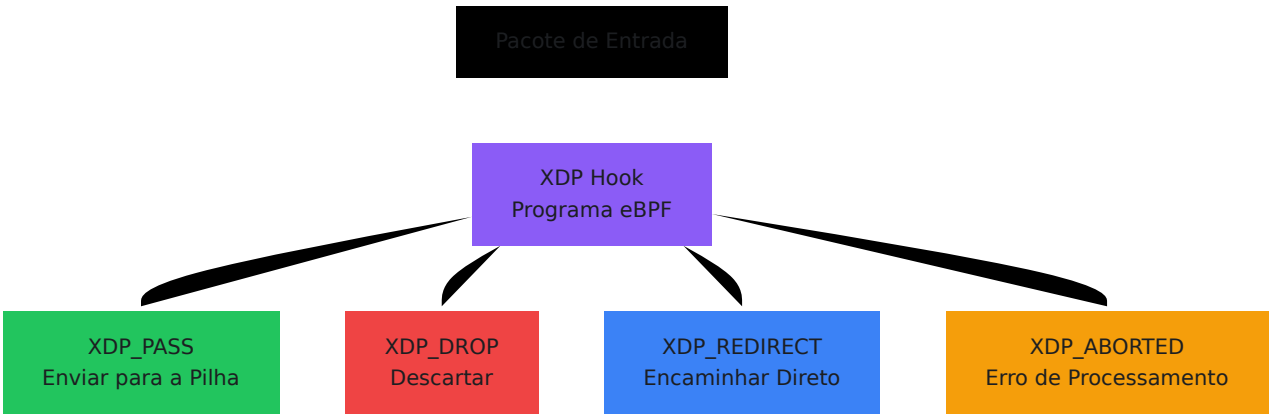
- **RX N3**: Pacotes recebidos da RAN (tráfego GTP-U de uplink)
- **TX N3**: Pacotes transmitidos para a RAN (tráfego GTP-U de downlink)
- **RX N6**: Pacotes recebidos da Rede de Dados (IP nativo de downlink)
- **TX N6**: Pacotes transmitidos para a Rede de Dados (IP nativo de uplink)
- **Total**: Contagem agregada de pacotes em todas as interfaces

Comportamento Esperado:

- **RX N3 ≈ TX N6**: Pacotes de uplink fluem da RAN para a Rede de Dados
- **RX N6 ≈ TX N3**: Pacotes de downlink fluem da Rede de Dados para a RAN
- Um desequilíbrio significativo pode indicar:
  - Tráfego assimétrico (downloads >> uploads)
  - Descartes de pacotes ou erros de encaminhamento
  - Configurações de roteamento incorretas

Estatísticas de XDP

As estatísticas de XDP (eXpress Data Path) mostram o desempenho do processamento de pacotes a nível de kernel.



Métricas:

- **Abortado**: O programa XDP encontrou um erro (deve ser sempre 0)
- **Descartar**: Pacotes intencionalmente descartados pelo programa XDP

- **Passar**: Pacotes passados para a pilha de rede para processamento adicional
- **Redirecionar**: Pacotes redirecionados diretamente para a interface de saída
- **TX**: Pacotes transmitidos via XDP

Interpretação:

- **Abortado > 0**: Problema crítico com o programa eBPF ou compatibilidade do kernel
- **Descartar > 0**: Descates baseados em políticas ou pacotes inválidos
- **Passar alto**: A maioria dos pacotes processados na pilha de rede (normal)
- **Redirecionar alto**: Pacotes encaminhados diretamente (desempenho ideal)

Estatísticas de Pacotes

Detalhamento do protocolo de pacotes e contadores de processamento.

Contadores de Protocolo:

- **RX ARP**: Pacotes do Protocolo de Resolução de Endereços
- **RX GTP ECHO**: GTP-U Echo Request/Response (keepalive)
- **RX GTP OUTRO**: Outras mensagens de controle GTP
- **RX GTP PDU**: Dados de usuário encapsulados em GTP-U (tráfego principal)
- **RX GTP INEXP**: Tipos de pacotes GTP inesperados
- **RX ICMP**: Protocolo de Mensagem de Controle da Internet (ping, erros)
- **RX ICMP6**: Pacotes ICMPv6
- **RX IP4**: Pacotes IPv4
- **RX IP6**: Pacotes IPv6
- **RX OUTRO**: Outros protocolos
- **RX TCP**: Pacotes do Protocolo de Controle de Transmissão
- **RX UDP**: Pacotes do Protocolo de Datagrama do Usuário

Casos de Uso:

- **Monitorar contagem de GTP-U PDU**: Indicador primário de tráfego de usuário
- **Verificar ICMP para conectividade**: Teste de acessibilidade da rede
- **Acompanhar a proporção TCP vs UDP**: Padrões de tráfego de aplicação
- **Detectar protocolos inesperados**: Questões de segurança ou configuração incorreta

---

Estatísticas de Roteamento

Estatísticas de olhadas FIB (Forwarding Information Base) para decisões de roteamento.

Olhar FIB IPv4:

- **Cache**: Olhadas de rota em cache (caminho rápido)
- **OK**: Olhadas de rota bem-sucedidas

Olhar FIB IPv6:

- **Cache**: Olhadas de rota IPv6 em cache
- **OK**: Olhadas de rota IPv6 bem-sucedidas

Indicadores de Desempenho:

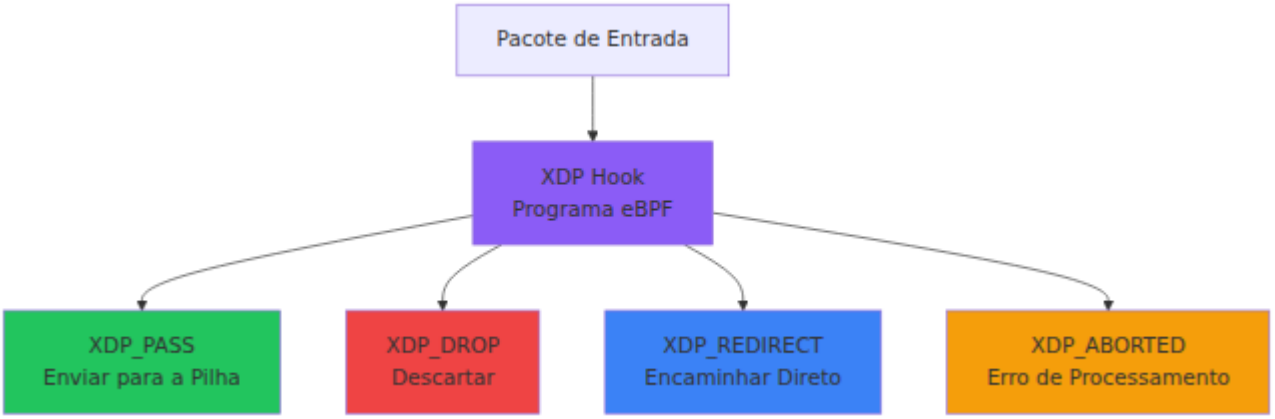
- **Alta Taxa de Acerto de Cache**: Indica bom desempenho do cache de roteamento
- **Alta Contagem de OK**: Confirma que as tabelas de roteamento estão corretamente configuradas
- **Olhadas Baixas ou Zero**: Pode indicar tráfego não fluindo ou desvio de roteamento

---

Monitoramento de Capacidade

Capacidade do Mapa eBPF

O monitoramento da capacidade do mapa eBPF previne falhas na estabelecimento de sessões devido à exaustão de recursos.



Mapas eBPF Críticos

far_map (Regras de Ação de Encaminhamento):

- **Capacidade**: 131,070 entradas
- **Tamanho da Chave**: 4 B (ID FAR)
- **Tamanho do Valor**: 16 B (parâmetros de encaminhamento)
- **Uso de Memória**: ~2.6 MB
- **Criticidade**: Alta - Usado para todas as decisões de encaminhamento de pacotes

pdr_map_downlin (PDRs de Downlink - IPv4):

- **Capacidade**: 131,070 entradas
- **Tamanho da Chave**: 4 B (endereço IPv4 do UE)
- **Tamanho do Valor**: 208 B (informações do PDR)
- **Uso de Memória**: ~27 MB
- **Criticidade**: Crítica - Falha na estabelecimento de sessão se cheio

pdr_map_downlin_ip6 (PDRs de Downlink - IPv6):

- **Capacidade**: 131,070 entradas
- **Tamanho da Chave**: 16 B (endereço IPv6 do UE)
- **Tamanho do Valor**: 208 B (informações do PDR)
- **Uso de Memória**: ~29 MB
- **Criticidade**: Crítica - Falha na estabelecimento de sessão IPv6 se cheia

pdr_map_teid_ip (PDRs de Uplink):

- **Capacidade**: 131,070 entradas
- **Tamanho da Chave**: 4 B (TEID)
- **Tamanho do Valor**: 208 B (informações do PDR)
- **Uso de Memória**: ~27 MB
- **Criticidade**: Crítica - Tráfego de uplink falha se cheio

qer_map (Regras de Aplicação de QoS):

- **Capacidade**: 65,535 entradas
- **Tamanho da Chave**: 4 B (ID QER)
- **Tamanho do Valor**: 32 B (parâmetros de QoS)
- **Uso de Memória**: ~2.3 MB
- **Criticidade**: Média - Aplicação de QoS apenas

urr_map (Regras de Relatório de Uso):

- **Capacidade**: 131,070 entradas
- **Tamanho da Chave**: 4 B (ID URR)
- **Tamanho do Valor**: 16 B (contadores de volume)
- **Uso de Memória**: ~2.6 MB
- **Criticidade**: Baixa - Afeta apenas a cobrança

Limiares de Capacidade

Limite	Ação Necessária
0-50% (Verde)	Operação normal - Nenhuma ação necessária
50-70% (Amarelo)	Cuidado - Monitorar tendências de crescimento, planejar aumento de capacidade
70-90% (Âmbar)	Aviso - Agendar aumento de capacidade dentro de 1 semana
90-100% (Vermelho)	Crítico - Ação imediata necessária, novas sessões falharão

Procedimento de Aumento de Capacidade

Antes de aumentar a capacidade:

1. Revisar tendências de uso atuais
2. Estimar a taxa de crescimento futura
3. Calcular a capacidade necessária

Passos para aumentar a capacidade do mapa:

1. Parar o serviço OmniUPF
2. Atualizar o arquivo de configuração do UPF com os novos tamanhos de mapa
3. Reiniciar o serviço OmniUPF
4. Verificar nova capacidade na visualização de Capacidade
5. Monitorar para estabelecimento de sessão bem-sucedido

Nota: Alterar a capacidade do mapa eBPF requer reinício do UPF e limpa todas as sessões existentes.

Métricas de Desempenho

Para informações detalhadas sobre todas as métricas Prometheus expostas pelo OmniUPF, consulte a **Referência de Métricas**.

Taxa de Processamento de Pacotes

Cálculo:

```
Taxa de Pacotes (pps) = (Delta de Contagem de Pacotes) / (Delta de Tempo em segundos)
```

Exemplo:

- Pacotes RX iniciais: 7,000

- Após 10 segundos: 17,000
- Taxa de Pacotes = (17,000 - 7,000) / 10 = 1,000 pps

Metas de Desempenho:

- **UPF Pequeno**: 10,000 - 100,000 pps
- **UPF Médio**: 100,000 - 1,000,000 pps
- **UPF Grande**: 1,000,000 - 10,000,000 pps

Indicadores de Gargalo:

- Contagem de XDP abortada aumentando
- Alta utilização da CPU
- Aumento de descartes de pacotes
- Aumento de latência

Cálculo de Throughput

Cálculo:

```
Throughput (Mbps) = (Delta de Contagem de Bytes × 8) / (Delta de
Tempo em segundos × 1,000,000)
```

Exemplo:

- Bytes RX iniciais: 500 MB
- Após 60 segundos: 800 MB
- Throughput = (300 MB × 8) / (60 × 1,000,000) = 40 Mbps

Throughput por direção é exportado como contadores `_total` (`upf_n3_rx_packets_total`, `upf_n3_rx_bytes_total`, ... equivalentes N6) quando `expose_xdp_metrics` está habilitado. N3 é o lado GTP-U (gNB), N6 é o lado SGi (rede de dados). A leitura do eBPF `upf_ext_stat` é feita pelo caminho de coleta pelo `Sampler` em segundo plano (a leitura por coleta foi um gatilho comprovado de `virtio_net` TX-ring-wedge, problema #57), e está bloqueada atrás da flag. Os mesmos campos também estão disponíveis sob demanda

como JSON em `/api/v1/xdp_stats`. Consulte a Referência de Métricas para detalhes.

Planejamento de Capacidade:

- Monitorar horários de pico de throughput (por exemplo, horas da noite)
- Comparar com a capacidade do link (velocidades das interfaces N3/N6)
- Planejar para 2x throughput de pico para margem de segurança

---

Taxa de Descarte

Cálculo:

```
Taxa de Descarte (%) = (Pacotes Descartados / Total de Pacotes RX)
× 100
```

Limiares Aceitáveis:

- **< 0.1%**: Excelente (perda de pacotes normal devido a erros)
- **0.1% - 1%**: Bom (problemas menores ou limitação de taxa)
- **1% - 5%**: Ruim (investigar problemas de QoS ou capacidade)
- **> 5%**: Crítico (grande problema de encaminhamento ou capacidade)

Causas Comuns de Descarte:

- Limitação de taxa de QER (MBR excedido)
- Falhas de busca no mapa eBPF
- TEIDs ou IPs de UE inválidos
- Erros de roteamento

Alertas e Limiares

Alertas Recomendados

Alertas Críticos (Resposta imediata necessária):

- Capacidade do mapa eBPF > 90%
- Contagem de abortos de XDP > 0
- Taxa de descarte > 5%
- Verificação de saúde do UPF falhou

Alertas de Aviso (Resposta dentro de 1 hora):

- Capacidade do mapa eBPF > 70%
- Taxa de descarte > 1%
- Taxa de pacotes se aproximando da capacidade do link
- TTL do buffer excedido (pacotes com mais de 30s)

Alertas Informativos (Monitorar tendências):

- Capacidade do mapa eBPF > 50%
- Contagem de pacotes em buffer aumentando
- Novas associações PFCP estabelecidas/liberadas
- Limiares de volume URR excedidos

Configuração de Alertas

Os alertas podem ser configurados via:

1. **Métricas Prometheus**: Exportar métricas para monitoramento externo (consulte Referência de Métricas para a lista completa)
2. **Monitoramento de Logs**: Analisar logs do OmniUPF para padrões de erro
3. **Polling da API REST**: Consultar periodicamente os endpoints `/map_info`, `/packet_stats`
4. **Monitoramento da Interface Web**: Monitoramento manual através das páginas de Estatísticas e Capacidade

Planejamento de Capacidade

Estimativa de Capacidade de Sessão

Calcular sessões máximas:

```
Máx Sessões = min(
  Capacidade do Mapa PDR / 2,  # PDRs de Downlink + Uplink por
sessão
  Capacidade do Mapa FAR / 2,  # FARs de Downlink + Uplink por
sessão
  Capacidade do Mapa QER       # Opcional, um QER por sessão
)
```

Exemplo:

- Capacidade do Mapa PDR: 131,070
- Capacidade do Mapa FAR: 131,070
- Capacidade do Mapa QER: 65,535

Máx Sessões = min(131,070 / 2, 131,070 / 2, 65,535) = **65,535 sessões**

Capacidade de Memória

Calcular a memória total do mapa eBPF:

```
Memória = Σ (Capacidade do Mapa × (Tamanho da Chave + Tamanho do
Valor))
```

Exemplo de Configuração:

- Mapas PDR: 3 × 131,070 × 212 B = 83.3 MB
- Mapa FAR: 131,070 × 20 B = 2.6 MB
- Mapa QER: 65,535 × 36 B = 2.3 MB
- Mapa URR: 131,070 × 20 B = 2.6 MB

- **Total**: ~91 MB de memória do kernel

Considerações de Memória do Kernel:

- Garantir limite de memória bloqueada suficiente (`ulimit -l`)
- Reservar 2x do uso estimado para margem de segurança
- Monitorar a disponibilidade de memória do kernel

Capacidade de Tráfego

Calcular a capacidade de throughput necessária:

1. **Estimar o throughput médio da sessão**:
    - Streaming de vídeo: ~5 Mbps
    - Navegação na web: ~1 Mbps
    - VoIP: ~0.1 Mbps
2. **Calcular o throughput agregado**:

    ```
    Throughput Total = Sessões × Throughput Médio da Sessão
    ```

3. **Adicionar margem**:

    ```
    Capacidade Necessária = Throughput Total × 2  # 100% de margem
    ```

Exemplo:

- 10,000 sessões simultâneas
- Média de 2 Mbps por sessão
- Total: 20 Gbps
- Capacidade necessária: 40 Gbps (interfaces N3 + N6)

Planejamento de Crescimento

Análise de Tendências:

1. Registrar a contagem de sessões máximas diárias
2. Calcular a taxa de crescimento semanal
3. Extrapolar para o limite de capacidade

Fórmula da Taxa de Crescimento:

```
Semanas até a Capacidade = (Capacidade - Uso Atual) / (Crescimento Semanal)
```

Exemplo:

- Sessões atuais: 30,000
- Capacidade: 65,535 sessões
- Crescimento semanal: 2,000 sessões
- Semanas até a capacidade: (65,535 - 30,000) / 2,000 = **17.8 semanas**

Ação: Planejar atualização de capacidade em 12 semanas (deixando 5 semanas de buffer).

Solução de Problemas de Desempenho

Alta Taxa de Descarte de Pacotes

Sintomas: Taxa de descarte > 1%, reclamações de usuários sobre conectividade ruim

Diagnóstico:

1. Verificar Estatísticas → Estatísticas de Pacotes
2. Identificar se os descartes são específicos de protocolo
3. Revisar Estatísticas de XDP para descartes vs. abortos de XDP

Causas Comuns:

- **Limitação de Taxa de QER**: Verificar valores de MBR de QER vs. tráfego real
- **TEIDs Inválidos**: Verificar se o TEID do PDR de uplink corresponde à atribuição do gNB
- **IPs de UE Desconhecidos**: Verificar se o PDR de downlink existe para o IP do UE
- **Overflow de Buffer**: Verificar estatísticas de buffer

Resolução:

- Aumentar o MBR do QER se houver limitação de taxa
- Verificar se o SMF criou os PDRs corretos
- Limpar buffers se overflow detectado

Erros de Processamento de XDP

Sintomas: XDP abortado > 0

Diagnóstico:

1. Navegar até Estatísticas → Estatísticas de XDP
2. Verificar contador de abortos
3. Revisar logs do OmniUPF para erros de eBPF

Causas Comuns:

- Falha na verificação do programa eBPF
- Incompatibilidade da versão do kernel
- Erros de acesso ao mapa eBPF
- Corrupção de memória

Resolução:

- Reiniciar o serviço OmniUPF
- Verificar se a versão do kernel atende aos requisitos mínimos (Linux 5.4+)
- Revisar logs do programa eBPF

- Contatar suporte se o problema persistir

Exaustão de Capacidade

Sintomas: Falhas na estabelecimento de sessões, capacidade do mapa em 100%

Diagnóstico:

1. Navegar até a página de Capacidade
2. Identificar qual mapa está em 100%
3. Verificar se as sessões estão presas (não sendo excluídas)

Mitigação Imediata:

1. Identificar sessões antigas (verificar página de Sessões)
2. Solicitar ao SMF que exclua sessões antigas
3. Limpar buffers para liberar entradas de FAR

Resolução a Longo Prazo:

1. Aumentar a capacidade do mapa eBPF
2. Agendar reinício do UPF com mapas maiores
3. Implementar políticas de limpeza de sessões

Degradação de Desempenho

Sintomas: Alta latência, baixo throughput, saturação da CPU

Diagnóstico:

1. Verificar taxa de pacotes vs. linha de base histórica
2. Revisar estatísticas de XDP para atrasos de processamento
3. Monitorar a utilização da CPU no host do UPF
4. Verificar a utilização das interfaces N3/N6

Causas Comuns:

- Tráfego excedendo a capacidade do UPF
- Núcleos de CPU insuficientes para processamento de pacotes
- Gargalo na interface de rede
- Colisões de hash no mapa eBPF

Resolução:

- Escalar o UPF horizontalmente (adicionar mais instâncias)
- Atualizar a CPU ou habilitar RSS (Receive Side Scaling)
- Atualizar interfaces de rede para maior velocidade
- Ajustar a função de hash do mapa eBPF

Documentação Relacionada

- **Referência de Métricas** - Referência completa de métricas Prometheus
- **Guia de Operações do UPF** - Arquitetura e operações gerais do UPF
- **Guia de Gerenciamento de Regras** - Configuração de PDR, FAR, QER, URR
- **Guia de Operações da Interface Web** - Recursos de monitoramento do painel de controle
- **Guia de Solução de Problemas** - Problemas comuns e diagnósticos
- **Guia de Arquitetura** - Caminho de dados eBPF e otimização de desempenho

Referência de Códigos de Causa PFCP

Visão Geral

PFCP (Protocolo de Controle de Encaminhamento de Pacotes) utiliza códigos de causa em mensagens de resposta para indicar o resultado das solicitações. Este documento descreve os códigos de causa implementados no OmniUPF e quando eles ocorrem durante o processamento de mensagens PFCP.

Todos os códigos de causa estão em conformidade com as especificações **3GPP TS 129.244** e são retornados em mensagens de resposta PFCP para indicar sucesso, falha ou condições de erro específicas.

Monitoramento de Códigos de Causa

OmniUPF rastreia os resultados das mensagens PFCP usando métricas do Prometheus. Cada resposta PFCP inclui um código de causa que é registrado em:

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

Isso permite o monitoramento de:

- **Taxas de sucesso** por tipo de mensagem e nó do plano de controle
- **Padrões de erro** indicando configurações incorretas ou problemas de protocolo
- **Saúde da associação** com base nas taxas de rejeição

Veja a Referência de Métricas para a documentação completa das métricas PFCP.

Categorias de Códigos de Causa

Códigos de Sucesso

Código	Nome	Quando Ocorre
1	**RequestAccepted**	Solicitação processada com sucesso. Todos os IEs obrigatórios presentes e válidos. Regras criadas/modificadas/excluídas com sucesso.

Códigos de Erro do Cliente

Código	Nome	Quando Ocorre
64	**RequestRejected**	Rejeição geral por erros não especificados. Usado quando nenhum código de causa específico se aplica.
65	**SessionContextNotFound**	Modificação ou exclusão de sessão solicitada para SEID desconhecido. A sessão especificada não existe neste UPF.
66	**MandatoryIEMissing**	Elemento de Informação obrigatório ausente. Exemplos: NodeID ausente na Configuração de Associação, F-SEID ausente na Estabelecimento de Sessão, RecoveryTimeStamp ausente.
67	**ConditionalIEMissing**	IE condicionalmente obrigatório ausente com base em outros IEs presentes. Usado quando IEs dependem da presença uns dos outros.
69	**MandatoryIEIncorrect**	IE obrigatório presente, mas contém dados inválidos. Exemplos: Formato de NodeID não

Código	Nome	Quando Ocorre
		analisável, valor de RecoveryTimeStamp inválido, F-SEID malformado.
72	**NoEstablishedPFCPAssociation**	Operação de sessão tentada sem associação ativa. Deve estabelecer associação PFCP antes de criar sessões.
73	**RuleCreationModificationFailure**	Erro ao aplicar regras PDR, FAR, QER ou URR ao caminho de dados eBPF. Causas possíveis: capacidade do mapa eBPF esgotada, parâmetros de regra inválidos, falha na alocação de recursos.

Códigos de Erro do Servidor/Recurso

Código	Nome	Quando Ocorre
74	**PFCPEntityInCongestion**	UPF enfrentando alta carga ou exaustão de recursos. Temporariamente incapaz de processar solicitações.
75	**NoResourcesAvailable**	Recursos insuficientes para atender à solicitação. Exemplos: capacidade do mapa eBPF esgotada, falha na alocação de memória, pool de TEID esgotado.
77	**SystemFailure**	Erro interno crítico impedindo o processamento da solicitação. Exemplos: falha no programa eBPF, erro na interface do kernel, corrupção de banco de dados.

Códigos de Funcionalidade Não Suportada

Código	Nome	Quando Ocorre
68	**InvalidLength**	O campo de comprimento do IE não corresponde ao comprimento real dos dados. Atualmente não utilizado no OmniUPF.
70	**InvalidForwardingPolicy**	Política de encaminhamento não suportada pelo UPF. Atualmente não utilizado no OmniUPF.
71	**InvalidFTEIDAllocationOption**	Opção de alocação de F-TEID não suportada. Atualmente não utilizado no OmniUPF.
76	**ServiceNotSupported**	Serviço ou funcionalidade solicitada não implementada. Atualmente não utilizado no OmniUPF.
78	**RedirectionRequested**	UPF solicita redirecionamento para outra instância do UPF. Atualmente não utilizado no OmniUPF.

Cenários e Causas Comuns

Falhas na Configuração da Associação

Cenário: NodeID Ausente

```
SMF → UPF: Associação Configuração Solicitação (sem NodeID)
UPF → SMF: Associação Configuração Resposta (Causa:
MandatoryIEMissing)
```

Resolução: Garantir que o SMF inclua o IE NodeID em todas as Solicitações de Configuração de Associação.

Cenário: Formato de NodeID Inválido

```
SMF → UPF: Associação Configuração Solicitação (NodeID="inválido")
UPF → SMF: Associação Configuração Resposta (Causa:
MandatoryIEIncorrect)
```

Resolução: NodeID deve ser um FQDN válido ou endereço IPv4/IPv6.

Cenário: Timestamp de Recuperação Ausente

```
SMF → UPF: Associação Configuração Solicitação (sem
RecoveryTimeStamp)
UPF → SMF: Associação Configuração Resposta (Causa:
MandatoryIEMissing)
```

Resolução: Incluir RecoveryTimeStamp na Solicitação de Configuração de Associação.

Falhas na Estabelecimento de Sessão

Cenário: Nenhuma Associação Estabelecida

```
SMF → UPF: Estabelecimento de Sessão Solicitação
UPF → SMF: Estabelecimento de Sessão Resposta (Causa:
NoEstablishedPFCPAssociation)
```

Resolução: Estabelecer associação PFCP antes de criar sessões.

Cenário: Falha na Criação de Regra

```
SMF → UPF: Estabelecimento de Sessão Solicitação
UPF processa FARs, QERs, URRs com sucesso
UPF falha ao criar PDR (mapa eBPF cheio)
UPF → SMF: Estabelecimento de Sessão Resposta (Causa:
RuleCreationModificationFailure)
```

Resolução:

- Verifique a capacidade do mapa eBPF (veja Monitoramento de Capacidade)
- Aumente os tamanhos dos mapas na configuração do UPF
- Reduza a contagem de sessões ativas

Cenário: F-SEID Ausente

```
SMF → UPF: Estabelecimento de Sessão Solicitação (sem CP F-SEID)
UPF → SMF: Estabelecimento de Sessão Resposta (Causa:
MandatoryIEMissing)
```

Resolução: Incluir CP F-SEID na Solicitação de Estabelecimento de Sessão.

Falhas na Modificação de Sessão

Cenário: SEID Desconhecido

```
SMF → UPF: Modificação de Sessão Solicitação (SEID=12345)
UPF não possui sessão com SEID 12345
UPF → SMF: Modificação de Sessão Resposta (Causa:
SessionContextNotFound)
```

Resolução:

- Verifique se o SEID corresponde ao valor da Resposta de Estabelecimento de Sessão
- Verifique se a sessão já foi excluída
- Certifique-se de usar a instância correta do UPF (cenários de loopback N9)

Falhas na Exclusão de Sessão

Cenário: SEID Desconhecido

```
SMF → UPF: Exclusão de Sessão Solicitação (SEID=67890)
UPF não possui sessão com SEID 67890
UPF → SMF: Exclusão de Sessão Resposta (Causa:
SessionContextNotFound)
```

Resolução: O SEID pode já ter sido excluído ou nunca existiu.

Solução de Problemas com Códigos de Causa

Usando Métricas do Prometheus

Consultas Prometheus para identificar padrões de erro:

```
# Taxa de erro por código de causa
rate(upf_pfcp_rx_errors{cause_code!="RequestAccepted"}[5m])

# Principais causas de rejeição
topk(5, sum by (cause_code) (upf_pfcp_rx_errors))

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

# Falhas na estabelecimento de sessão
upf_pfcp_rx_errors{message_name="SessionEstablishmentRequest",
cause_code!="RequestAccepted"}
```

Usando a Interface Web

Navegue até a página **Sessões** para visualizar:

- Contagem de sessões ativas por nó do plano de controle
- Taxas de sucesso/falha no estabelecimento de sessão
- Erros recentes de sessão

Navegue até a página **Capacidade** para diagnosticar:

- Utilização do mapa eBPF (causa raiz de RuleCreationModificationFailure)
- Indicadores de exaustão de recursos

Veja o Guia da Interface Web para instruções detalhadas de monitoramento.

Passos Comuns de Depuração

Alta Taxa de MandatoryIEMissing:

1. Verifique a configuração do SMF para IEs obrigatórios
2. Verifique a compatibilidade da versão da biblioteca PFCP
3. Revise os logs do SMF para erros de construção de IE

Falhas Frequentes de RuleCreationModificationFailure:

1. Verifique a capacidade do mapa eBPF: `GET /api/v1/map_info`
2. Monitore o uso do mapa: `upf_ebpf_map_used / upf_ebpf_map_capacity`
3. Aumente os tamanhos dos mapas na configuração se > 70% utilizado
4. Veja Planejamento de Capacidade

Erros NoEstablishedPFCPAssociation:

1. Verifique se a associação existe: `GET /api/v1/pfcp_associations`
2. Verifique a configuração do tempo limite de heartbeat
3. Revise os logs de configuração da associação
4. Certifique-se de que SMF e UPF possam se alcançar

SessionContextNotFound na Modificação:

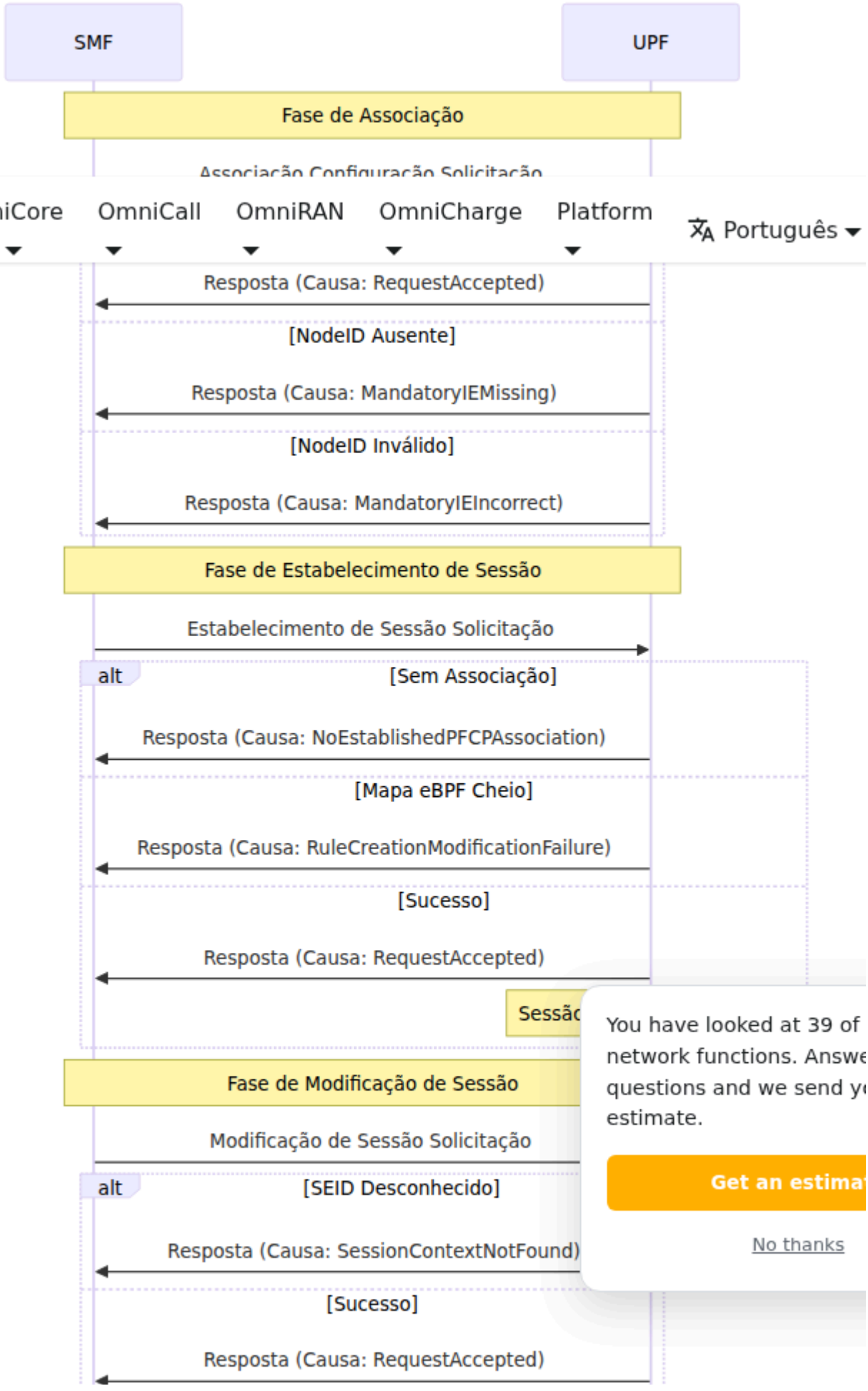
1. Verifique o SEID da resposta de estabelecimento de sessão
2. Verifique se a sessão foi excluída
3. Para loopback N9: Certifique-se de usar o endpoint correto do UPF

4. Consulte sessões ativas: `GET /api/v1/pfcp_sessions`

Impacto do Código de Causa nas

Operações

Ciclo de Vida da Sessão


SMF
UPF
Fase de Associação
Associação Configuração Solicitação
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Resposta (Causa: RequestAccepted)
[NodeID Ausente]
Resposta (Causa: MandatoryIEMissing)
[NodeID Inválido]
Resposta (Causa: MandatoryIEIncorrect)
Fase de Estabelecimento de Sessão
Estabelecimento de Sessão Solicitação
alt
[Sem Associação]
Resposta (Causa: NoEstablishedPFCPAssociation)
[Mapa eBPF Cheio]
Resposta (Causa: RuleCreationModificationFailure)
[Sucesso]
Resposta (Causa: RequestAccepted)
Sessão
Fase de Modificação de Sessão
Modificação de Sessão Solicitação
alt
[SEID Desconhecido]
Resposta (Causa: SessionContextNotFound)
[Sucesso]
Resposta (Causa: RequestAccepted)
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Métricas e Alertas

Alertas Recomendados:

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

# Aviso: Problemas de capacidade
- alert: PfcpRuleCreationFailures
  expr: |

rate(upf_pfcp_rx_errors{cause_code="RuleCreationModificationFailure"}
[5m]) > 0
  annotations:
    summary: "Falhas na criação de regras PFCP detectadas"

# Aviso: Problemas de associação
- alert: PfcpNoAssociation
  expr: |

rate(upf_pfcp_rx_errors{cause_code="NoEstablishedPFCPAssociation"}
[5m]) > 0
  annotations:
    summary: "Sessões PFCP tentadas sem associação"
```

Conformidade com Padrões 3GPP

OmniUPF implementa códigos de causa de acordo com:

- **3GPP TS 129.244 v16.4.0** - especificação PFCP
- **Seção 8.2.1** - definição de IE de Causa

- **Seção 8.19** - tabela de valores de Causa

Documentação Relacionada

- **Integração do Protocolo PFCP** - arquitetura PFCP e tratamento de mensagens
- **Referência de Métricas** - documentação da métrica upf_pfcp_rx_errors
- **Guia de Monitoramento** - monitoramento de capacidade e alertas
- **Guia de Solução de Problemas** - problemas de associação e sessão PFCP
- **Guia da Interface Web** - monitoramento de sessões e associações

Guia de Gerenciamento de Regras

Índice

1. Visão Geral
2. Regras de Detecção de Pacotes (PDR)
3. Regras de Ação de Encaminhamento (FAR)
4. Regras de Aplicação de QoS (QER)
5. Regras de Relatório de Uso (URR)
6. Relações entre Regras
7. Operações Comuns
8. Solução de Problemas

Visão Geral

OmniUPF utiliza um conjunto de regras interconectadas para classificar, encaminhar, moldar e rastrear o tráfego do plano do usuário. Essas regras são instaladas pelo SMF via PFCP e armazenadas em mapas eBPF para processamento de pacotes de alto desempenho. Compreender essas regras e suas relações é crítico para operar e solucionar problemas no UPF.

Tipos de Regras

Tipo de Regra	Propósito	Campo Chave	Instalado Por
PDR (Regra de Detecção de Pacote)	Classificar pacotes em fluxos	TEID ou IP do UE	SMF via Estabelecimento/Modificação de Sessão PFCP
FAR (Regra de Ação de Encaminhamento)	Determinar a ação de encaminhamento	ID do FAR	SMF via Estabelecimento/Modificação de Sessão PFCP
QER (Regra de Aplicação de QoS)	Aplicar limites de largura de banda e marcação	ID do QER	SMF via Estabelecimento/Modificação de Sessão PFCP
URR (Regra de Relatório de Uso)	Rastrear volumes de dados para cobrança	ID do URR	SMF via Estabelecimento/Modificação de Sessão PFCP

Fluxo de Processamento de Regras



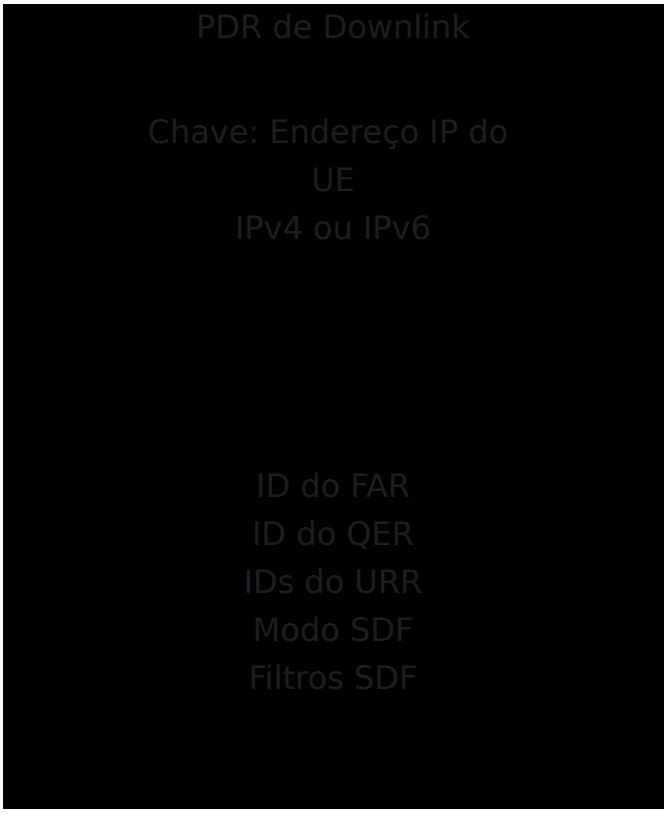


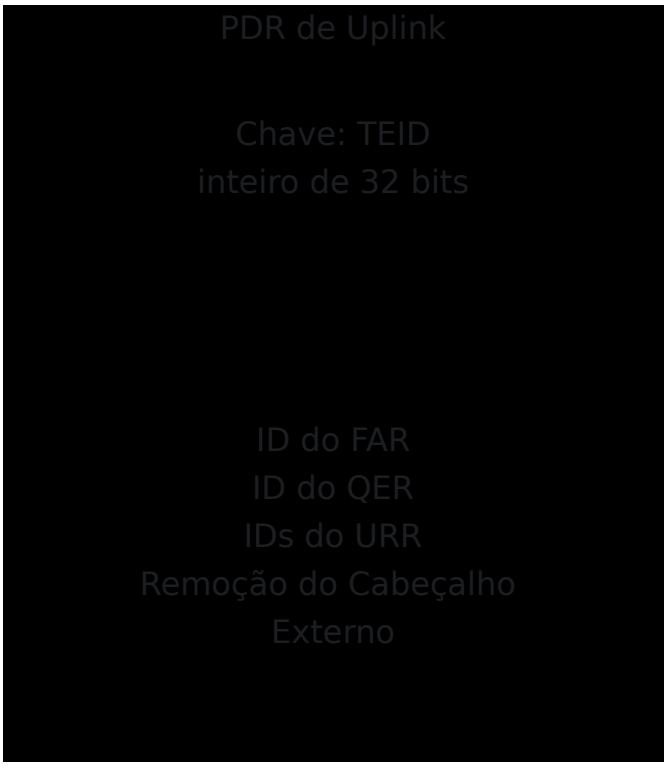
Regras de Detecção de Pacotes (PDR)

Propósito

As PDRs classificam pacotes de entrada em fluxos de tráfego. Elas são o ponto de entrada para todo o processamento de pacotes no UPF.

Estrutura da PDR





PDRs de Uplink

As PDRs de Uplink combinam pacotes que chegam na interface N3 da RAN.

Campo Chave: TEID (Identificador de Ponto de Extremidade do Túnel)

- inteiro não assinado de 32 bits
- Atribuído pelo SMF e sinalizado para o gNB
- Único por fluxo de tráfego do UE

Campos de Valor:

- **ID do FAR**: Referência à regra de ação de encaminhamento
- **ID do QER**: Referência à regra de aplicação de QoS (opcional)
- **IDs do URR**: Lista de regras de relatório de uso (opcional)
- **Remoção do Cabeçalho Externo**: Flag para remover a encapsulação GTP-U

Processo de Busca:

1. Extrair TEID do cabeçalho GTP-U
2. Busca hash no mapa eBPF `uplink_pdr_map`
3. Se uma combinação for encontrada, recuperar ID do FAR, ID do QER e IDs do URR
4. Se não houver combinação, descartar pacote

Exemplo:

```
TEID: 5678
ID do FAR: 2
ID do QER: 1
Remoção do Cabeçalho Externo: Falso
Modo SDF: Sem SDF
```

PDRs de Downlink

As PDRs de Downlink combinam pacotes que chegam na interface N6 da rede de dados.

Campo Chave: Endereço IP do UE

- endereço IPv4 (32 bits) ou endereço IPv6 (128 bits)
- Atribuído pelo SMF durante o estabelecimento da sessão PDU
- Único por UE

Campos de Valor:

- **ID do FAR**: Referência à regra de ação de encaminhamento
- **ID do QER**: Referência à regra de aplicação de QoS (opcional)
- **IDs do URR**: Lista de regras de relatório de uso (opcional)
- **Modo SDF**: Modo de filtro de Fluxo de Dados de Serviço
  - `Sem SDF`: Sem filtragem, todo o tráfego combina

    - `Somente SDF`: Somente tráfego combinado com SDF é encaminhado
    - `SDF + Padrão`: Tráfego combinado com SDF usa regras específicas, outro tráfego usa o FAR padrão
- **Filtros SDF**: Filtros específicos de aplicação (portas, protocolos, intervalos de IP)

Processo de Busca:

1. Extrair IP de destino do cabeçalho do pacote
2. Busca hash no `downlink_pdr_map` (IPv4) ou `downlink_pdr_map_ip6` (IPv6)
3. Se uma combinação for encontrada, verificar filtros SDF (se configurados)
4. Recuperar ID do FAR, ID do QER e IDs do URR
5. Se não houver combinação, descartar pacote

Exemplo:

```
IP do UE: 10.45.0.1
ID do FAR: 1
ID do QER: 1
Remoção do Cabeçalho Externo: Falso
Modo SDF: Sem SDF
```

Filtros SDF (Fluxo de Dados de Serviço)

Os filtros SDF fornecem classificação de tráfego específica de aplicação dentro de uma PDR.

Casos de Uso:

- Diferenciar o tráfego do YouTube da navegação na web
- Aplicar QoS diferente para VoIP vs. dados de melhor esforço
- Roteamento de aplicações específicas através de diferentes caminhos de rede

Critérios de Filtro:

- **Protocolo**: TCP, UDP, ICMP
- **Intervalo de Portas**: Portas de destino (por exemplo, 443 para HTTPS, 5060 para SIP)
- **Intervalo de Endereço IP**: Redes de destino específicas
- **Descrição do Fluxo**: Modelos de fluxo definidos pela 3GPP

Exemplo de Configuração SDF:

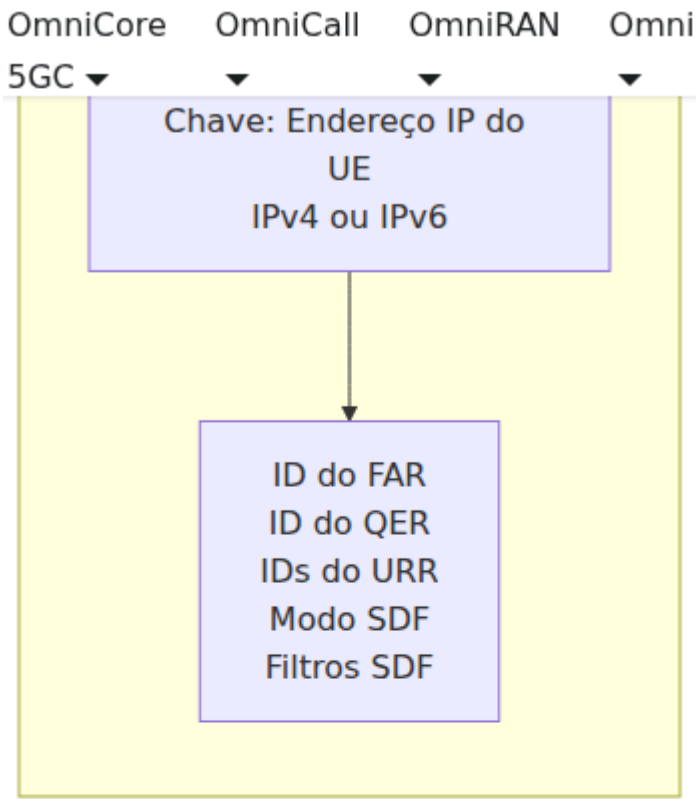
```
ID da PDR: 10
IP do UE: 10.45.0.1
Modo SDF: Somente SDF
Filtros SDF:
  - Protocolo: UDP, Portas: 5060-5061 → ID do FAR 5 (FAR de VoIP)
  - Protocolo: TCP, Porta: 443 → ID do FAR 1 (FAR Padrão)
```

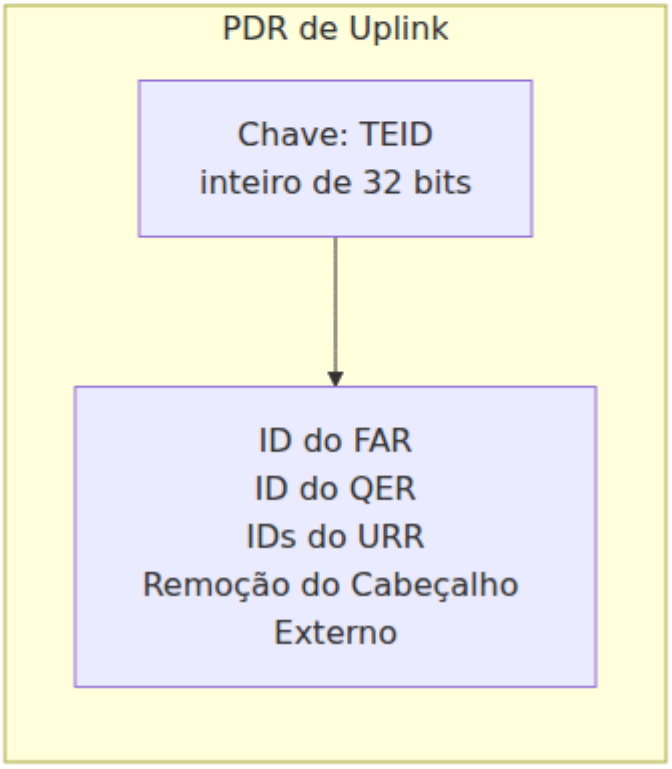
Regras de Ação de Encaminhamento (FAR)

Propósito

As FARs determinam o que fazer com pacotes que combinam com uma PDR. Elas definem ações de encaminhamento, parâmetros de encapsulação GTP-U e pontos de destino.

Estrutura da FAR





Flags de Ação

As ações da FAR são flags bit a bit que podem ser combinadas:

Flag	Bit	Valor	Descrição
ENCAMINHAR	1	2	Encaminhar pacote para o destino
BUFFER	2	4	Armazenar pacote em buffer
DESCARTAR	0	1	Descartar pacote
NOTIFICAR	3	8	Enviar notificação para o plano de controle
DUPLICAR	4	16	Duplicar pacote para múltiplos destinos

Combinações Comuns de Ação:

- `Ação: 2 (ENCAMINHAR)` - Encaminhamento normal (mais comum)
- `Ação: 6 (ENCAMINHAR + BUFFER)` - Encaminhar e armazenar durante a transferência
- `Ação: 4 (BUFFER)` - Somente armazenar (durante a troca de caminho)
- `Ação: 1 (DESCARTAR)` - Descartar pacote (raro, geralmente para aplicação de políticas)

Controle de Buffering

A flag BUFFER (bit 2) controla o armazenamento de pacotes durante eventos de mobilidade. O armazenamento é um recurso crítico do UPF que previne a perda de pacotes durante transições de estado do UE.

Quando o Buffering é Usado

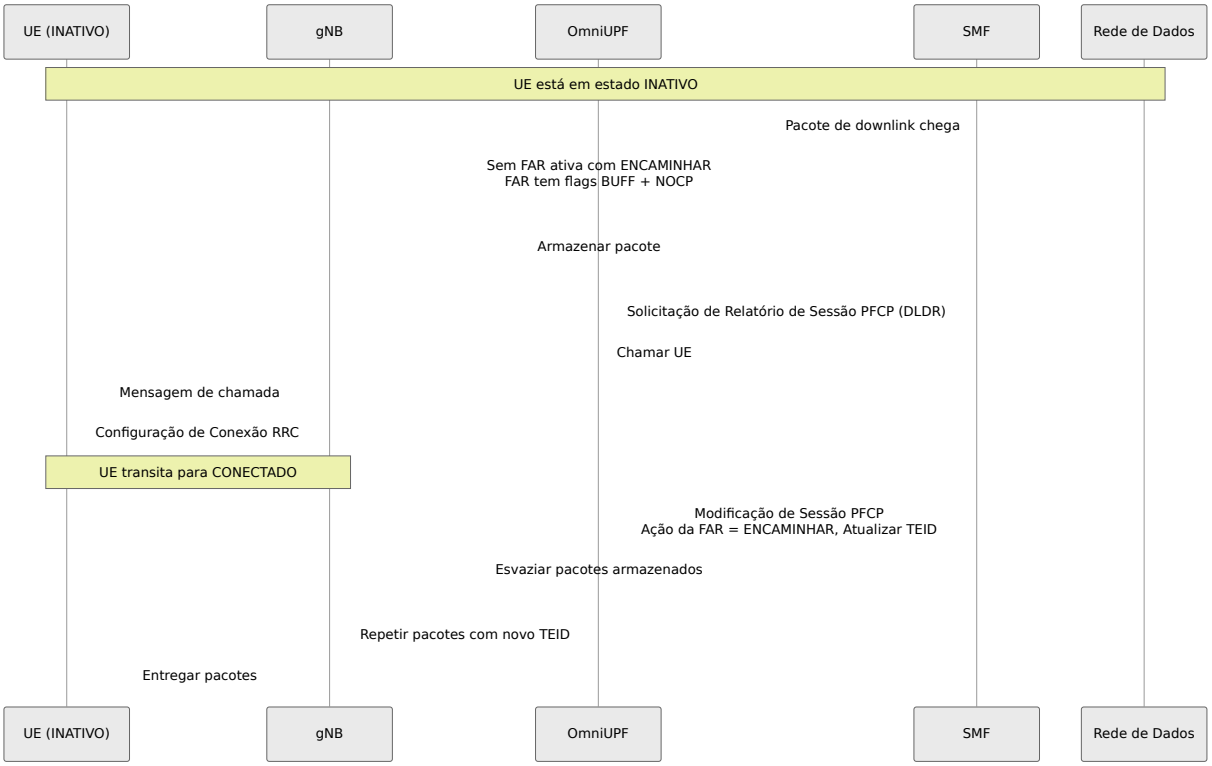
Transição de Inativo para Conectado: Quando pacotes de downlink chegam para um UE em estado INATIVO (não conectado ao gNB), o UPF:

1. Armazena os pacotes

2. Envia uma Notificação de Dados de Downlink (DLDR) para o SMF
3. O SMF chama o UE para acordar e conectar
4. Uma vez conectado, o SMF atualiza a FAR com a ação de ENCAMINHAR
5. O UPF esvazia os pacotes armazenados para o UE

Transferência (Conectado para Conectado): Durante a transferência de gNB para gNB, o UPF temporariamente armazena pacotes para prevenir perda:

1. A conexão antiga do gNB é descartada
2. O SMF define a ação da FAR como BUFFER
3. Pacotes são enfileirados durante a troca de caminho
4. O UE conecta ao novo gNB
5. O SMF atualiza a FAR com novo TEID e ação de ENCAMINHAR
6. O UPF esvazia os pacotes para o novo gNB



Capacidade e Limites do Buffer

Limites Globais de Buffer:

- **Máximo Total de Pacotes**: 100.000 (configurável)
- **Máximo Total de Bytes**: Baseado na memória disponível

- **TTL (Tempo de Vida)**: 60 segundos (configurável)
- **Pacotes que excedem o TTL**: Descartados automaticamente

Limites por FAR:

- **Máximo de Pacotes por FAR**: 10.000 (configurável)
- **Propósito**: Prevenir que uma única FAR esgote a capacidade do buffer

Comportamento de Overflow do Buffer:

- Quando o limite global ou por FAR é atingido, novos pacotes são descartados
- Métricas rastreiam descartes com `reason="global_limit"` ou `reason="far_limit"`
- Pacotes mais antigos NÃO são automaticamente removidos (descartes explícitos apenas na expiração do TTL)

Notificação de Dados de Downlink (DLDR)

Quando o UPF armazena um pacote para um UE INATIVO, ele envia uma Solicitação de Relatório de Sessão PFCP para o SMF:

Conteúdos do DLDR:

- **Tipo de Relatório**: Relatório de Dados de Downlink (DLDR)
- **ID do FAR**: O FAR que acionou o armazenamento
- **Informações do Serviço de Dados de Downlink**: QFI opcional, Indicador de Política de Chamada

Ações do SMF no DLDR:

1. Chamar o UE via AMF → gNB
2. Aguardar o UE estabelecer a conexão RRC
3. Enviar Solicitação de Modificação de Sessão PFCP para atualizar a FAR
4. A ação da FAR muda de `BUFF+NOCP` para `FORW`
5. O UPF esvazia os pacotes armazenados

Métricas para DLDR:

- `upf_dldr_sent_total`: Total de DLDRs enviados
- `upf_dldr_send_errors`: DLDRs falhados
- `upf_buffer_notify_to_flush_duration_seconds`: Latência do DLDR até o esvaziamento

Veja Referência de Métricas para a lista completa.

Operações de Buffering

Habilitar Buffering (Definir flag BUFF):

- Ação da FAR `|= 0x04` (definir bit 2)
- Exemplo: `Ação: 2 (FORW)` → `Ação: 6 (FORW+BUFF)`
- Usado durante a preparação para a transferência

Modo Somente Buffer (BUFF sem FORW):

- Ação da FAR `= 0x04` (somente BUFF)
- Pacotes são armazenados, mas NÃO são encaminhados
- Usado para estado INATIVO do UE (pendente de chamada)

Desabilitar Buffering (Limpar flag BUFF):

- Ação da FAR `&= ~0x04` (limpar bit 2)
- Exemplo: `Ação: 6 (FORW+BUFF)` → `Ação: 2 (FORW)`
- Pacotes armazenados permanecem até serem esvaziados ou limpos

Esvaziar Buffer:

- Repetir todos os pacotes armazenados usando as regras da FAR **atual**
- Pacotes são encaminhados com TEID/destino atualizados
- O buffer é esvaziado após o esvaziamento bem-sucedido
- A FAR deve ter a ação FORW definida

Limpar Buffer:

- Descartar todos os pacotes armazenados sem encaminhamento
- Usar quando a transferência falhar ou a sessão for excluída

- Métricas rastreiam com `reason="cleared"`

Monitorando Pacotes Armazenados

Página de Buffers (UI Web): Navegue até **Buffers** para visualizar:

- Total de pacotes armazenados
- Total de bytes armazenados
- Número de FARs com pacotes armazenados
- Contagens de pacotes por FAR
- Timestamp do pacote mais antigo
- Habilitar/Desabilitar buffering por FAR
- Operações de esvaziamento ou limpeza

Indicadores Chave:

- **Pacotes > 10 segundos antigos**: Atraso potencial na chamada
- **Pacotes > 30 segundos antigos**: Falha provável na chamada, limpar buffer
- **Alta contagem de pacotes**: Verificar sessões travadas ou falhas de chamada

Métricas Prometheus:

- `upf_buffer_packets_current`: Pacotes armazenados atuais
- `upf_buffer_bytes_current`: Bytes armazenados atuais
- `upf_buffer_fars_active`: FARs com pacotes armazenados
- `upf_buffer_packets_dropped{reason}`: Contagens de pacotes descartados

Veja Referência de Métricas para métricas completas de buffer.

Cenários Comuns de Buffering

Cenário 1: UE INATIVO Dados de Downlink

```
Estado Inicial:
- UE em modo INATIVO (sem conexão gNB)
- Ação da FAR: 0x04 (somente BUFF)

Chegada de Dados:
1. DN envia pacote de downlink
2. UPF combina PDR, aplica FAR
3. FAR tem flag BUFF → pacote armazenado
4. UPF envia DLDR para SMF
5. SMF chama UE
6. UE conecta ao gNB
7. SMF modifica FAR: Ação = 0x02 (FORW)
8. UPF esvazia pacotes armazenados com novo TEID
```

Cenário 2: Preparação para Transferência

```
Estado Inicial:
- UE conectado ao gNB-1 (TEID 1234)
- Ação da FAR: 0x02 (FORW)

Processo de Transferência:
1. SMF modifica FAR: Ação = 0x06 (FORW+BUFF)
2. Pacotes encaminhados para gNB-1 E armazenados
3. UE muda para gNB-2
4. SMF modifica FAR: TEID = 5678, Ação = 0x02 (FORW)
5. UPF esvazia pacotes armazenados para gNB-2 com novo TEID
6. Sem perda de pacotes durante a transferência
```

Cenário 3: Troca de Caminho

```
Estado Inicial:
- UE conectado, fluxo de dados ativo

Troca de Caminho:
1. SMF modifica FAR: Ação = 0x04 (somente BUFF)
2. Todos os pacotes de entrada armazenados (não encaminhados)
3. Rede reconfigura caminho
4. SMF modifica FAR: Ação = 0x02 (FORW), novo destino
5. UPF esvazia todos os pacotes armazenados para o novo caminho
```

Criação de Cabeçalho Externo

Determina se a encapsulação GTP-U deve ser adicionada.

FAR de Uplink (N3 → N6):

- Criação de Cabeçalho Externo: Falso
- Ação: Remover GTP-U, encaminhar pacote IP nativo

FAR de Downlink (N6 → N3):

- Criação de Cabeçalho Externo: Verdadeira
- IP Remoto: endereço IP do gNB (por exemplo, 198.51.100.10)
- TEID: ID do túnel para tráfego do UE
- Ação: Adicionar cabeçalho GTP-U, encaminhar para gNB

Busca da FAR na UI Web

A página de Gerenciamento de Regras fornece busca da FAR por ID:

Passos:

1. Navegue até Regras → aba FARs
2. Insira o ID da FAR no campo de busca
3. Clique em "Buscar" para visualizar detalhes da FAR

Informações Exibidas:

- ID da FAR
- Ação (numérica + flags decodificadas)
- Status de Buffering (LIGADO/DESLIGADO)
- Criação de Cabeçalho Externo
- Endereço IP remoto (com representação inteira)
- TEID
- Marcação de Nível de Transporte

Regras de Aplicação de QoS (QER)

Propósito

As QERs aplicam parâmetros de Qualidade de Serviço aos fluxos de tráfego, incluindo limites de largura de banda e marcação de pacotes.

Estrutura da QER


Parâmetros da QER
QFI
Identificador de Fluxo
de QoS
Estado do Portão UL
Aberto/Fechado
Estado do Portão DL
Aberto/Fechado
ID do QER
Identificador Único
MBR Uplink
Taxa Máxima de Bits
MBR Downlink
Taxa Máxima de Bits
GBR Uplink
Taxa Garantida de Bits
GBR Downlink
Taxa Garantida de Bits

Parâmetros de QoS

QFI (Identificador de Fluxo de QoS):

- Identificador de 6 bits para fluxos de QoS 5G
- Valores de 1-9 são padronizados (por exemplo, QFI 9 = portadora padrão)
- Usado para marcação de pacotes no 5GC

Estado do Portão:

- **Aberto (0)**: Tráfego permitido
- **Fechado (não zero)**: Tráfego bloqueado

Taxa Máxima de Bits (MBR):

- Largura de banda máxima permitida para o fluxo de tráfego
- Especificado em kbps
- **MBR = 0**: Sem limite de taxa (ilimitado)
- Tráfego que excede o MBR é descartado

Taxa Garantida de Bits (GBR):

- Largura de banda mínima garantida para o fluxo de tráfego
- Especificado em kbps
- **GBR = 0**: Melhor esforço (sem garantia)
- **GBR > 0**: Fluxo priorizado com largura de banda garantida

Tipos de Fluxos de QoS

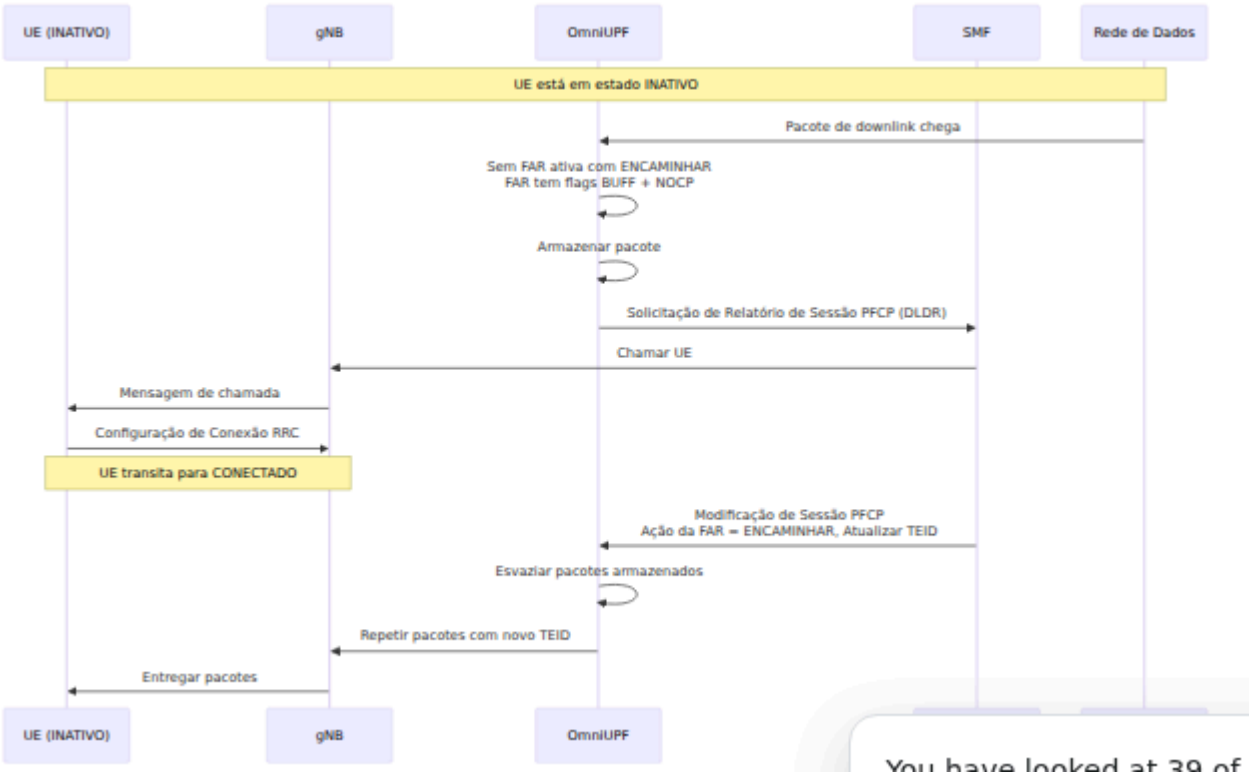
Fluxos de Melhor Esforço (GBR = 0):

```
ID do QER: 1
QFI: 9
MBR Uplink: 100000 kbps (100 Mbps)
MBR Downlink: 100000 kbps (100 Mbps)
GBR Uplink: 0 kbps
GBR Downlink: 0 kbps
```

Fluxos Garantidos (GBR > 0):

```
ID do QER: 2
QFI: 1
MBR Uplink: 10000 kbps (10 Mbps)
MBR Downlink: 10000 kbps (10 Mbps)
GBR Uplink: 5000 kbps (5 Mbps)
GBR Downlink: 5000 kbps (5 Mbps)
```

Algoritmo de Aplicação de QoS



Mecanismo de Aplicação de MBR

OmniUPF aplica limites de MBR (Taxa Máxima de Bits) usando um **limitador de taxa de janela deslizante** implementado no caminho de dados eBPF. Este algoritmo opera com precisão de nanossegundos diretamente na camada XDP, garantindo desempenho em linha sem trocas de contexto do kernel.

Como Funciona

Algoritmo: Limitação de Taxa de Janela Deslizante

Para cada pacote, o UPF realiza as seguintes verificações:

1. **Verificação do Estado do Portão**: Se o estado do portão é `FECHADO` (não zero), descartar pacote imediatamente
2. **Verificação do MBR**: Se MBR = 0, ignorar limitação de taxa (largura de banda ilimitada)
3. **Cálculo do Tempo de Transmissão**:

```
tx_time = (tamanho_pacote_bytes × 8) × (1.000.000.000 ns/sec) /
MBR_kbps
```

4. **Verificação da Janela**: Se o tempo atual estiver dentro da janela deslizante de 5ms, descartar pacote
5. **Avanço da Janela**: Se o pacote for permitido, avançar a janela por `tx_time`

Exemplo de Cálculo:

Assuma:

- MBR = 100.000 kbps (100 Mbps)
- Tamanho do pacote = 1500 bytes
- Tamanho da janela = 5.000.000 ns (5 ms)

```
Passo 1: Calcular tempo de transmissão a 100 Mbps
tx_time = (1500 bytes × 8 bits/byte) × (1.000.000.000 ns/sec) /
100.000.000 bps
        = 12.000.000.000 / 100.000.000
        = 120 ns

Passo 2: Verificar se o pacote cabe na janela
current_time = 1000000000 ns
window_start = 999990000 ns
if (window_start + tx_time > current_time):
    DESCARTAR pacote (excederia limite de taxa)

Passo 3: Se permitido, avançar janela
window_start = window_start + 120 ns
PASSAR pacote
```

Comportamento da Janela Deslizante

Tamanho da Janela de 5ms:

- O algoritmo utiliza uma janela deslizante de 5 milissegundos
- A janela é redefinida automaticamente se inativa por mais de 5ms
- Previne a fome de explosões enquanto aplica a taxa média

Tratamento de Explosões:

- Pequenas explosões são permitidas dentro da janela de 5ms
- Tráfego sustentado acima do MBR é limitado por taxa
- Mais preciso do que algoritmos simples de balde de token

Limitação de Taxa por Direção:

- MBR de uplink usa timestamp `qer->ul_start`
- MBR de downlink usa timestamp `qer->dl_start`
- Cada direção é limitada por taxa independentemente

Pontos de Aplicação de Limitação de MBR

Uplink (N3 → N6):

1. Pacote chega na interface N3 (do gNB)
2. Busca PDR por TEID
3. Busca QER por ID do QER
4. Verificar `ul_gate_status` → descartar se fechado
5. Aplicar `limit_rate_sliding_window()` com `ul_maximum_bitrate`
6. Se passar, encaminhar para N6 e atualizar contadores do URR

Downlink (N6 → N3):

1. Pacote chega na interface N6 (da Rede de Dados)
2. Busca PDR por endereço IP do UE
3. Busca QER por ID do QER
4. Verificar `dl_gate_status` → descartar se fechado
5. Aplicar `limit_rate_sliding_window()` com `dl_maximum_bitrate`
6. Se passar, adicionar cabeçalho GTP-U e encaminhar para N3

Loopback N9 (SGWU ↔ PGWU):

- Tanto QERs de uplink quanto de downlink podem ser aplicados em cenários de loopback N9
- Cada QER é verificada independentemente nas fronteiras SGWU e PGWU

MBR vs. Throughput Observado

Por que o throughput observado pode diferir do MBR:

- **Sobrecarga de Protocolo**: Cabeçalhos GTP-U, UDP, IP adicionam ~50-60 bytes por pacote
- **Variação no Tamanho do Pacote**: Pacotes menores = mais sobrecarga, menor eficiência
- **Precisão da Limitação de Taxa**: A aplicação acontece por pacote, não por byte
- **Comportamento de Redefinição da Janela**: Períodos de inatividade de 5ms permitem explosões breves acima do MBR

Exemplo:

```
MBR Configurado: 100 Mbps
Throughput Observado: ~95-98 Mbps (devido à sobrecarga GTP-
U/UDP/IP)
```

Como Verificar a Limitação de Taxa:

1. Verifique os contadores de volume do URR ao longo do tempo: `upf_urr_*_volume_bytes`
2. Calcule o throughput: `(volume_delta_bytes × 8) / time_delta_seconds / 1000` = kbps
3. Compare com o MBR configurado na QER

GBR (Taxa Garantida de Bits)

Importante: O OmniUPF **não** aplica atualmente mínimos de GBR. GBR é armazenado na QER, mas não é usado para priorização de tráfego ou controle de admissão.

Comportamento do GBR:

- Valores de GBR são aceitos do SMF via PFCP
- GBR é armazenado no mapa QER e visível via API

- **Sem reserva de largura de banda** ou priorização de tráfego com base no GBR
- GBR serve como metadados para rastreamento do tipo de fluxo (melhor esforço vs. garantido)

Melhoria Futura:

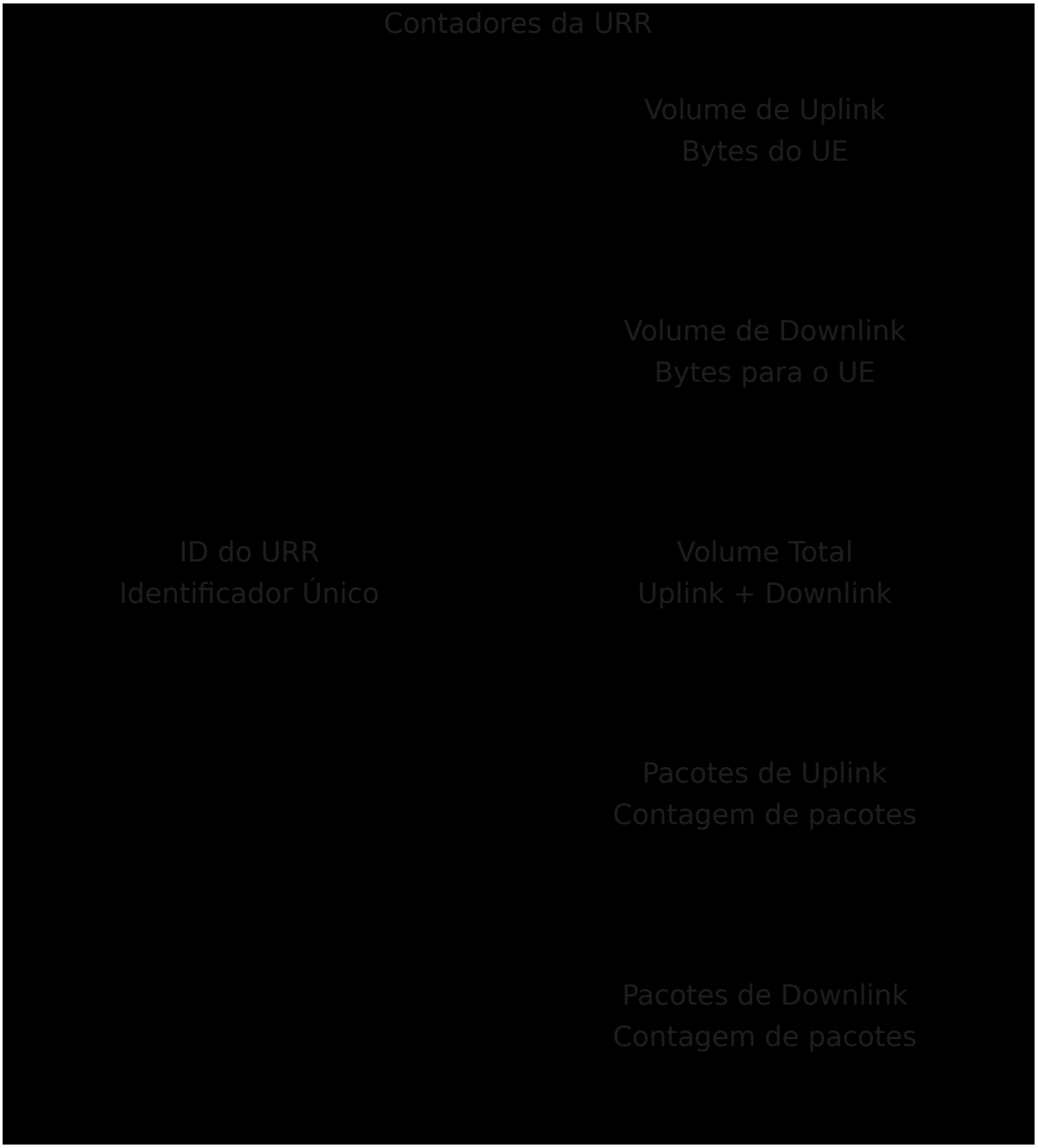
- A aplicação do GBR requer agendamento de tráfego ou enfileiramento ponderado
- Pode ser implementado usando capacidades de QoS eBPF em lançamentos futuros

Regras de Relatório de Uso (URR)

Propósito

As URRs rastreiam volumes de dados para cobrança, análises e aplicação de políticas. Elas mantêm contadores de pacotes e bytes que são relatados ao SMF para registros de cobrança.

Estrutura da URR



Rastreamento de Volume

Volume de Uplink:

- Bytes transmitidos do UE para a Rede de Dados
- Medido após a desencapsulação GTP-U
- Inclui cabeçalho IP e payload

Volume de Downlink:

- Bytes transmitidos da Rede de Dados para o UE
- Medido antes da encapsulação GTP-U
- Inclui cabeçalho IP e payload

Volume Total:

- Soma dos volumes de uplink e downlink
- Usado para relatório de uso total

Gatilhos de Relatório de Uso

As URRs podem acionar relatórios com base em:

Limite de Volume:

- Relatar quando o volume exceder o limite configurado
- Exemplo: Relatar a cada 1 GB de uso

Limite de Tempo:

- Relatar em intervalos periódicos
- Exemplo: Relatar a cada 5 minutos

Baseado em Eventos:

- Relatar na terminação da sessão
- Relatar na mudança de QoS
- Relatar na transferência

Formatação de Exibição de Volume

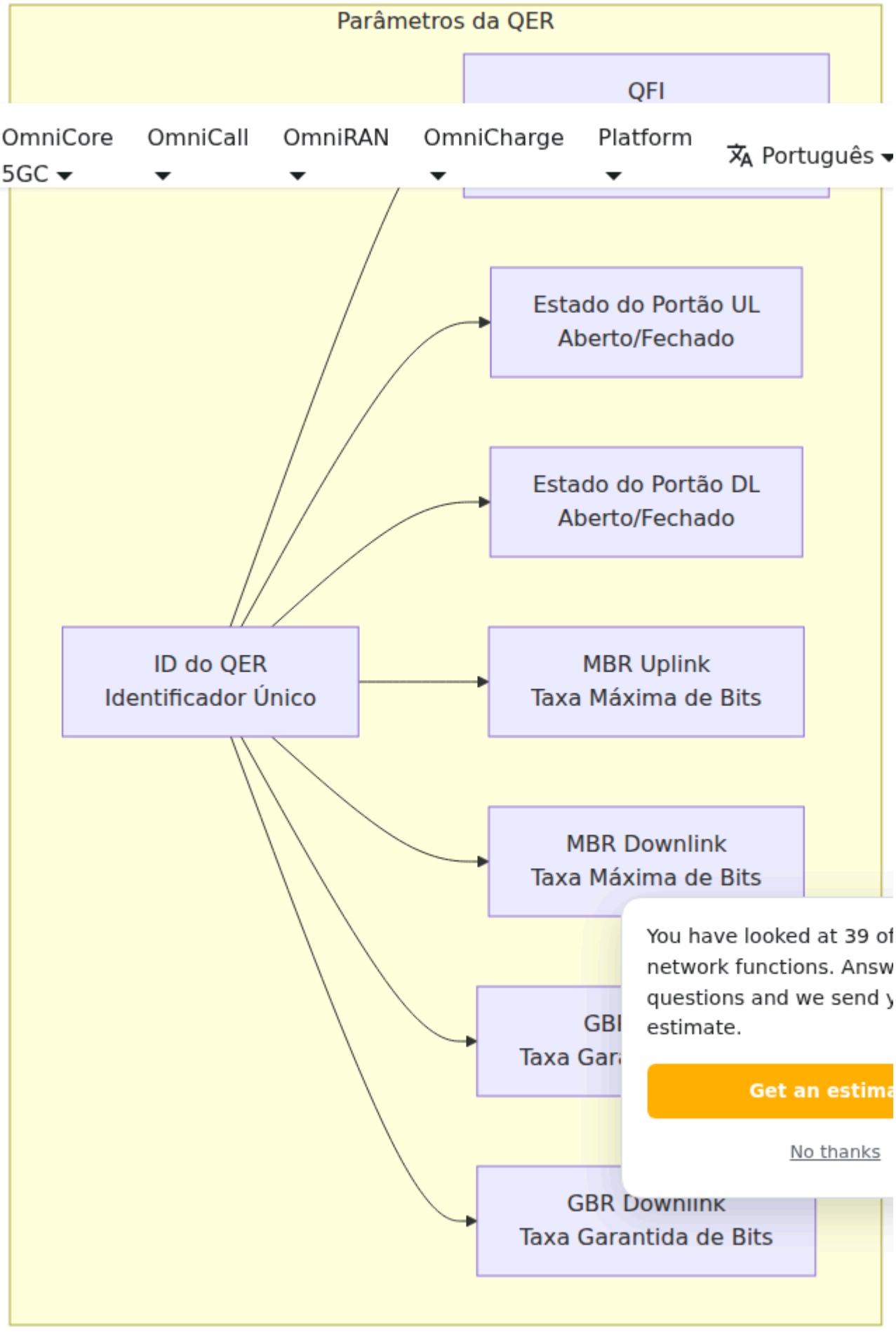
A UI Web formata automaticamente o volume em unidades legíveis por humanos:

Bytes	Exibição
0 - 1023	B (Bytes)
1024 - 1048575	KB (Kilobytes)
1048576 - 1073741823	MB (Megabytes)
1073741824 - 1099511627775	GB (Gigabytes)
1099511627776+	TB (Terabytes)

Exemplo:

```
ID do URR: 0
Volume de Uplink: 12.3 KB
Volume de Downlink: 9.0 KB
Volume Total: 21.3 KB
```

Fluxo de Relatório da URR

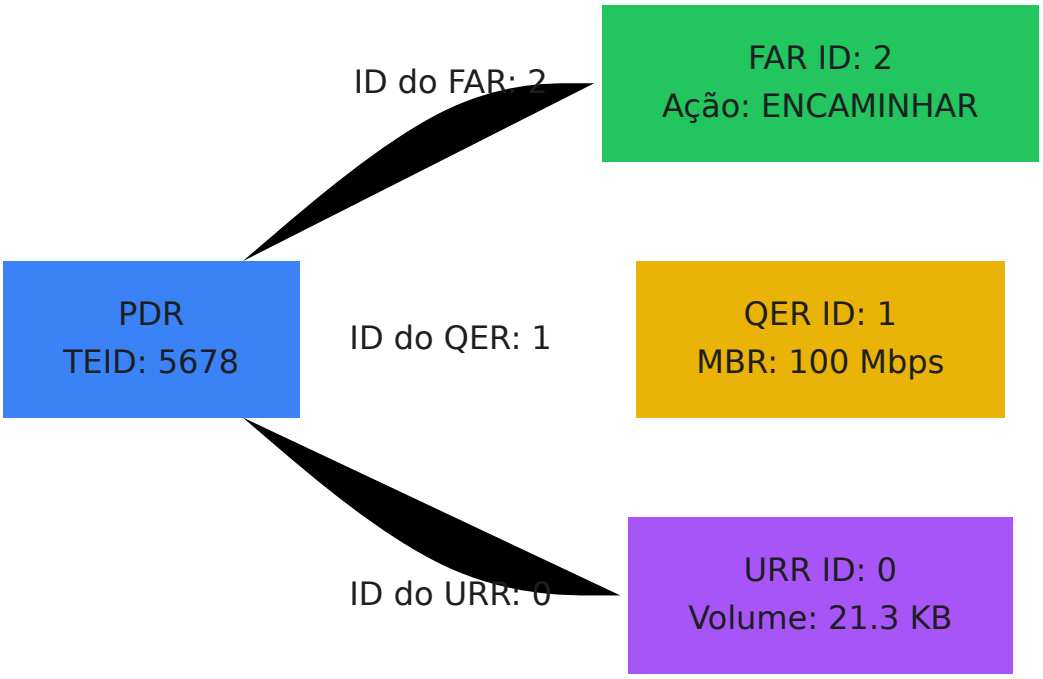


You have looked at 39 of
network functions. Answ
questions and we send y
estimate.
Get an estima
No thanks

Relações entre Regras

Cadeia PDR → FAR → QER → URR

Cada PDR referencia uma FAR, que pode referenciar uma QER e uma ou mais URRs.



Exemplo de Configuração de Sessão

PDR de Uplink:

```
TEID: 5678
ID do FAR: 2
ID do QER: 1
IDs do URR: [0]
Remoção do Cabeçalho Externo: Falso
```

PDR de Downlink:

```
IP do UE: 10.45.0.1
ID do FAR: 1
ID do QER: 1
IDs do URR: [0]
Modo SDF: Sem SDF
```

FAR ID 1 (Downlink):

```
Ação: 2 (ENCAMINHAR)
Criação de Cabeçalho Externo: Verdadeira
IP Remoto: 198.51.100.10
TEID: 5678
```

FAR ID 2 (Uplink):

```
Ação: 2 (ENCAMINHAR)
Criação de Cabeçalho Externo: Falsa
```

QER ID 1:

```
QFI: 9
MBR Uplink: 100000 kbps
MBR Downlink: 100000 kbps
GBR Uplink: 0 kbps
GBR Downlink: 0 kbps
```

URR ID 0:

```
Volume de Uplink: 12.3 KB
Volume de Downlink: 9.0 KB
Volume Total: 21.3 KB
```

Operações Comuns

Visualizar Regras para uma Sessão

Via Página de Sessões:

1. Navegue até Sessões
2. Encontre o UE pelo IP ou TEID
3. Clique em "Expandir" para visualizar todas as regras (PDR, FAR, QER, URR)

Via Página de Regras:

1. Navegue até Regras
2. Use a busca por TEID (uplink) ou IP do UE (downlink) na aba PDR
3. Anote o ID do FAR, ID do QER, IDs do URR
4. Mude para as abas FAR/QER/URR para visualizar as regras referenciadas

Habilitar/Desabilitar Buffering

Cenário: Durante a transferência, armazenar pacotes para prevenir perda

Passos:

1. Navegue até Regras → FARs
2. Insira o ID da FAR no campo de busca
3. Clique em "Buscar"
4. Se o buffering estiver DESLIGADO, clique em "Habilitar Buffering"
5. Verifique se o bit 2 da ação da FAR está definido (valor da ação aumenta em 4)

Alternativa via Página de Buffers:

1. Navegue até Buffers
2. Visualize FARs com buffering habilitado
3. Clique em "Desabilitar Buffer" quando a transferência for concluída

Monitorar Conformidade de QoS

Verifique se o tráfego está sendo limitado por taxa:

1. Navegue até Regras → QERs
2. Encontre o ID do QER associado à sessão do UE
3. Anote os valores de MBR Uplink e MBR Downlink
4. Compare com a taxa de crescimento do volume do URR

Calcule o Throughput Médio:

```
Throughput (kbps) = (Delta de Volume em bytes × 8) / (Delta de
Tempo em segundos × 1000)
```

Se o throughput se aproximar do MBR, o tráfego está sendo limitado por taxa.

Rastrear Uso de Dados

Monitore os volumes do URR:

1. Navegue até Regras → URRs
2. Visualize os volumes de uplink, downlink e total
3. Classifique por Volume Total para encontrar os maiores usuários
4. Atualize periodicamente para observar o crescimento do volume

Casos de Uso:

- Verificar a integração de cobrança
- Detectar uso anômalo de dados
- Planejar capacidade com base em padrões de tráfego

Solução de Problemas

Sem Tráfego Fluindo

Verifique a PDR:

1. Verifique se a PDR existe para TEID (uplink) ou IP do UE (downlink)
2. Confirme se o ID do FAR é válido
3. Verifique se os filtros SDF não estão bloqueando o tráfego

Verifique a FAR:

1. Verifique se a ação da FAR é ENCAMINHAR (não DESCARTAR ou somente BUFFER)
2. Confirme se a criação do cabeçalho externo corresponde à direção
3. Verifique se o IP Remoto e o TEID estão corretos para downlink

Verifique a QER:

1. Verifique se o Estado do Portão está Aberto (0)
2. Verifique se o MBR não é muito restritivo

Pacotes Sendo Descartados

Verifique a Limitação de Taxa da QER:

1. Navegue até Regras → QERs
2. Verifique se o MBR é adequado para a carga de tráfego
3. Verifique se o crescimento do volume do URR corresponde ao throughput esperado

Verifique a Ação da FAR:

1. Navegue até Regras → FARs
2. Verifique se a ação é ENCAMINHAR, não DESCARTAR
3. Verifique se o buffering não está preso no modo somente BUFFER

Problemas de Buffering

Pacotes presos no buffer:

1. Navegue até a página de Buffers
2. Verifique o timestamp do pacote mais antigo
3. Se > 30 segundos, a transferência pode ter falhado

4. Esvazie ou limpe manualmente o buffer
5. Desabilite o buffering na FAR

Overflow do buffer:

1. Verifique pacotes totais vs. Máximo Total (padrão 100.000)
2. Verifique pacotes por FAR vs. Máximo por FAR (padrão 10.000)
3. Limpe buffers se estiverem cheios
4. Investigue por que o buffering não foi desabilitado

URR Não Rastreando

Contadores de volume em zero:

1. Verifique se a PDR referencia o ID do URR
2. Verifique se os pacotes estão combinando com a PDR
3. Verifique se a FAR está encaminhando (não descartando) pacotes
4. Confirme se o ID do URR existe no mapa do URR

Volume não relatando ao SMF:

1. Verifique a configuração do Relatório de Sessão PFCP
2. Verifique os gatilhos de relatório do URR (limites de volume/tempo)
3. Revise os logs para mensagens de Relatório de Sessão PFCP

Documentação Relacionada

- **Guia de Operações do UPF** - Visão geral da arquitetura e componentes do OmniUPF
- **Guia de Operações da UI Web** - Uso do painel de controle para visualização de regras
- **Guia de Monitoramento** - Estatísticas e monitoramento de capacidade
- **Guia de Solução de Problemas** - Problemas comuns e diagnósticos

Guia de Solução de Problemas do OmniUPF

Índice

1. Visão Geral
2. Ferramentas de Diagnóstico
3. Problemas de Instalação
4. Problemas de Configuração
5. Problemas de Associação PFCP
6. Problemas de Processamento de Pacotes
7. Problemas de XDP e eBPF
8. Problemas de Desempenho
9. Problemas Específicos de Hipervisor
10. Problemas de NIC e Driver
11. Falhas na Estabelecimento de Sessão
12. Problemas de Bufferização

Visão Geral

Este guia fornece procedimentos sistemáticos de solução de problemas para problemas comuns do OmniUPF. Cada seção inclui sintomas, etapas de diagnóstico, causas raiz e procedimentos de resolução.

Lista de Verificação Rápida de Diagnóstico

Antes de realizar uma solução de problemas profunda, verifique:

```
# 1. Verifique se o OmniUPF está em execução
systemctl status omniupf

# 2. Verifique a associação PFCP
curl http://localhost:8080/api/v1/upf_pipeline

# 3. Verifique se os mapas eBPF estão carregados
ls /sys/fs/bpf/

# 4. Verifique se o programa XDP está anexado
ip link show | grep -i xdp

# 5. Verifique os logs do kernel em busca de erros
dmesg | tail -50
journalctl -u omniupf -n 50
```

Ferramentas de Diagnóstico

API REST do OmniUPF

Verifique o status do UPF:

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

Verifique as associações PFCP:

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

Verifique a contagem de sessões:

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

Verifique a capacidade do mapa eBPF:

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

Verifique as estatísticas de pacotes:

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

Verifique as estatísticas do XDP:

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

Inspeção do Mapa eBPF

Liste todos os mapas eBPF:

```
ls -lh /sys/fs/bpf/
bpftool map list
```

Mostre detalhes do mapa:

```
bpftool map show
bpftool map dump name pdr_map_downlin
```

Conte entradas no mapa:

```
bpftool map dump name far_map | grep -c "key:"
```

Inspeção do Programa XDP

Verifique se o programa XDP está anexado:

```
ip link show eth0 | grep xdp
```

Liste todos os programas XDP:

```
bpftool net list
```

Mostre detalhes do programa XDP:

```
bpftool prog show
```

Despeje estatísticas do XDP:

```
bpftool prog dump xlated name xdp_upf_func
```

Depuração de Rede

Capture tráfego PFCP na N4 (plano de controle):

```
# PFCP não é processado pelo XDP, tcpdump funciona normalmente
tcpdump -i eth0 -n udp port 8805 -w /tmp/pfcp_traffic.pcap
```

Capture tráfego GTP-U na N3 (requer captura fora de banda):

```
# AVISO: O tcpdump padrão no host UPF NÃO PODE capturar pacotes
processados pelo XDP!
# O XDP processa GTP-U antes que a pilha de rede do kernel veja os
pacotes.

# Use captura fora de banda em vez disso:
# 1. TAP de rede entre gNB e UPF
# 2. Espelhamento de porta de switch/SPAN para copiar tráfego N3
# 3. Espelhamento de porta de switch virtual para VM de análise

# No host de análise/monitoramento (NÃO no UPF):
# tcpdump -i <mirror_interface> -n udp port 2152 -w
/tmp/n3_capture.pcap

# Ou use a API de estatísticas para contagens de pacotes:
curl http://localhost:8080/api/v1/packet_stats
curl http://localhost:8080/api/v1/n3n6_stats
```

Monitore contadores de pacotes:

```
watch -n 1 'ip -s link show eth0'
```

Verifique a tabela de roteamento:

```
ip route show
ip route get 10.45.0.100  # Verifique a rota para o IP do UE
```

Verifique a tabela ARP:

```
ip neigh show
```

Problemas de Instalação

Problema: "sistema de arquivos eBPF não montado"

Sintomas:

```
ERRO[0000] falha ao carregar objetos eBPF: monte o sistema de
arquivos bpf em /sys/fs/bpf
```

Causa: sistema de arquivos eBPF não montado

Resolução:

```
# Monte o sistema de arquivos eBPF
sudo mount bpffs /sys/fs/bpf -t bpf

# Torne persistente (adicione ao /etc/fstab)
echo "bpffs /sys/fs/bpf bpf defaults 0 0" | sudo tee -a /etc/fstab

# Verifique a montagem
mount | grep bpf
```

Problema: Versão do kernel muito antiga

Sintomas:

```
ERRO[0000] versão do kernel 5.4.0 é muito antiga, o mínimo
requerido é 5.15.0
```

Causa: versão do kernel Linux abaixo do requisito mínimo

Resolução:

```
# Verifique a versão do kernel
uname -r

# Atualize o kernel (Ubuntu/Debian)
sudo apt update
sudo apt install linux-generic-hwe-22.04
sudo reboot

# Verifique o novo kernel
uname -r  # Deve ser >= 5.15.0
```

Problema: Dependência libbpf ausente

Sintomas:

```
erro ao carregar bibliotecas compartilhadas: libbpf.so.0: não é
possível abrir arquivo de objeto compartilhado
```

Causa: biblioteca libbpf não instalada

Resolução:

```
# Instale libbpf (Ubuntu/Debian)
sudo apt update
sudo apt install libbpf-dev

# Verifique a instalação
ldconfig -p | grep libbpf
```

Problemas de Configuração

Problema: Arquivo de configuração inválido

Sintomas:

```
ERRO[0000] não foi possível ler o arquivo de configuração: erros
de deserialização
```

Causa: erro de sintaxe no arquivo de configuração Elixir

Resolução:

```
# Verifique a sintaxe do arquivo de configuração
cd /opt/omniupf && bin/upf_ex eval "IO.puts(:ok)"

# Problemas comuns:
# - Falta de aspas em torno de valores de string
# - Sintaxe Elixir incorreta (por exemplo, usando : em vez de =)
# - Strings ou colchetes não fechados

# O arquivo de configuração está em /etc/omniupf/runtime.exs
# Exemplo de sintaxe correta:
# xdp_interfaces = "eth0"
# xdp_attach_mode = "generic"
# api_port = 8080
# pfcp_address = "10.0.0.1"
```

Problema: Nome da interface não encontrado

Sintomas:

```
ERRO[0000] interface eth0 não encontrada
```

Causa: interface configurada não existe

Resolução:

```
# Liste todas as interfaces de rede
ip link show

# Verifique o status da interface
ip addr show eth0

# Se a interface tiver um nome diferente, atualize
/etc/omniupf/runtime.exs:
# xdp_interfaces = "ens1f0"  # Use o nome real da interface

# Para VMs, verifique o esquema de nomenclatura da interface
ls /sys/class/net/
```

Problema: Porta já em uso

Sintomas:

```
ERRO[0000] falha ao iniciar o servidor API: endereço já em uso
```

Causa: Porta 8080, 8805 ou 9090 já vinculada por outro processo

Resolução:

```
# Encontre o processo usando a porta
sudo lsof -i :8080
sudo netstat -tulpn | grep :8080

# Mate o processo conflitante
sudo kill <PID>

# Ou altere a porta do OmniUPF em /etc/omniupf/runtime.exs
# api_port = 8081
# pfcp_port = 8806
```

Problema: ID de Nó PFCP inválido

Sintomas:

```
ERRO[0000] pfcp_node_id inválido: deve ser um endereço IPv4 válido
```

Causa: ID de Nó PFCP não é um endereço IPv4 válido

Resolução:

```
# Em /etc/omniupf/runtime.exs
# Correto: Use endereço IP (não nome de host)
node_id = "10.100.50.241"

# Incorreto:
# node_id = "localhost"
# node_id = "upf.example.com"
```

Problemas de Associação PFCP

Problema: Nenhuma associação PFCP estabelecida

Sintomas:

- A interface da Web mostra "Nenhuma associação"
- Logs do SMF mostram "Falha na configuração da associação PFCP"

Diagnóstico:

```
# 1. Verifique se o servidor PFCP está escutando
sudo netstat -ulpn | grep 8805

# 2. Verifique as regras do firewall
sudo iptables -L -n | grep 8805
sudo ufw status

# 3. Capture o tráfego PFCP
tcpdump -i any -n udp port 8805 -vv

# 4. Verifique as associações PFCP via API
curl http://localhost:8080/api/v1/upf_pipeline
```

Causas Comuns & Resoluções:

Firewall bloqueando PFCP

Resolução:

```
# Permita tráfego PFCP (UDP 8805)
sudo ufw allow 8805/udp
sudo iptables -A INPUT -p udp --dport 8805 -j ACCEPT
```

ID de Nó PFCP incorreto

Resolução:

```
# Em /etc/omniupf/runtime.exs, defina o ID do Nó para o IP correto
da interface N4
node_id = "10.100.50.241"    # Deve corresponder ao IP na rede N4
```

Rede inacessível ao SMF

Resolução:

```
# Teste a conectividade com o SMF
ping <SMF_IP>

# Verifique o roteamento para o SMF
ip route get <SMF_IP>

# Adicione rota se estiver faltando
sudo ip route add <SMF_NETWORK>/24 via <GATEWAY>
```

SMF configurado com IP do UPF incorreto

Resolução:

- Verifique a configuração do SMF para o endereço do UPF
- Certifique-se de que o SMF tenha o IP do `pfcp_node_id` do UPF configurado
- Verifique se o SMF pode rotear para a rede N4 do UPF

Problema: Falhas de heartbeat PFCP

Sintomas:

```
WARN[0030] tempo limite de heartbeat PFCP para associação
10.100.50.10
```

Diagnóstico:

```
# Verifique as estatísticas PFCP
curl http://localhost:8080/api/v1/upf_pipeline | jq
'.associations[] | {remote_id, uplink_teid_count}'

# Monitore os logs de heartbeat
journalctl -u omniupf -f | grep heartbeat
```

Causas & Resoluções:

Perda de pacotes na rede

Resolução:

```
# Verifique a perda de pacotes para o SMF
ping -c 100 <SMF_IP> | grep loss

# Se a perda for alta, investigue a rede:
# - Verifique o status do link
# - Verifique a saúde do switch/roteador
# - Verifique se há congestionamento
```

Intervalo de heartbeat muito agressivo

Resolução:

```
# Em /etc/omniupf/runtime.exs
heartbeat_interval_ms = 30_000    # Aumente de 5000 para 30000 ms
heartbeat_retries = 5             # Aumente as tentativas
heartbeat_timeout_ms = 10_000     # Aumente o tempo limite
```

Problemas de Processamento de Pacotes

Problema: Nenhum pacote fluindo (contadores RX/TX em 0)

Sintomas:

- A página de estatísticas mostra 0 pacotes RX/TX
- UE não consegue estabelecer sessão de dados

Diagnóstico:

```
# 1. Verifique se o programa XDP está anexado
ip link show eth0 | grep xdp

# 2. Verifique se a interface está ATIVA
ip link show eth0

# 3. Verifique as estatísticas de pacotes (ciente do XDP)
# Nota: tcpdump não pode ver pacotes GTP-U processados pelo XDP
curl http://localhost:8080/api/v1/packet_stats
```

Resoluções:

Programa XDP não anexado

Resolução:

```
# Reinicie o OmniUPF para re-anexar o XDP
sudo systemctl restart omniupf

# Verifique a anexação
ip link show eth0 | grep xdp
bpftool net list
```

Interface inativa ou sem link

Resolução:

```
# Ative a interface
sudo ip link set eth0 up

# Verifique o status do link
ethtool eth0 | grep "Link detected"

# Se o link estiver inativo, verifique a conexão física ou a
configuração de rede da VM
```

Interface configurada incorretamente

Resolução:

```
# Em /etc/omniupf/runtime.exs
xdp_interfaces = "ens1f0"    # Use o nome real da interface do 'ip
link show'
```

Problema: Pacotes recebidos mas não encaminhados (alta taxa de queda)

Sintomas:

- Contadores RX aumentando mas contadores TX não
- Taxa de queda > 1%

Diagnóstico:

```
# Verifique as estatísticas de queda
curl http://localhost:8080/api/v1/xdp_stats | jq '.drop'

# Verifique as estatísticas de rota
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'

# Monitore quedas de pacotes
watch -n 1 'curl -s http://localhost:8080/api/v1/packet_stats | jq
".total_rx, .total_tx, .total_drop"'
```

Causas Comuns:

Sem correspondência PDR (TEID ou IP do UE desconhecido)

Resolução:

```
# Verifique se as sessões existem
curl http://localhost:8080/api/v1/sessions

# Se não houver sessões, verifique:
# - A associação PFCP está estabelecida
# - O SMF criou sessões
# - O estabelecimento da sessão foi bem-sucedido

# Verifique as entradas do mapa PDR
bpftool map dump name pdr_map_teid_ip | grep -c key
bpftool map dump name pdr_map_downlin | grep -c key
```

Falhas de roteamento

Resolução:

```
# Verifique falhas de pesquisa FIB
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'

# Teste o roteamento para o IP do UE
ip route get 10.45.0.100

# Adicione a rota que está faltando
sudo ip route add 10.45.0.0/16 dev eth1  # Roteie o pool do UE
para N6
```

Limitação de taxa QER

Sintomas:

- Throughput menor que o esperado
- Tráfego limitado a uma taxa específica
- Contadores de volume URR mostram comportamento de platô
- Contadores de queda do XDP aumentando durante picos de tráfego

Diagnóstico:

1. **Verifique o MBR configurado para a sessão**:

```
# Encontre o ID do QER da sessão
curl http://localhost:8080/api/v1/pfcp_sessions | jq '.data[] |
select(.ue_ip == "10.45.0.1")'

# Procure a configuração do QER
curl http://localhost:8080/api/v1/qer_map | jq '.data[] |
select(.qer_id == 1)'
```

2. **Verifique o status do portão**:

```
# O status do portão deve ser 0 (ABERTO) para uplink e downlink
curl http://localhost:8080/api/v1/qer_map | jq '.data[] |
{qer_id, ul_gate: .ul_gate_status, dl_gate: .dl_gate_status}'
```

3. **Calcule o throughput real a partir do URR**:

```
# Consulte os contadores de volume URR em dois pontos no tempo
curl http://localhost:8080/api/v1/urr_map | jq '.data[] |
select(.urr_id == 0)'

# Calcule o throughput (manual):
# throughput_kbps = (volume_delta_bytes × 8) /
time_delta_seconds / 1000
```

4. **Compare MBR vs. throughput real**:
    - Throughput esperado ≈ 95-98% do MBR (devido à sobrecarga do protocolo)
    - Se o throughput estiver significativamente abaixo do MBR, verifique outros gargalos
    - Se o throughput corresponder exatamente ao MBR, a limitação de taxa está funcionando conforme o esperado

Resolução:

- **Se o MBR for muito baixo**: Solicite ao SMF para atualizar o QER com um MBR mais alto via Modificação de Sessão PFCP

- **Se o portão estiver fechado**: Investigue por que o SMF fechou o portão (política, cota ou erro)
- **Se a limitação de taxa for inesperada**: Verifique a configuração da política do SMF e o perfil de QoS

Entendendo a Aplicação do MBR:

O OmniUPF usa um algoritmo de janela deslizante para aplicar limites de MBR com precisão de nanosegundos no caminho de dados eBPF. Veja o Guia de Gerenciamento de Regras - Mecanismo de Aplicação de MBR para uma explicação detalhada sobre:

- Como o tamanho e a taxa dos pacotes determinam as decisões de queda
- Por que o throughput observado difere do MBR configurado
- Limitação de taxa por direção (uplink/downlink)
- Comportamento da janela deslizante de 5ms

Cenários Comuns:

- **Chamadas VoIP caindo**: Verifique se o MBR é suficiente para a taxa de bits do codec (G.711 = ~80 kbps)
- **Bufferização de streaming de vídeo**: Certifique-se de que o MBR > taxa de bits do vídeo + sobrecarga (1080p = ~5-10 Mbps)
- **Tráfego de pico**: Pequenos picos permitidos dentro da janela de 5ms, tráfego sustentado limitado pela taxa

---

Problema: Tráfego unidirecional (uplink funciona, downlink não)

Sintomas:

- Pacotes RX N3 mas nenhum pacote TX N3 (problema de downlink)
- Pacotes RX N6 mas nenhum pacote TX N6 (problema de uplink)

Diagnóstico:

```
# Verifique as estatísticas da interface N3/N6 (método ciente do
XDP)
curl http://localhost:8080/api/v1/n3n6_stats
curl http://localhost:8080/api/v1/packet_stats

# Nota: O tcpdump padrão não pode capturar tráfego GTP-U
processado pelo XDP
# Use a API de estatísticas ou xdpdump para análise de tráfego
# Veja a seção "Captura de Pacotes com XDP" para detalhes
```

Falha de Uplink (RX N3, sem TX N6):

Causa: Nenhuma ação FAR ou problema de roteamento para N6

Resolução:

```
# Verifique se o FAR tem ação FORWARD
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
select(.applied_action == 2)'

# Verifique se a rota N6 existe
ip route get 8.8.8.8  # Teste a rota para a internet

# Adicione a rota padrão se estiver faltando
sudo ip route add default via <N6_GATEWAY> dev eth1
```

Falha de Downlink (RX N6, sem TX N3):

Causa: Nenhum PDR de downlink ou encapsulamento GTP ausente

Resolução:

```
# Verifique se o PDR de downlink existe para o IP do UE
curl http://localhost:8080/api/v1/sessions | jq '.[].pdrs[] |
select(.pdi.ue_ip_address)'

# Verifique se o FAR tem CRIAÇÃO_DE_CABEÇALHO_EXTERNO
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
.outer_header_creation'

# Verifique a acessibilidade do gNB
ping <GNB_N3_IP>
```

Problemas de XDP e eBPF

Para configuração detalhada do XDP, seleção de modo e solução de problemas, veja o Guia de Modos XDP.

Problema: Programa XDP falhou ao carregar

Sintomas:

```
ERRO[0000] falha ao carregar o programa XDP: argumento inválido
```

Diagnóstico:

```
# Verifique o suporte XDP do kernel
grep XDP /boot/config-$(uname -r)

# Deve mostrar:
# CONFIG_XDP_SOCKETS=y
# CONFIG_BPF=y
# CONFIG_BPF_SYSCALL=y

# Verifique dmesg para erro detalhado
dmesg | grep -i bpf
```

Causas & Resoluções:

Kernel não possui suporte a XDP

Resolução:

```
# Recompile o kernel com suporte a XDP ou atualize para um kernel
mais novo
# Ubuntu 22.04+ tem XDP habilitado por padrão
sudo apt install linux-generic-hwe-22.04
sudo reboot
```

Falha de verificação do programa XDP

Resolução:

```
# Verifique os logs do OmniUPF em busca de erros do verificador
journalctl -u omniupf | grep verifier

# Problemas comuns:
# - Complexidade eBPF excede limites (aumente os limites do
kernel)
# - Acesso à memória inválido (bug no código eBPF)

# Aumente o nível de log do verificador eBPF para depuração
sudo sysctl kernel.bpf_stats_enabled=1
```

Problema: Contagem de abortos do XDP aumentando

Sintomas:

- Estatísticas do XDP mostram abortos > 0
- Aumentando quedas de pacotes

Diagnóstico:

```
# Verifique a contagem de abortos do XDP
curl http://localhost:8080/api/v1/xdp_stats | jq '.aborted'

# Monitore as estatísticas do XDP
watch -n 1 'curl -s http://localhost:8080/api/v1/xdp_stats'
```

Causa: programa eBPF encontrou erro em tempo de execução

Resolução:

```
# Verifique os logs do kernel em busca de erros eBPF
dmesg | grep -i bpf

# Reinicie o OmniUPF para recarregar o programa eBPF
sudo systemctl restart omniupf

# Se o problema persistir, ative o log eBPF (requer recompilação):
# Compile o OmniUPF com BPF_ENABLE_LOG=1
```

Problema: Mapa eBPF cheio (capacidade esgotada)

Sintomas:

- Estabelecimento de sessão falha
- Capacidade do mapa em 100%

Diagnóstico:

```
# Verifique a capacidade do mapa
curl http://localhost:8080/api/v1/map_info | jq '.[] | {map_name,
capacity, used, usage_percent}'

# Identifique mapas cheios
curl http://localhost:8080/api/v1/map_info | jq '.[] |
select(.usage_percent > 90)'
```

Mitigação Imediata:

```
# 1. Identifique sessões obsoletas
curl http://localhost:8080/api/v1/sessions | jq '.[] | {seid,
uplink_teid, created_at}'

# 2. Solicite ao SMF para excluir sessões antigas
# (via interface de administração do SMF ou API)

# 3. Monitore a diminuição do uso do mapa
watch -n 5 'curl -s http://localhost:8080/api/v1/map_info | jq ".
[] | select(.map_name==\"pdr_map_downlin\") | .usage_percent"'
```

Resolução a Longo Prazo:

```
# Em /etc/omniupf/runtime.exs
max_sessions = 200_000    # Aumente de 100_000

# Ou defina tamanhos individuais de mapa
far_map_size = 400_000
qer_map_size = 200_000
```

Importante: Alterar tamanhos de mapa requer reinício do OmniUPF e **limpa todas as sessões existentes**.

Problemas de Desempenho

Problema: Baixo throughput (abaixo do esperado)

Sintomas:

- Throughput < 1 Gbps apesar de NIC capaz
- Alta utilização da CPU

Diagnóstico:

```
# Verifique a taxa de pacotes
curl http://localhost:8080/api/v1/packet_stats | jq '.total_rx,
.total_tx'

# Verifique as estatísticas da NIC
ethtool -S eth0 | grep -i drop

# Verifique o modo XDP
ip link show eth0 | grep xdp
```

Resoluções:

Usando modo XDP genérico

Resolução:

```
# Em /etc/omniupf/runtime.exs
xdp_attach_mode = "native"    # Requer NIC/drivers compatíveis com
XDP
```

Gargalo de núcleo único

Resolução:

```
# Ative RSS (Receber Side Scaling) na NIC
ethtool -L eth0 combined 4  # Use 4 filas RX/TX

# Verifique se o RSS está habilitado
ethtool -l eth0

# Prenda interrupções a CPUs específicas
# Veja /proc/interrupts e use irqbalance ou afinidade manual
```

Buffer bloat

Resolução:

```
Os limites de buffer são gerenciados pelo plano de controle
Elixir.
Verifique a configuração do buffer em /etc/omniupf/runtime.exs.
```

Problema: Alta latência

Sintomas:

- Latência de ping > 50ms
- Degradação da experiência do usuário

Diagnóstico:

```
# Teste a latência para o UE
ping -c 100 <UE_IP> | grep avg

# Verifique pacotes em buffer
curl http://localhost:8080/api/v1/upf_buffer_info | jq
'.total_packets_buffered'

# Verifique o desempenho do cache de rota
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'
```

Resoluções:

Pacotes sendo excessivamente bufferizados

Resolução:

```
# Verifique por que os pacotes estão em buffer
curl http://localhost:8080/api/v1/upf_buffer_info | jq '.buffers[]
| {far_id, packet_count, direction}'

# Limpe os buffers se estiverem travados
# (reinicie o OmniUPF ou acione a modificação da sessão PFCP para
aplicar FAR)
```

Latência de pesquisa FIB

Resolução:

```
# Certifique-se de que o cache de rota esteja habilitado (opção de
tempo de compilação)
# Compile com BPF_ENABLE_ROUTE_CACHE=1

# Otimize a tabela de roteamento
# Use menos rotas, mais específicas em vez de muitas rotas
pequenas
```

Problemas Específicos do Hypervisor

Para instruções passo a passo de configuração do hypervisor, consulte o Guia de Modos XDP.

Proxmox: XDP não funcionando na VM

Sintomas:

- Não é possível anexar o programa XDP em modo nativo
- Apenas o modo genérico funciona

Causa: VM usando rede em ponte sem SR-IOV

Resolução:

Opção 1: Usar modo genérico (mais simples)

```
# Em /etc/omniupf/runtime.exs
xdp_attach_mode = "generic"
```

Opção 2: Configurar passthrough SR-IOV

```
# No host Proxmox:
# 1. Habilitar IOMMU
nano /etc/default/grub
# Adicionar: intel_iommu=on iommu=pt
update-grub
reboot

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

# 3. Atribuir VF à VM na UI do Proxmox
# Hardware → Adicionar → Dispositivo PCI → Selecionar VF
```

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

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

VMware: Modo promíscuo necessário

Sintomas:

- Pacotes não recebidos pelo OmniUPF

Causa: vSwitch bloqueando endereços MAC não correspondentes

Resolução:

```
# Habilitar modo promíscuo no vSwitch (no vSphere Client):
# 1. Selecionar vSwitch → Editar Configurações
# 2. Segurança → Modo Promíscuo: Aceitar
# 3. Segurança → Mudanças de Endereço MAC: Aceitar
# 4. Segurança → Transmissões Forjadas: Aceitar
```

VirtualBox: Desempenho muito baixo

Sintomas:

- Throughput < 100 Mbps

Causa: VirtualBox não suporta SR-IOV ou XDP nativo

Resolução:

```
# Em /etc/omniupf/runtime.exs
xdp_attach_mode = "generic"    # Única opção para VirtualBox
```

Otimize as configurações do VirtualBox:

- Use adaptador VirtIO-Net (se disponível)
- Habilite o modo promíscuo "Permitir Todos"
- Aloque mais núcleos de CPU para a VM
- Use rede em ponte em vez de NAT
- Considere migrar para KVM/Proxmox para melhor desempenho

Problemas de NIC e Driver

Problema: O driver da NIC não suporta XDP

Sintomas:

```
ERRO[0000] falha ao anexar programa XDP: operação não suportada
```

Diagnóstico:

```
# Verificar driver da NIC
ethtool -i eth0 | grep driver

# Verificar se o driver suporta XDP
modinfo <driver_name> | grep -i xdp

# Listar interfaces compatíveis com XDP
ip link show | grep -B 1 "xdpgeneric\|xdpdrv\|xdpoffload"
```

Resolução:

Opção 1: Usar modo genérico

```
# Em /etc/omniupf/runtime.exs
xdp_attach_mode = "generic"
```

Opção 2: Atualizar driver da NIC

```
# Verificar atualizações de driver (Ubuntu)
sudo apt update
sudo apt install linux-modules-extra-$(uname -r)

# Ou instalar driver específico do fornecedor
# Exemplo para Intel:
# Baixar de https://downloadcenter.intel.com/
```

Opção 3: Substituir NIC

```
# Usar NIC compatível com XDP:
# - Intel X710, E810
# - Mellanox ConnectX-5, ConnectX-6
# - Broadcom BCM57xxx (driver bnxt_en)
```

Problema: Driver falha ou causa pânico no kernel

Sintomas:

- Pânico no kernel após anexar XDP
- NIC para de responder

Diagnóstico:

```
# Verificar logs do kernel
dmesg | tail -100

# Verificar bugs do driver
journalctl -k | grep -E "BUG:|panic:"
```

Resolução:

```
# 1. Atualizar kernel e drivers
sudo apt update
sudo apt upgrade
sudo reboot

# 2. Em /etc/omniupf/runtime.exs, desabilitar offload XDP (usar
apenas nativo)
# xdp_attach_mode = "native"

# 3. Usar modo genérico como solução alternativa
# xdp_attach_mode = "generic"

# 4. Reportar bug ao fornecedor da NIC ou equipe do kernel Linux
```

Problema: Erros intermitentes de SLAAC IPv6 / checksum UDP (generic-XDP + virtio)

Sintomas:

```
# UEs IPv6 intermitentemente (~50%) nunca formam um endereço
global via SLAAC.
# O RA é emitido pelo UPF, mas nunca é visto pelo RAN.
# No eNB/gNB (ou qualquer host downstream) o contador de erro de
checksum UDP do kernel
# aumenta, um por RA descartado:
nstat -az | grep UdpInCsumErrors      # incrementa durante
tentativas de SLAAC com falha
```

- Afeta **apenas** SLAAC IPv6 / Anúncios de Roteador — dados do plano do usuário estão normais.
- Ocorre **apenas** quando um SGW-U e PGW-U de um bearer estão em **membros** de pool **diferentes** (dividido); um bearer SGW-U+PGW-U colapsado na mesma caixa sempre tem sucesso.

Causa: O Anúncio de Roteador é gerado em **espaço do usuário** (plano de controle), então seu buffer de socket é marcado para offload de checksum TX (`CHECKSUM_PARTIAL`; a NIC do datapath tem `tx-checksum-ip-generic: on`). Em um datapath **XDP em modo genérico**, o pacote encaminhado é um `skb` do kernel: quando um membro SGW-U de pool dividido retransmite esse RA, ele reescreve o IP externo + TEID e zera `udp->check`, mas o skb mantém o `csum_start`/`csum_offset` offload agora obsoleto. Na saída, a NIC (virtio) recomputa o checksum UDP sobre o pacote modificado a partir desse offset obsoleto, produzindo um checksum errado, e o receptor o descarta. O tráfego do plano do usuário XDP-nativo não é afetado porque é construído em eBPF com `CHECKSUM_NONE` (um checksum zero genuíno), então a NIC nunca o recomputa — é por isso que apenas o SLAAC falha. Modos XDP nativo/offload não enfrentam isso (sem skb, sem recomputação de offload); é específico para **modo genérico em uma NIC com offload de checksum (por exemplo, virtio)**.

Resolução:

```
# Desabilitar offload de checksum TX na(s) NIC(s) do datapath para
que o kernel compute o
# checksum UDP inline (corretamente) para pacotes de controle
originados em espaço do usuário. O XDP
# do plano de dados contorna o caminho de checksum do kernel,
então a taxa de encaminhamento não é afetada.
ethtool -K <n3_iface> tx off
ethtool -K <n6_iface> tx off

# Verificar (deve ler "off"):
ethtool -k <n3_iface> | awk '/^tx-checksumming/{print}'

# Re-testar: SLAAC deve agora ter sucesso para cada bearer e
UdpInCsumErrors deve permanecer estável.
```

> Torne isso persistente em sua implantação (por exemplo, na unidade de inicialização da NIC do datapath) para que sobreviva a reinicializações. Onde a NIC o suporta, o modo XDP nativo (`xdp_attach_mode: native`) também evita o problema completamente.

Falhas na Estabelecimento de Sessão

Problema: Falha na estabelecimento de sessão

Sintomas:

- SMF relata falha na estabelecimento de sessão
- UE não consegue estabelecer sessão PDU

Consulte o Referência de Códigos de Causa PFCP para cenários comuns de falha e resoluções.

Diagnóstico:

```
# Verificar logs do OmniUPF para erros de sessão
journalctl -u omniupf | grep -i "estabelecimento de sessão"

# Verificar contagem de sessões PFCP
curl http://localhost:8080/api/v1/sessions | jq 'length'

# Capturar tráfego PFCP durante o estabelecimento da sessão
tcpdump -i any -n udp port 8805 -w /tmp/pfcp_session.pcap
```

Causas Comuns:

Capacidade do mapa cheia

Resolução:

```
# Verificar uso do mapa
curl http://localhost:8080/api/v1/map_info | jq '.[] |
select(.usage_percent > 90)'

# Aumentar capacidade (veja a seção de mapa eBPF cheia acima)
```

Parâmetros PDR/FAR inválidos

Resolução:

```
# Verificar logs do OmniUPF para erros de validação
journalctl -u omniupf | grep -E "inválido|erro" | tail -20

# Problemas comuns:
# - Endereço IP UE inválido (0.0.0.0 ou duplicado)
# - TEID inválido (0 ou duplicado)
# - FAR ausente para PDR
# - Ação FAR inválida

# Verificar configuração do SMF e parâmetros da sessão
```

Recurso não suportado (UEIP/FTUP)

Resolução:

```
# Em /etc/omniupf/runtime.exs
feature_ueip = true
ueip_pool = "10.60.0.0/16"

feature_ftup = true
teid_pool_start = 1
teid_pool_end = 0xFFFFFFFF
```

Problemas de Bufferização

Problema: Pacotes presos no buffer

Sintomas:

- Contagem de pacotes em buffer aumentando
- Pacotes não entregues após a transferência

Diagnóstico:

```
# Verificar estatísticas do buffer
curl http://localhost:8080/api/v1/upf_buffer_info

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

# Monitorar tamanho do buffer
watch -n 5 'curl -s http://localhost:8080/api/v1/upf_buffer_info |
jq ".total_packets_buffered"'
```

Causas & Resoluções:

FAR nunca atualizado para FORWARD

Causa: SMF nunca enviou Modificação de Sessão PFCP para aplicar FAR

Resolução:

```
# Verificar status do FAR
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
{far_id, applied_action}'

# Ação BUFF = 1 (bufferização)
# Ação FORW = 2 (encaminhamento)

# Se preso no estado BUFF, solicitar ao SMF que:
# - Envie Solicitação de Modificação de Sessão PFCP
# - Atualize FAR com ação FORW
```

TTL do buffer expirado

Causa: Pacotes expiraram antes da atualização do FAR

Resolução: O TTL do buffer é gerenciado pelo plano de controle Elixir. Verifique a configuração do buffer em `/etc/omniupf/runtime.exs`.

Overflow do buffer

Causa: Muitos pacotes em buffer por FAR

Resolução: Os limites do buffer são gerenciados pelo plano de controle Elixir. Verifique a configuração do buffer em `/etc/omniupf/runtime.exs`.

Depuração Avançada

Habilitar Registro de Depuração

```
# Em /etc/omniupf/runtime.exs
log_level = :debug    # :debug | :info | :warning | :error
```

```
# Reiniciar OmniUPF com registro de depuração
sudo systemctl restart omniupf

# Monitorar logs em tempo real
journalctl -u omniupf -f --output cat
```

Rastreamento de Programa eBPF

```
# Rastrear execução de programa eBPF (requer bpftrace)
sudo bpftrace -e 'tracepoint:xdp:* { @[probe] = count(); }'

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

Captura de Pacotes com XDP

Entendendo as Limitações da Captura de Pacotes XDP:

XDP processa pacotes **antes** da pilha de rede do kernel, portanto, o `tcpdump` padrão **não consegue ver o tráfego processado pelo XDP**. Pacotes GTP-U (porta UDP 2152) na N3 são processados pelo XDP e não aparecerão no tcpdump no host UPF.

Métodos Recomendados para Análise de Tráfego:

```
# Método 1: Usar API de estatísticas para monitoramento
(RECOMENDADO)
curl http://localhost:8080/api/v1/xdp_stats
curl http://localhost:8080/api/v1/packet_stats | jq
curl http://localhost:8080/api/v1/n3n6_stats

# Método 2: Capturar tráfego PFCP (não afetado pelo XDP)
tcpdump -i any -n udp port 8805 -w /tmp/pfcp.pcap

# Método 3: Captura de pacotes fora de banda (RECOMENDADO para
GTP-U)
# Use TAP de rede ou espelhamento de porta de switch para capturar
tráfego
# Exemplos:
# - TAP físico entre gNB e UPF
# - Porta SPAN/mirror de switch copiando tráfego N3 para
analisador
# - Espelhamento de porta de switch virtual no hypervisor
#
# No host de captura (NÃO o UPF):
# tcpdump -i <mirror_interface> -n udp port 2152 -w
/tmp/n3_mirror.pcap
```

Exemplos de Configuração de Captura Fora de Banda:

Rede Física:

```
# Use um TAP de rede ou configure espelhamento de porta do switch
# Exemplo: configuração SPAN de switch Cisco
(config)# monitor session 1 source interface Gi1/0/1
(config)# monitor session 1 destination interface Gi1/0/24

# No host de monitoramento conectado ao Gi1/0/24:
tcpdump -i eth0 -n udp port 2152 -w /tmp/n3_capture.pcap
```

Ambiente Virtual (VMware, KVM, etc.):

```
# Configure espelhamento de porta do switch virtual para enviar
tráfego UPF para VM analisadora
# Exemplo: ponte Linux com tcpdump em VM diferente
# No hypervisor, espelhe a interface N3 do UPF para a interface do
analisador

# Na VM analisadora:
tcpdump -i eth1 -n udp port 2152 -w /tmp/n3_virtual.pcap
```

Por que é Necessário Captura Fora de Banda:

- O XDP contorna completamente a pilha de rede do kernel
- Pacotes são processados no driver da NIC ou hardware
- O tcpdump baseado em host vê pacotes DEPOIS do processamento XDP (tarde demais)
- A captura fora de banda vê o tráfego bruto antes do processamento do UPF

O que você PODE Capturar no Host UPF:

- □ Tráfego PFCP (UDP 8805) - plano de controle, não processado pelo XDP
- □ Respostas da API e métricas
- □ Tráfego GTP-U (UDP 2152) - plano de dados, processado pelo XDP

Obtendo Ajuda

Se os passos de solução de problemas não resolverem seu problema:

1. **Coletar informações de diagnóstico**:

```
# Informações do sistema
uname -a
cat /etc/os-release

# Informações do OmniUPF
curl http://localhost:8080/api/v1/upf_status
curl http://localhost:8080/api/v1/map_info
curl http://localhost:8080/api/v1/packet_stats

# Logs
journalctl -u omniupf --since "1 hour ago" > /tmp/omniupf.log
dmesg > /tmp/dmesg.log

# Informações de rede
ip addr > /tmp/network.txt
ip route >> /tmp/network.txt
ethtool eth0 >> /tmp/network.txt
```

2. **Reportar problema** com:
    - Versão do OmniUPF
    - Versão do kernel Linux
    - Diagrama de topologia de rede
    - Arquivo de configuração (redigir informações sensíveis)
    - Trechos de log relevantes
    - Passos para reproduzir

Documentação Relacionada

- **Guia de Configuração** - Parâmetros de configuração e exemplos
- **Guia de Arquitetura** - Internos e ajuste de desempenho do eBPF/XDP
- **Guia de Monitoramento** - Estatísticas, capacidade e alertas
- **Referência de Métricas** - Métricas Prometheus para solução de problemas
- **Códigos de Causa PFCP** - Códigos de erro PFCP e solução de problemas
- **Guia de Gerenciamento de Regras** - Conceitos de PDR, FAR, QER, URR

- **Guia de Operações** - Arquitetura e visão geral do UPF

OmniUPF Jardim Murado / Redirecionamento Fora de Crédito

Índice

1. Visão Geral
2. Arquitetura
3. Fluxo de Sinalização PFCP
4. Detecção de Portal Cativo
5. Configuração
6. Gerenciamento de Lista Branca
7. URLs de Redirecionamento por Sessão
8. API
9. Métricas Prometheus
10. Solução de Problemas

Visão Geral

O recurso Jardim Murado fornece **aplicação nativa de fora de crédito** diretamente no UPF, eliminando a necessidade de sistemas de aplicação externos (listas de endereços MikroTik, regras de mangle, DNAT).

Quando um assinante fica sem crédito, o SMF envia uma Modificação de Sessão PFCP com um FAR contendo `redirect_information`. O OmniUPF intercepta todo o tráfego dessa sessão no espaço do usuário e aplica uma experiência de portal cativo:

- **Consultas DNS** são falsificadas para retornar o IP do servidor do portal, acionando a detecção de portal cativo em todos os principais dispositivos (Apple, Android, Windows)
- **Tráfego para o portal e IPs na lista branca** (processadores de pagamento, serviços CAPTCHA) é encaminhado normalmente para que o assinante possa recarregar
- **Todo o outro tráfego** é silenciosamente descartado

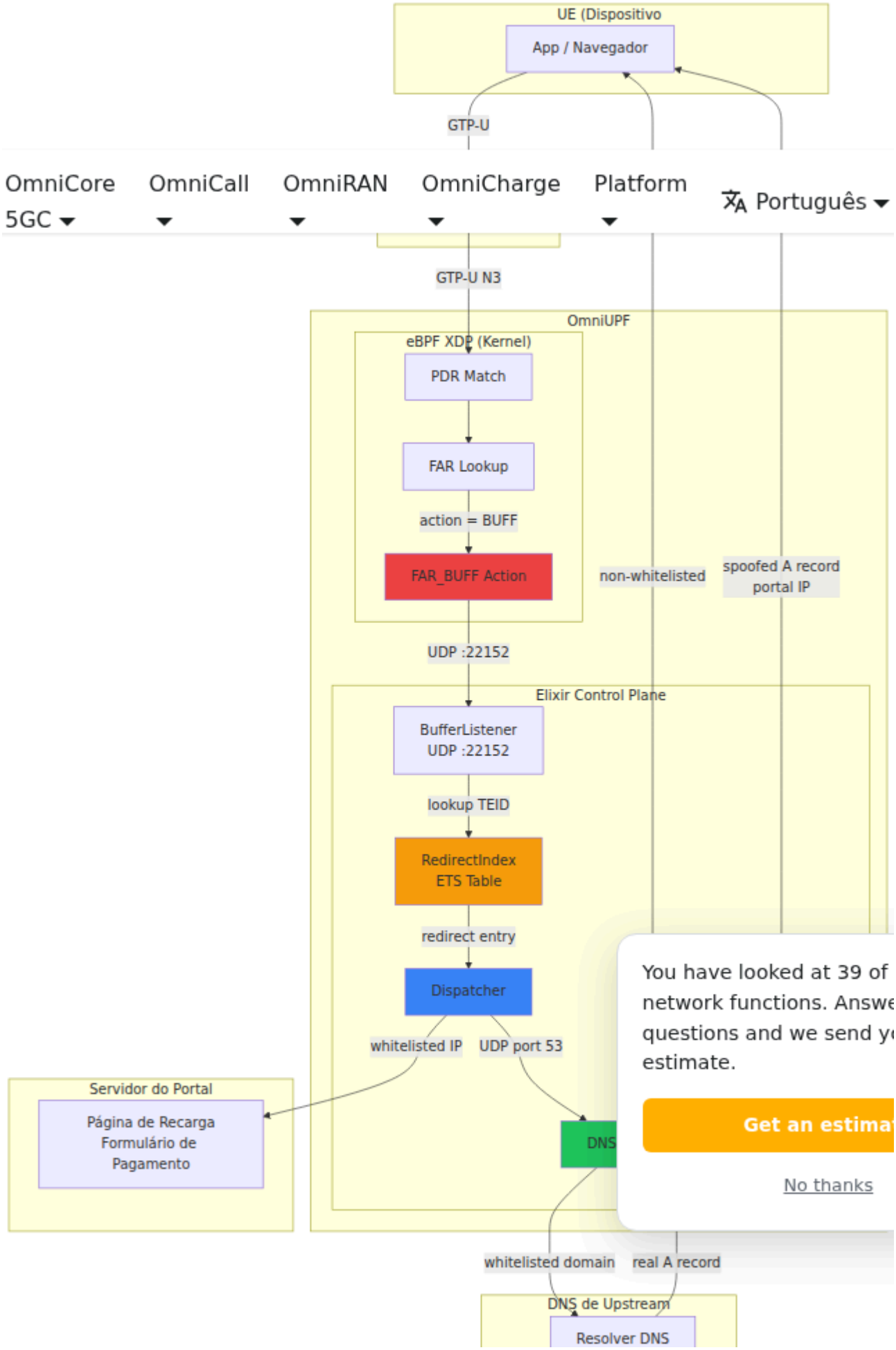
O assinante vê um prompt de portal cativo e é direcionado a uma página de recarga/pagamento. Assim que o crédito é restaurado, o SMF atualiza o FAR de volta para FORW e o encaminhamento normal é retomado imediatamente.

Pontos de Design Chave

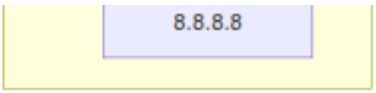
- **Nenhuma alteração no eBPF necessária** -- reutiliza a ação `FAR_BUFF` existente para redirecionar pacotes para o espaço do usuário
- **Redirecionamento por sessão** -- cada sessão pode ter um IP de portal e URL de redirecionamento diferentes, determinados pelo IE `redirect_information` do SMF
- **Lista branca vive no UPF** -- o SMF apenas diz "redirecionar esta sessão"; o UPF decide qual tráfego permitir
- **Intercepção apenas de uplink** -- o FAR de downlink permanece FORW para que as respostas do portal cheguem ao UE através do caminho de encapsulamento GTP normal

---

Arquitetura



You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Fluxo de Pacotes

1. **UE envia pacote de uplink** (consulta DNS, requisição HTTP, etc.) via GTP-U para o UPF
2. **Correspondência de PDR eBPF** encontra o PDR correspondente, procura o FAR
3. **Ação do FAR é BUFF** (substituída de FORW quando o redirecionamento está ativo) -- eBPF envia o pacote para o BufferListener na porta UDP 22152
4. **BufferListener** extrai o TEID, verifica a tabela ETS RedirectIndex
5. Se o TEID estiver no RedirectIndex: **Dispatcher** processa o pacote IP interno
6. **Árvore de decisão**:
   - Consulta DNS para domínio não listado: falsificar A record com IP do portal
   - Consulta DNS para domínio listado: encaminhar para o resolvedor real, armazenar IPs resolvidos
   - Tráfego para IP do portal ou IP listado: encaminhar via socket bruto
   - Todo o resto: descartar silenciosamente
7. **Respostas DNS/GTP-U** são enviadas de volta ao UE via o caminho de downlink GTP-U (encapsuladas com o TEID DL e enviadas para o gNB)

Tratamento de Loopback N9 (Sessão Dual SGW + PGW)

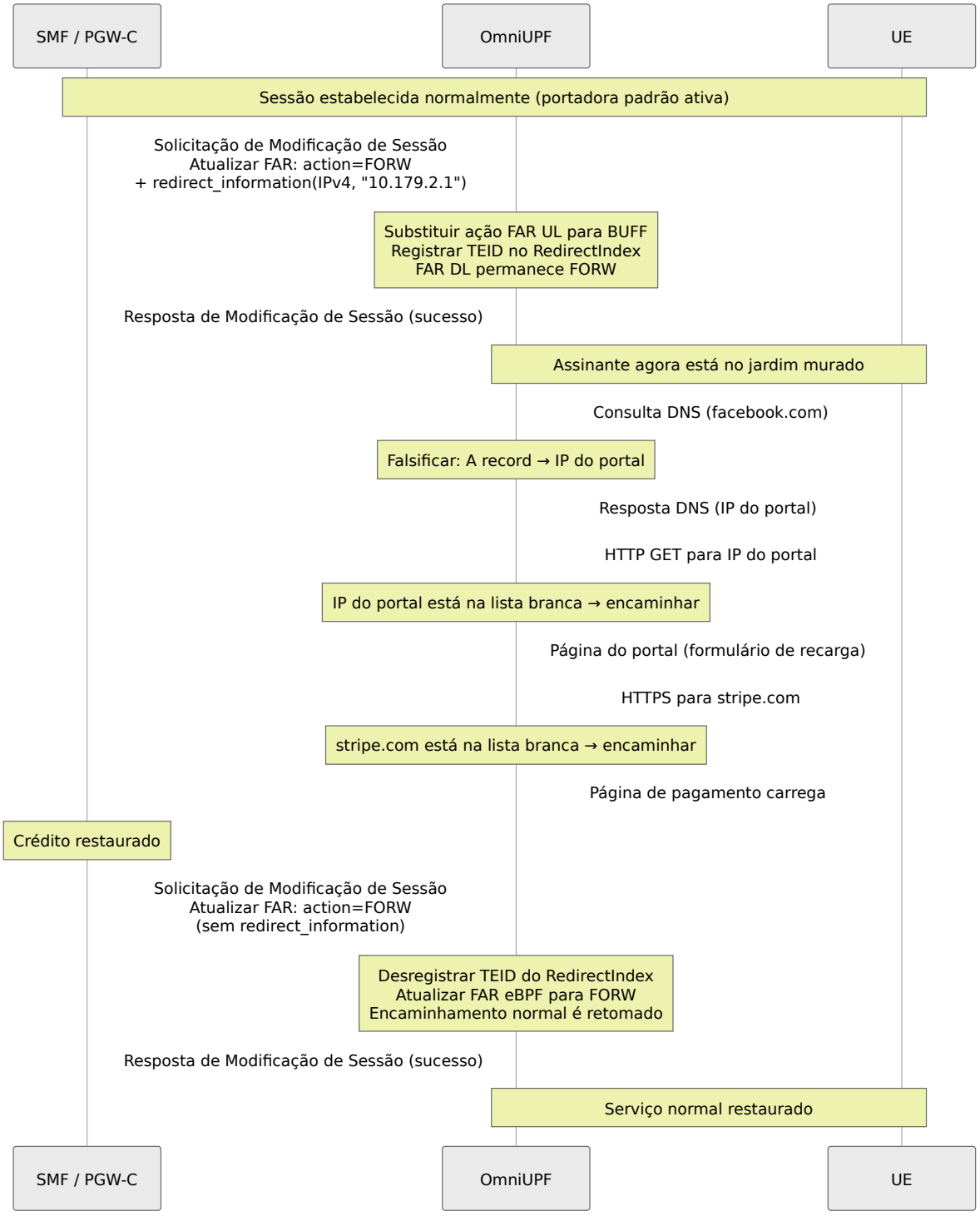
Em implantações EPC 4G, o OmniUPF frequentemente atua como **SGW-U e PGW-U simultaneamente** no mesmo nó. O SGW-C e o PGW-C estabelecem cada um uma sessão PFCP separada. A sessão SGW tem um FAR N9 que encaminha pacotes para o TEID de uplink da sessão PGW (um loopback através da interface N3). A sessão PGW realiza a aplicação real da política do assinante.

Ao ativar um redirecionamento de jardim murado via `POST /v1/walled_garden`, a lógica de ativação detecta essa topologia automaticamente:

1. **TEID(s) da sessão PGW** são identificados a partir dos PDRs de uplink do SEID alvo.
2. O código escaneia todas as sessões PFCP para encontrar uma sessão (a sessão SGW) que tenha um FAR cujo `teid` corresponda a um dos TEIDs de PDR UL do PGW e tenha `outer_header_creation` definido. Esse FAR é o FAR de encaminhamento N9 apontando para o PGW.
3. **Ambos** os FARs UL do PGW e os FARs UL voltados para o gNB do SGW são substituídos por `BUFF` no mapa eBPF. Isso é necessário porque o programa eBPF vê o pacote à medida que chega do eNB com o TEID do SGW, não com o TEID do PGW.
4. Apenas os **PDRs UL voltados para o gNB do SGW** são registrados no RedirectIndex (não os PDRs UL do PGW). O TEID do PDR UL do SGW é o que o eBPF verá no pacote de entrada do eNB.
5. Para o **caminho DL** (enviando respostas de volta ao UE), o Dispatcher usa o **FAR DL da sessão SGW** (o FAR que encaminha em direção ao eNB, com `remoteip != n3_address`). Esse FAR contém o TEID atual do eNB e o IP do gNB. O TEID DL é consultado ao vivo a partir do estado da sessão no momento da resposta, não armazenado no momento da ativação - isso lida com o caso em que o TEID do eNB é atribuído por uma Modificação de Sessão subsequente após a ativação do redirecionamento.

Em uma topologia UPF direta (sem SGW, SMF fala diretamente com UPF), apenas os próprios FARs e TEIDs UL da sessão estão envolvidos - o código de detecção de SGW não encontra nada e os PDRs UL da sessão principal são registrados diretamente.

Fluxo de Sinalização PFCP



IEs PFCP Envolvidos

O SMF aciona um redirecionamento incluindo `redirect_information` nos `forwarding_parameters` do FAR:

IE	Descrição
`apply_action`	Definido como FORW pelo SMF (UPF substitui internamente para BUFF)
`redirect_information`	Contém o tipo e o endereço do redirecionamento
`forwarding_parameters`	Contém o `redirect_information` e `outer_header_creation`

Tipos de Informações de Redirecionamento (por 3GPP TS 29.244 Tabela 8.2.20-1):

Tipo	Valor	Comportamento do UPF
IPv4	0	Use a string de endereço IPv4 fornecida como o IP do portal
IPv6	1	Use a string de endereço IPv6 fornecida como o IP do portal
URL	2	Resolva o nome do host da URL para um IP; use isso como o IP do portal. O caminho da URL não é usado pelo UPF -- ele é armazenado na API apenas para visibilidade. O servidor web do portal é responsável por lidar com qualquer roteamento baseado em caminho.
SIP URI	3	Não suportado atualmente

Detecção de Portal Cativo

O jardim murado aciona **detecção automática de portal cativo** em todas as principais plataformas de dispositivos ao falsificar respostas DNS. Quando um

dispositivo se conecta e tenta verificar a conectividade com a internet, ele consulta domínios bem conhecidos. A resposta DNS falsificada redireciona essas verificações para o IP do portal, que o dispositivo interpreta como um portal cativo.

Domínios de Detecção de Plataforma

Plataforma	Domínio de Detecção	Resposta Esperada
Apple (iOS/macOS)	`captive.apple.com`	HTTP 200 com `<HTML> <HEAD> <TITLE>Success</TITLE> </HEAD> <BODY>Success</BODY> </HTML>`
Android	`connectivitycheck.gstatic.com`	HTTP 204
Windows	`www.msftconnecttest.com`	HTTP 200 com `Microsoft Connect Test`
Samsung	`connectivitycheck.samsung.com`	HTTP 200

Quando esses domínios resolvem para o IP do portal em vez de seus endereços reais, o dispositivo detecta um portal cativo e apresenta a página do portal ao usuário como:

- Uma notificação/pop-up (iOS, Android)
- Um redirecionamento automático do navegador (Windows)

O servidor do portal no IP do portal configurado deve fornecer respostas apropriadas para essas URLs de detecção e, em seguida, redirecionar para a página de recarga/pagamento.

Requisitos do Servidor do Portal (IP Dedicado / vHost Padrão)

O portal deve servir a página do portal cativo como a *resposta padrão* de seu endereço IP, independentemente do cabeçalho HTTP `Host`. Na prática, isso significa dar ao portal seu **próprio IP dedicado**, ou torná-lo o **vhost virtual padrão** em um host compartilhado.

Esse requisito segue diretamente de como o jardim murado funciona:

1. Cada consulta DNS não listada é **falsificada para o IP do portal** -- incluindo as sondagens de portal cativo da plataforma (`connectivitycheck.gstatic.com`, `captive.apple.com`, `www.msftconnecttest.com`, ...).
2. O UE então abre HTTP para o IP do portal, mas a solicitação carrega o cabeçalho `Host` da **sondagem** (por exemplo, `Host: connectivitycheck.gstatic.com`), não o nome do host do portal.
3. O UPF encaminha essa solicitação para o servidor do portal **verbatim, sem reescrever o cabeçalho `Host`** (veja Encaminhamento de Socket Bruto).

Se o IP do portal for um **vhost virtual baseado em nome** -- ou seja, a página do portal é servida apenas para um valor específico de `Host:` e o IP é compartilhado com outros vhosts -- o servidor não encontrará um vhost correspondente ao cabeçalho `Host` da sondagem e servirá seu **vhost padrão**, que é a página errada. O assinante aterrissa em qualquer outra coisa que esteja hospedada nesse IP (outro aplicativo, um CRM, um site padrão em branco) em vez da página de recarga.

Faça isso:

- Dê ao portal cativo seu **próprio endereço IP** (qualquer `Host` servido por esse IP retorna a página do portal), **ou**
- Faça do portal o **vhost virtual padrão** no host compartilhado, para que qualquer `Host:` não correspondido caia na página do portal.

Evite: apontar `walled_garden_portal_ip` (ou um `redirect_information` do tipo URL cujo nome do host resolve) para um IP cuja resposta HTTP **padrão**

não é a página do portal. O UPF não realiza seleção de vhost baseada em `Host` ou reescrita de `Host` em nome do portal.

Configuração

A configuração do jardim murado reside no arquivo de configuração em tempo de execução. Em uma instalação de produção (`.deb` package), o arquivo de configuração está em `/etc/omniupf/runtime.exs`. O script de inicialização da release (`rel/env.sh`) verifica se este arquivo está presente e, se estiver, define `RELEASE_CONFIG_DIR=/etc/omniupf` para que a release Erlang o utilize em vez do `config/runtime.exs` incluído. O UPF deve ser reiniciado após alterações na configuração.

```
#
================================================================
# Jardim Murado / Redirecionamento Fora de Crédito
#
================================================================

# Habilitar aplicação de redirecionamento do jardim murado
walled_garden_enabled = true

# IP do servidor do portal (portal cativo / página de recarga)
walled_garden_portal_ip = "10.179.2.1"

# Endereço do servidor do portal IPv6 opcional (fallback para assinar
IPv6)
walled_garden_portal_ip6 = "fd00:179:2::1"

# Resolver DNS upstream para consultas de domínio na lista branca
walled_garden_dns_resolver = "8.8.8.8"

# Domínios na lista branca (os assinantes podem acessar esses enquant
redirecionados)
# Suporta curingas: "*.stripe.com" corresponde a "api.stripe.com"
walled_garden_whitelist = [
  "stripe.com",
  "*.stripe.com",
  "js.stripe.com",
  "hcaptcha.com",
  "*.hcaptcha.com",
  "newassets.hcaptcha.com",
]
```

Parâmetros

Parâmetro	Tipo	Obrigatório	Padrão	
`walled_garden_enabled`	Booleano	Não	`false`	In fu m `r` n
`walled_garden_portal_ip`	String (IPv4)	Sim (se habilitado)	`10.179.2.1`	E p E fa n `r` S re ti é IP IE Pa d m FA
`walled_garden_portal_ip6`	String (IPv6)	Não	`nil`	E o p as fc `w` é Q fa

Parâmetro	Tipo	Obrigatório	Padrão	
				re es
`walled_garden_dns_resolver`	String (IPv4)	Não	`8.8.8.8`	R us co na co lis pa to sã lo
`walled_garden_whitelist`	Lista de Strings	Não	`[]`	Pa os ac ne G B pa

Gerenciamento de Lista Branca

A lista branca controla quais domínios (e seus IPs resolvidos) os assinantes podem acessar enquanto redirecionados. Isso é configurado no UPF, não no SMF -- o SMF apenas aciona o redirecionamento.

Sintaxe de Padrão

Padrão	Corresponde	Não Corresponde
`stripe.com`	`stripe.com`, `api.stripe.com`, `js.stripe.com`	`evilstripe.com`, `notstripe.com`
`*.stripe.com`	`api.stripe.com`, `js.stripe.com`, `dashboard.stripe.com`	`stripe.com` (exato), `evilstripe.com`
`hcaptcha.com`	`hcaptcha.com`, `newassets.hcaptcha.com`	`evihcaptcha.com`

Os padrões usam **ancoragem de subdomínio**: `stripe.com` corresponde ao domínio em si e a qualquer subdomínio (`foo.stripe.com`), mas não a domínios que apenas contêm a string (`evilstripe.com`). A correspondência é insensível a maiúsculas e minúsculas.

Cache de IP

Quando o proxy DNS encaminha uma consulta para um domínio na lista branca, os endereços IP resolvidos são **automaticamente armazenados em cache** na lista branca. Isso significa:

1. O assinante consulta `api.stripe.com`
2. O proxy DNS encaminha para o resolvedor real, obtém `203.0.113.7`
3. `203.0.113.7` é adicionado ao cache de IPs na lista branca
4. O tráfego HTTP/HTTPS subsequente para `203.0.113.7` é encaminhado (não descartado)

O IP do portal é sempre listado independentemente da configuração.

Domínios Recomendados para a Lista Branca

Para um portal de recarga típico com processamento de pagamento Stripe e hCaptcha:

```
walled_garden_whitelist = [
  # Processador de pagamento
  "stripe.com",
  "*.stripe.com",

  # Serviço CAPTCHA
  "hcaptcha.com",
  "*.hcaptcha.com",

  # Google Fonts (se o portal os usar)
  "fonts.googleapis.com",
  "fonts.gstatic.com",

  # CDN para ativos do portal (se hospedados externamente)
  "cdn.example.com",
]
```

URLs de Redirecionamento por Sessão

Cada sessão PFCP pode ter um **alvo de redirecionamento diferente**. O SMF controla isso via o IE `redirect_information` no FAR:

- **Tipo IPv4**: A string de IP fornecida é analisada e usada como o IP do portal para essa sessão
- **Tipo IPv6**: A string IPv6 fornecida é analisada e usada como o IP do portal para essa sessão
- **Tipo URL**: O nome do host é extraído e resolvido via DNS no momento da criação do FAR. O IP resolvido é usado como o IP do portal. O caminho da URL não é usado pelo UPF -- ele é armazenado apenas para visibilidade da API.

Isso permite cenários onde diferentes níveis de assinantes, MVNOs ou planos de serviço redirecionam para diferentes portais:

Sessão	`redirect_information` do SMF	IP do Portal Usado	URL c
Sessão A	`IPv4: 10.179.2.1`	`10.179.2.1`	`10.179.2`
Sessão B	`IPv6: 2001:db8::1`	`2001:db8::1`	`2001:db8`
Sessão C	`URL: https://topup.mvno.com/recharge`	IP resolvido de `topup.mvno.com`	`https://`

O UPF armazena o IP do portal por sessão e a URL de redirecionamento, que são visíveis via a API.

API

O caminho base para todos os endpoints do jardim murado é `/v1/walled_garden` (sem prefixo `/api`). Esses endpoints são servidos pelo servidor HTTP Phoenix na `api_port` configurada (padrão: 8080).

GET /v1/walled_garden

Retorna todas as sessões de redirecionamento ativas do jardim murado, IPs na lista branca, intervalos CIDR em cache e detalhes da sessão.

Resposta:

```
{
  "redirect_count": 2,
  "redirects": [
    {
      "teid": "0x4000",
      "session_seid": 1,
      "portal_ip": "10.179.2.1",
      "redirect_url": null,
      "ue_ip": "10.60.0.1",
      "gnb_ip": "10.179.1.21",
      "dl_teid": "0x5000",
      "far_global_id": 42
    },
    {
      "teid": "0x4001",
      "session_seid": 2,
      "portal_ip": "10.179.2.2",
      "redirect_url": "https://topup.mvno.com",
      "ue_ip": "10.60.0.2",
      "gnb_ip": "10.179.1.21",
      "dl_teid": "0x5001",
      "far_global_id": 43
    }
  ],
  "whitelisted_ips": [
    {"ip": "10.179.2.1", "type": "portal"},
    {"ip": "203.0.113.7", "type": "resolved"},
    {"ip": "203.0.113.6", "type": "resolved"}
  ],
  "whitelisted_cidrs": ["192.168.0.0/24"]
}
```


Campos da Resposta:

Campo	Descrição
`redirect_count`	Número de sessões ativas do jardim murado
`redirects[].teid`	TEID de uplink sendo interceptado (hex)
`redirects[].session_seid`	SEID da sessão PFCP
`redirects[].portal_ip`	IP do portal para esta sessão específica
`redirects[].redirect_url`	URL de redirecionamento do SMF (se do tipo URL), ou nulo
`redirects[].ue_ip`	Endereço IP do UE
`redirects[].gnb_ip`	IP do gNB para respostas GTP-U
`redirects[].dl_teid`	TEID de downlink para encapsulamento GTP-U
`redirects[].far_global_id`	ID global interno do FAR
`whitelisted_ips`	Todos os IPs atualmente na lista branca (portal + resoluções DNS em cache)
`whitelisted_cidrs`	Intervalos CIDR adicionados via API da lista branca

POST /v1/walled_garden

Ativa o redirecionamento do jardim murado em uma sessão PFCP existente pelo SEID. Este é o caminho acionado pelo operador/API - o caminho normal é via Modificação de Sessão PFCP com `redirect_information` em um FAR. O caminho da API é útil para testes e para operadores que precisam redirecionar manualmente uma sessão sem a participação do SMF.

Corpo da solicitação:

```
{
  "seid": 1,
  "url": "http://10.179.2.1/"
}
```

Campo	Tipo	Obrigatório	Descrição
`seid`	Inteiro	Sim	SEID local da sessão PFCP a ser redirecionada
`url`	String	Sim	Alvo do redirecionamento - um endereço IPv4/IPv6 ou uma URL. Se uma URL for fornecida, o nome do host é resolvido e o IP resultante é usado como o IP do portal. A string completa da URL é armazenada para visibilidade da API.

Resposta (200 OK):

```
{
  "status": "redirect activated",
  "info": {
    "seid": 1,
    "sgw_seid": 7,
    "portal_ip": "10.179.2.1",
    "ue_ip": "10.60.0.1",
    "gnb_ip": "10.179.1.21",
    "ul_teids": ["0x4000", "0x4006"]
  }
}
```

O campo `sgw_seid` é não nulo quando um loopback N9 (sessão emparelhada SGW+PGW) foi detectado e seu TEID de uplink também foi BUFF'd. `ul_teids`

lista todos os TEIDs de uplink que foram registrados no índice de redirecionamento.

Respostas de erro:

Status	Significado
404	Sessão não encontrada (SEID não corresponde a nenhuma sessão PFCP ativa)
400	Parâmetros obrigatórios ausentes

DELETE /v1/walled_garden/:seid

Desativa o redirecionamento do jardim murado para uma sessão, restaurando o encaminhamento normal. Todos os FARs de uplink para a sessão (incluindo FARs de sessão SGW emparelhados em uma topologia de loopback N9) são definidos de volta para `action=FORW` no mapa eBPF, e os TEIDs são desregistrados do índice de redirecionamento.

Parâmetro de caminho: `:seid` - o SEID local da sessão a ser desativada.

Resposta (200 OK):

```
{"status": "redirect removed", "info": {"seid": 1}}
```

Respostas de erro:

Status	Significado
404	Sessão não encontrada

GET /v1/walled_garden/whitelist

Retorna a lista branca atual: IPs individuais em cache (IP do portal e IPs resolvidos por DNS) e quaisquer intervalos CIDR adicionados via API.

Resposta:

```
{
  "ips": [
    {"ip": "10.179.2.1", "type": "portal"},
    {"ip": "203.0.113.7", "type": "resolved"}
  ],
  "cidrs": ["192.168.100.0/24"]
}
```

POST /v1/walled_garden/whitelist

Adiciona um endereço IP ou intervalo CIDR à lista branca em tempo de execução, sem reiniciar o UPF. As alterações são apenas em memória e não persistem entre reinicializações - adicione entradas permanentes a `walled_garden_whitelist` em `runtime.exs`.

Corpo da solicitação (adicionar um IP):

```
{"ip": "203.0.113.10"}
```

Corpo da solicitação (adicionar um intervalo CIDR):

```
{"cidr": "192.168.100.0/24"}
```

Exatamente um de `ip` ou `cidr` deve estar presente. Adicionar um intervalo CIDR significa que qualquer tráfego para IPs dentro desse intervalo é encaminhado pelo dispatcher sem a necessidade de uma entrada de cache DNS por IP.

Resposta (200 OK - IP adicionado):

```
{"status": "added", "ip": "203.0.113.10"}
```

Resposta (200 OK - CIDR adicionado):

```
{"status": "added", "cidr": "192.168.100.0/24"}
```

Respostas de erro:

Status	Significado
400	IP inválido, CIDR inválido ou campos do corpo ausentes

Métricas Prometheus

Medidores

Métrica: `upf_walled_garden_active_redirects` **Tipo**: Medidor **Descrição**: Número de sessões atualmente no estado de redirecionamento do jardim murado

Consultas de exemplo:

```
# Contagem atual de redirecionamentos
upf_walled_garden_active_redirects
```

Contadores

Métrica: `upf_walled_garden_packets_intercepted_total` **Tipo**: Contador **Descrição**: Total de pacotes interceptados pelo jardim murado (todo o tráfego de uplink de sessões redirecionadas)

Métrica: `upf_walled_garden_packets_dropped_total` **Tipo**: Contador **Descrição**: Total de pacotes descartados pelo jardim murado (tráfego não listado, não-DNS)

Métrica: `upf_walled_garden_packets_forwarded_total` **Tipo**: Contador **Rótulos**:

- `dst_ip` - IP de destino do pacote encaminhado **Descrição**: Pacotes encaminhados através do jardim murado para destinos na lista branca. Rotulado pelo IP de destino para visibilidade por destino.

Métrica: `upf_walled_garden_bytes_forwarded_total` **Tipo**: Contador **Rótulos**:

- `dst_ip` - IP de destino do tráfego encaminhado **Descrição**: Bytes encaminhados através do jardim murado por IP de destino. Use isso para identificar quais serviços na lista branca os assinantes acessam enquanto redirecionados.

Métrica: `upf_walled_garden_dns_spoofed_total` **Tipo**: Contador **Rótulos**:

- `domain` - O domínio que foi falsificado **Descrição**: Consultas DNS falsificadas pelo jardim murado. Rotulado pelo domínio consultado.

Métrica: `upf_walled_garden_dns_forwarded_total` **Tipo**: Contador **Rótulos**:

- `domain` - O domínio listado que foi encaminhado **Descrição**: Consultas DNS encaminhadas para o resolvedor real (domínios listados). Rotulado pelo domínio.

Consultas de Exemplo

```
# Sessões ativas do jardim murado
upf_walled_garden_active_redirects

# Taxa de intercepção (pacotes/segundo)
rate(upf_walled_garden_packets_intercepted_total[5m])

# Taxa de descarte (deve ser a maioria dos interceptados)
rate(upf_walled_garden_packets_dropped_total[5m])

# Tráfego encaminhado por destino (bytes/segundo)
sum by (dst_ip)
(rate(upf_walled_garden_bytes_forwarded_total[5m]))

# Top 5 destinos listados por volume de tráfego
topk(5, sum by (dst_ip)
(rate(upf_walled_garden_bytes_forwarded_total[5m])))

# Domínios falsificados mais consultados
topk(10, sum by (domain)
(rate(upf_walled_garden_dns_spoofed_total[5m])))

# Taxa de consulta de domínios listados
sum by (domain) (rate(upf_walled_garden_dns_forwarded_total[5m]))

# Proporção de descartados vs encaminhados
sum(rate(upf_walled_garden_packets_dropped_total[5m]))
  / sum(rate(upf_walled_garden_packets_intercepted_total[5m]))
```

Solução de Problemas

Portal Cativo Não Aparecendo no Dispositivo

Sintomas: O assinante é redirecionado (visível na API), mas o dispositivo não mostra o prompt do portal cativo.

Possíveis causas:

- Servidor do portal não respondendo no IP do portal configurado
- Servidor do portal não lidando com URLs de detecção específicas da plataforma
- Resposta DNS não chegando ao UE (verifique o caminho GTP-U)

Resolução:

1. Verifique se o servidor do portal é acessível: `curl http://<portal_ip>/`
2. Confirme se o portal lida com URLs de detecção (por exemplo, `GET /hotspot-detect.html` para Apple)
3. Verifique `GET /v1/walled_garden` para confirmar se a sessão está registrada
4. Verifique as métricas Prometheus: `upf_walled_garden_dns_spoofed_total` deve estar aumentando
5. Verifique se o caminho GTP-U de downlink está funcionando (FAR DL deve permanecer FORW)

Página de Pagamento Não Carregando

Sintomas: O assinante vê o portal cativo, mas não consegue acessar a página de pagamento (Stripe, etc.).

Possíveis causas:

- Domínio do processador de pagamento não está na lista branca
- Processador de pagamento usando um IP de CDN que não foi armazenado em cache
- Padrão da lista branca não correspondendo corretamente a subdomínios

Resolução:

1. Verifique `GET /v1/walled_garden` - verifique se `whitelisted_ips` inclui os IPs do processador de pagamento
2. Verifique Prometheus: `upf_walled_garden_dns_forwarded_total{domain="stripe.com"}` deve mostrar consultas

3. Adicione domínios ausentes à lista branca (comum: `js.stripe.com`, `m.stripe.network`)
4. Verifique `upf_walled_garden_bytes_forwarded_total` por `dst_ip` para ver qual tráfego está realmente fluindo

Redirecionamento Não Ativando

Sintomas: SMF envia Modificação de Sessão com `redirect_information`, mas o tráfego do assinante não está sendo interceptado.

Possíveis causas:

- `walled_garden_enabled` está `false`
- O IE `redirect_information` está no FAR errado (deve estar no FAR **de uplink**)
- A ação do FAR não está sendo substituída para BUFF

Resolução:

1. Verifique a configuração: `walled_garden_enabled = true`
2. Verifique `GET /v1/pfcp_sessions` - veja a ação do FAR para a sessão. O FAR de uplink deve mostrar ação `0x04` (BUFF)
3. Verifique `GET /v1/walled_garden` - o TEID da sessão deve aparecer na lista de redirecionamento
4. Verifique os logs do UPF para mensagens `redirect_info` durante a modificação da sessão

Redirecionamento Não Limpando Após Recarregar

Sintomas: O assinante recarregou, mas o tráfego ainda está sendo interceptado.

Possíveis causas:

- O SMF não enviou a Modificação de Sessão para remover o redirecionamento

- Transição de BUFF->FORW não está sendo detectada

Resolução:

1. Verifique `GET /v1/walled_garden` - a sessão ainda está listada?
2. Verifique `GET /v1/pfcp_sessions` - verifique se a ação do FAR foi atualizada de volta para `0x02` (FORW)
3. Verifique os logs do SMF para confirmar que ele enviou a Modificação de Sessão
4. Verifique os logs do UPF para a mensagem "redirect removed, unregistering walled garden"

TEID de Downlink Errado (Obsoleto Após Modificação de Sessão)

Sintomas: Respostas DNS falsificadas ou tráfego encaminhado não estão chegando ao UE; pacotes GTP-U são enviados para o IP correto do gNB, mas o eNB os descarta como TEID desconhecido.

Causa: Em topologias 4G/SGW+PGW, o TEID do eNB no FAR DL do SGW pode ser zero ou um TEID de espaço reservado no momento em que o redirecionamento do jardim murado é ativado (por exemplo, a troca de `Initial Context Setup` acontece após o Estabelecimento da Sessão). Quando o eNB aloca seu TEID e o envia de volta via Modificação de Sessão, o FAR DL na sessão SGW é atualizado - mas se o Dispatcher tiver armazenado em cache o valor antigo, ele usará o TEID errado.

Resolução: O Dispatcher resolve o TEID DL **ao vivo** no momento da resposta procurando o FAR DL atual da sessão SGW (o FAR com `outer_header_creation` definido e `remoteip != n3_address`). Ele só recorre ao valor armazenado no momento da ativação se a consulta da sessão falhar. Portanto, um TEID obsoleto deve se corrigir assim que a próxima resposta DNS ou encaminhada for enviada. Se você ainda estiver vendo o TEID errado:

1. Verifique `GET /v1/walled_garden` - verifique se `dl_teid` na entrada de redirecionamento parece plausível (não zero).
2. Verifique `GET /v1/pfcp_sessions` - veja as entradas FAR da sessão SGW; o FAR voltado para o gNB deve ter o `teid` e `remoteip` atuais.

3. Se a sessão SGW foi excluída (por exemplo, transferência ou liberação), a entrada de redirecionamento permanecerá no RedirectIndex, mas a consulta ao vivo falhará. Nesse caso, use `DELETE /v1/walled_garden/:seid` para limpar a entrada obsoleta e permitir que o SMF reestabeleça.

Portal Retorna 304 Não Modificado

Sintomas: O navegador do assinante mostra uma página em branco ou a página do portal cativo carrega uma vez, mas visitas subsequentes aparecem vazias.

Causa: Respostas HTTP 304 (Não Modificado) são enviadas quando o navegador tem uma versão em cache da página e o servidor confirma que nada mudou. Para fluxos de portal cativo, alguns servidores web de portal enviam 304 em resposta às solicitações de detecção da plataforma (`/hotspot-detect.html`, `/generate_204`, etc.) se o navegador enviar cabeçalhos `If-Modified-Since` ou `If-None-Match`. Algumas implementações de portal cativo também enviam 304 para a própria página de redirecionamento.

Resolução:

1. O UPF encaminha respostas HTTP do servidor do portal de forma transparente - não modifica códigos de resposta. O problema está na configuração do servidor web do portal.
2. No servidor do portal, certifique-se de que os endpoints de detecção da plataforma retornem o código de status correto (200, não 304) com o conteúdo do corpo esperado (veja Detecção de Portal Cativo).
3. Configure o servidor do portal para enviar `Cache-Control: no-store` e omitir cabeçalhos `ETag`/`Last-Modified` nos endpoints de detecção para que os navegadores não os armazenem em cache.
4. Verifique com: `curl -v -H "If-None-Match: foo" http://<portal_ip>/hotspot-detect.html` - a resposta deve ser 200, não 304.

Portal Mostra a Página Errada (Vhost Virtual Baseado em Nome)

Sintomas: O assinante é redirecionado (visível em `GET /v1/walled_garden`) e o prompt do portal cativo aparece, mas ao clicar nele, abre o **site errado** -- um aplicativo diferente, um CRM ou uma página padrão genérica -- em vez da página de recarga/cadastro. `curl` confirma que a página do portal só carrega com um cabeçalho `Host:` específico:

```
curl -s -H 'Host: portal.example'
http://<portal_ip>/   # -> página cativa/cadastro   (correto)
curl -s -H 'Host: connectivitycheck.gstatic.com'
http://<portal_ip>/   # -> vhost Padrão (errado)
curl -s -H 'Host: <portal_ip>'
http://<portal_ip>/   # -> vhost Padrão (errado)
```

Causa: O IP do portal é um **vhost virtual baseado em nome** compartilhado com outros sites. O jardim murado falsifica cada sondagem de portal cativo para o IP do portal e encaminha a solicitação HTTP do UE **verbatim, mantendo o cabeçalho `Host` original da sondagem** (por exemplo, `Host: connectivitycheck.gstatic.com`). O servidor web do portal não tem vhost correspondente a esse `Host`, então serve seu **vhost padrão** -- a página errada. O UPF não reescreve o cabeçalho `Host` ou faz seleção de vhost em nome do portal. Veja Requisitos do Servidor do Portal.

Resolução:

1. Dê ao portal cativo seu **próprio IP dedicado** para que qualquer `Host:` servido por esse IP retorne a página do portal, **ou**
2. Faça do portal o **vhost virtual padrão** no host compartilhado, para que valores `Host:` não correspondidos caiam na página do portal.
3. Verifique com os comandos `curl` acima: com um IP dedicado ou vhost padrão, **todos os três** pedidos devem retornar a página do portal.

Encaminhamento de Socket Bruto

Quando o Dispatcher encaminha um pacote para um IP listado, ele usa um **socket bruto** (`IPPROTO_RAW`, número do protocolo 255) através do módulo `:socket` do Erlang. Um novo socket bruto é aberto por pacote, o pacote IP interno completo (como recebido do eBPF) é enviado com `sendto`, e o socket é imediatamente fechado.

Como isso funciona: O pacote IP interno do payload GTP-U já tem um cabeçalho IP válido com o IP do UE como origem e o IP do servidor de destino como destino. Ao injetar esse pacote via um socket bruto usando `IPPROTO_RAW`, o kernel o roteia com base no IP de destino usando a tabela de roteamento do host. A interface N6 do UPF deve ter uma rota para o servidor do portal/listado para que isso funcione.

Problemas Comuns:

Sintoma	Causa Provável	Correção
Encaminhamento falha silenciosamente	`EPERM` - socket bruto requer root ou `CAP_NET_RAW`	Certifique-se de que o OmniUPF seja executado como root ou tenha a capacidade `CAP_NET_RAW`
Pacotes encaminhados, mas nenhuma resposta chega ao UE	Rota N6 para o portal ausente	Adicione rota para a sub-rede do portal no host do UPF
`Walled garden: raw socket open failed` nos logs	Capacidade ausente ou restrição do kernel	Verifique o serviço `systemd` `AmbientCapabilities=CAP_NET_RAW`
Encaminhamento funciona, mas UE recebe IP errado como origem	NAT no N6 reescreve a origem	Certifique-se de que as respostas do portal para o IP do UE sejam roteadas de volta através do UPF

Verifique os logs do UPF para mensagens `Walled garden forward failed` e `raw socket open failed`. Use as métricas Prometheus `upf_walled_garden_packets_forwarded_total` e `upf_walled_garden_bytes_forwarded_total` para confirmar que o tráfego está fluindo.

Aviso de Alta Cardinalidade para Prometheus

Nota: Os rótulos `dst_ip` e `domain` nas métricas do jardim murado podem produzir alta cardinalidade se muitos destinos ou domínios únicos forem

consultados. Em grandes implantações, considere usar regras de gravação para agregar essas métricas:

```
# Regra de gravação para agregar por sub-rede /24 em vez de IPs
individuais
sum by (dst_subnet) (
  label_replace(
    rate(upf_walled_garden_bytes_forwarded_total[5m]),
    "dst_subnet", "$1.0/24", "dst_ip", "(\d+\.\d+\.\d+)\.\d+"
  )
)
```

Guia de Operações da Interface Web

Índice

1. Visão Geral
2. Acessando o Painel de Controle
3. Visualização de Sessões
4. Gerenciamento de Regras
5. Gerenciamento de Buffers
6. Painel de Estatísticas
7. Monitoramento de Capacidade
8. Visualização de Configuração
9. Visualização de Rotas
10. Visualização de Capacidades XDP
11. Visualizador de Logs

Visão Geral

A Interface Web OmniUPF fornece um painel de controle abrangente para monitoramento e gerenciamento em tempo real da Função do Plano do Usuário. A interface é construída sobre o Phoenix LiveView e fornece:

- **Visibilidade em tempo real** nas sessões PFCP e conexões PDU ativas
- **Inspeção de regras** para PDR, FAR, QER e URR em todas as sessões
- **Gerenciamento de buffers** para armazenamento de pacotes durante eventos de mobilidade
- **Monitoramento de estatísticas** para processamento de pacotes, rotas e interfaces
- **Rastreamento de capacidade** para uso e limites de mapas eBPF
- **Visualização de logs ao vivo** para solução de problemas

Arquitetura

O painel de controle se comunica com várias instâncias do OmniUPF através de sua API REST para:

- Consultar sessões e associações PFCP
- Inspecionar regras de detecção e encaminhamento de pacotes
- Monitorar buffers de pacotes e seu status
- Acessar estatísticas e métricas de desempenho em tempo real
- Rastrear capacidade e utilização de mapas eBPF

Acessando o Painel de Controle

Acesso Padrão

O painel de controle é acessível via HTTPS no servidor de gerenciamento do OmniUPF:

```
https://<upf-server>:443/
```

Porta Padrão: 443 (HTTPS com certificado autoassinado)

Configuração

O painel de controle requer configuração do host OmniUPF em `config/config.exs`:

Múltiplas instâncias do UPF podem ser configuradas para implantações de múltiplas instâncias:

A configuração `upf_hosts` define quais instâncias do OmniUPF estão disponíveis no menu suspenso do seletor de host em toda a interface.

Navegação

O painel de controle fornece abas de navegação para cada área operacional:

- **Sess��es** - `/sessions` - Sessões e associações PFCP
- **Regras** - `/rules` - Inspeção de regras PDR, FAR, QER, URR
- **Buffers** - `/buffers` - Monitoramento e controle de buffers de pacotes
- **Estatísticas** - `/statistics` - Estatísticas de pacotes, rotas, XDP e interfaces
- **Capacidade** - `/capacity` - Uso e monitoramento de capacidade de mapas eBPF
- **Config** - `/upf_config` - Configuração do UPF e endereços do dataplane
- **Rotas** - `/routes` - Rotas de UE e sessões de protocolo de roteamento (OSPF, BGP)
- **Capacidades XDP** - `/xdp_capabilities` - Suporte ao modo XDP e capacidades de desempenho
- **Logs** - `/logs` - Streaming de logs ao vivo

Visualização de Sessões

URL: `/sessions`

Recursos

A visualização de Sessões exibe todas as sessões PFCP ativas e associações das instâncias do OmniUPF selecionadas.

Resumo de Associações PFCP

Exibe todas as associações PFCP ativas (conexões de controle do SMF/PGW-C):

Coluna	Descrição
Node ID	Identificador do nó SMF ou PGW-C (FQDN ou IP)
Endereço	Endereço IP do SMF/PGW-C para comunicação PFCP
Próximo ID de Sessão	Próximo ID de sessão PFCP disponível para esta associação

Propósito:

- Verificar a conectividade do SMF ao UPF
- Monitorar o número de conexões do plano de controle
- Rastrear a alocação de ID de sessão por associação

Tabela de Sessões Ativas

Exibe todas as sessões PFCP representando sessões PDU ativas de UE:

Coluna	Descrição
Local SEID	Identificador de ponto final de sessão atribuído pelo UPF
Remote SEID	Identificador de ponto final de sessão atribuído pelo SMF
UE IP	Endereço IPv4 ou IPv6 do equipamento do usuário
TEID	Identificador de Ponto de Túnel GTP-U para tráfego de uplink
PDRs	Número de regras de detecção de pacotes na sessão
FARs	Número de regras de ação de encaminhamento na sessão
QERs	Número de regras de aplicação de QoS na sessão
URRs	Número de regras de relatório de uso na sessão
Ações	Botão de expandir para visualizar informações detalhadas da regra

Recursos:

- **Filtrar por IP**: Encontrar sessões para um endereço IP de UE específico
- **Filtrar por TEID**: Encontrar sessões por ID de ponto de túnel
- **Expandir sessão**: Ver detalhes completos em JSON de PDR/FAR/QER/URR
- **Atualização automática**: Atualiza a cada 10 segundos

Visualização de Sessão Expandida:

Quando você clica em "Expandir" em uma sessão, a visualização mostra:

- **Regras de Detecção de Pacotes (PDRs)**: JSON completo com TEID, UE IP, FAR ID, QER ID, filtros SDF

    - **IDs de PDR são clicáveis** - Clique para navegar até a aba Regras e visualizar detalhes completos do PDR
    - PDRs de uplink (TEID ≠ 0) link para consulta de PDR de uplink
    - PDRs de downlink (IPv4) link para consulta de PDR de downlink
    - PDRs de downlink (IPv6) link para consulta de PDR de downlink IPv6
- **Regras de Ação de Encaminhamento (FARs)**: Flags de ação, criação de cabeçalho externo, pontos finais de destino
- **Regras de Aplicação de QoS (QERs)**: MBR, GBR, QFI e outros parâmetros de QoS
- **Regras de Relatório de Uso (URRs)**: Contadores de volume (uplink, downlink, total de bytes)

Visualização expandida de sessão mostrando PDRs, FARs e QERs detalhados para uma sessão específica.

Casos de Uso

Verificar Conectividade de UE:

1. Navegue até a visualização de Sessões
2. Insira o endereço IP da UE no filtro
3. Confirme que a sessão existe com o TEID correto
4. Expanda para verificar a configuração de PDR/FAR

Monitorar Contagem de Sessões:

- Verifique a contagem total de sessões no cabeçalho
- Compare entre várias instâncias do UPF
- Rastreie o crescimento de sessões ao longo do tempo

Solução de Problemas de Sessões:

- Pesquise por um IP de UE ou TEID específico
- Expanda a sessão para inspecionar a configuração da regra
- Verifique os parâmetros de encaminhamento do FAR
- Verifique as configurações de QoS do QER

Atualizações em Tempo Real

A visualização de Sessões atualiza automaticamente a cada 10 segundos. Um indicador de verificação de saúde mostra o status de conectividade do UPF:

- **SAUDÁVEL** (verde): UPF é acessível e respondendo
- **NÃO SAUDÁVEL** (vermelho): UPF não é acessível ou não está respondendo
- **DESCONHECIDO** (cinza): Status de saúde ainda não determinado

Gerenciamento de Regras

URL: `/rules`

A visualização de Regras fornece uma inspeção abrangente de todas as regras de detecção de pacotes, encaminhamento, QoS e relatório de uso em todas as sessões.

Aba PDR - Regras de Detecção de Pacotes

Visualize e inspecione todos os PDRs no UPF com **formulários de consulta** e **navegação clicável**:

PDRs de Uplink (N3 → N6):

- **Formulário de Consulta**: Pesquisar por TEID para visualizar detalhes específicos do PDR de uplink
- **TEID**: ID de ponto de túnel GTP-U do gNB (clicável - navega para consulta)
- **FAR ID**: Regra de ação de encaminhamento associada (clicável - navega para a aba FAR)
- **QER ID**: Regra de aplicação de QoS associada (clicável - navega para a aba QER)
- **URR IDs**: Regras de relatório de uso associadas (clicável - navega para a aba URR)
- **Remoção de Cabeçalho Externo**: Flag de desencapsulação GTP-U
- **Filtros SDF**: Regras de classificação de fluxo de dados de serviço

PDRs de Downlink (N6 → N3):

- **Formulário de Consulta**: Pesquisar por endereço IPv4 da UE para visualizar detalhes específicos do PDR de downlink
- **UE IP**: Endereço IPv4 do equipamento do usuário (exibido nos resultados da consulta)
- **FAR ID**: Regra de ação de encaminhamento associada (clicável - navega para a aba FAR)
- **QER ID**: Regra de aplicação de QoS associada (clicável - navega para a aba QER)
- **URR IDs**: Regras de relatório de uso associadas (clicável - navega para a aba URR)
- **Modo SDF**: Modo de filtro de fluxo de dados de serviço (nenhum, apenas sdf, sdf + padrão)
- **Paginação**: Navegue pelos PDRs com controles de página (padrão 100 por página, máximo 1000)

PDRs de Downlink IPv6:

- A API suporta paginação para PDRs de downlink IPv6
- Mesma estrutura que IPv4, mas indexada por endereços IPv6
- Uma aba completa da UI pode ser adicionada se necessário

Aba FAR - Regras de Ação de Encaminhamento

Visualize todos os FARs com suas ações de encaminhamento e parâmetros:

Recursos:

- **Formulário de Consulta**: Pesquisar por FAR ID para visualizar detalhes específicos do FAR
- **Auto-consulta**: Clicar em FAR IDs dos detalhes do PDR preenche automaticamente a consulta
- **Atualizações em Tempo Real**: O status do FAR reflete o estado atual do buffer

Coluna	Descrição
FAR ID	Identificador único da regra de encaminhamento
Ação	Flags de ação de encaminhamento (FORWARD, DROP, BUFFER, DUPLICATE, NOTIFY)
Buffering	Status atual de buffering (Ativado/Desativado)
Destino	Parâmetros de criação de cabeçalho externo (TEID, endereço IP)

Flags de Ação do FAR:

- **FORWARD (1)**: Encaminhar pacote para o destino
- **DROP (2)**: Descartar pacote
- **BUFFER (4)**: Armazenar pacote no buffer
- **NOTIFY (8)**: Enviar notificação para o plano de controle
- **DUPLICATE (16)**: Duplicar pacote para múltiplos destinos

Alternância de Buffering:

- Clique em "Ativar Buffer" ou "Desativar Buffer" para alternar a flag de buffering
- Útil para solucionar problemas de cenários de transferência
- Altera imediatamente a ação do FAR no mapa eBPF

Aba QER - Regras de Aplicação de QoS

Visualize as regras de QoS aplicadas aos fluxos de tráfego:

Recursos:

- **Navegação Clicável**: Clique em QER IDs dos detalhes do PDR para navegar e destacar QER específico
- **Auto-destaque**: A linha do QER é destacada quando navegada a partir do PDR
- **Paginação**: Navegue pelos QERs com controles de página (padrão 100 por página, máximo 1000)

Coluna	Descrição
QER ID	Identificador único da regra de QoS (clicável quando referenciado a partir dos PDRs)
MBR (Uplink)	Taxa máxima de bits para tráfego de uplink (kbps)
MBR (Downlink)	Taxa máxima de bits para tráfego de downlink (kbps)
GBR (Uplink)	Taxa garantida de bits para tráfego de uplink (kbps)
GBR (Downlink)	Taxa garantida de bits para tráfego de downlink (kbps)
QFI	Identificador de Fluxo de QoS (marcação 5G)

Interpretação de QoS:

- **MBR = 0**: Sem limite de taxa
- **GBR = 0**: Melhor esforço (sem largura de banda garantida)
- **GBR > 0**: Fluxo de taxa garantida (priorizado)

Aba URR - Regras de Relatório de Uso

Visualize regras de rastreamento de uso e contadores de volume:

Recursos:

- **Formulário de Consulta**: Pesquisar por URR ID para encontrar e destacar URR específico
- **Navegação Clicável**: Clique em URR IDs dos detalhes do PDR para navegar e destacar URR específico
- **Auto-destaque**: A linha do URR é destacada em azul quando navegada a partir do PDR ou pesquisada via consulta
- **Paginação**: Navegue pelos URRs com controles de página (padrão 100 por página, máximo 1000)

Coluna	Descrição
URR ID	Identificador único da regra de relatório de uso (clicável quando referenciado a partir dos PDRs)
Volume de Uplink	Bytes enviados da UE para a rede de dados
Volume de Downlink	Bytes enviados da rede de dados para a UE
Volume Total	Total de bytes em ambas as direções
Ações	Botão de excluir para redefinir contadores para este URR

Exibição de Volume:

- Formatado automaticamente (B, KB, MB, GB, TB)
- Contadores em tempo real atualizados a cada atualização
- Usado para cobrança e análises

Filtragem:

- Mostra apenas URRs com volume diferente de zero
- URRs inativos (todos os contadores em 0) são filtrados para desempenho

Casos de Uso

Inspecionar Classificação de Tráfego:

1. Navegue até Regras → aba PDR
2. Pesquise por TEID ou IP da UE específico
3. Verifique se o PDR associa com o FAR e QER corretos

Solução de Problemas de Encaminhamento:

1. Navegue até Regras → aba FAR
2. Localize o FAR ID a partir do PDR da sessão
3. Verifique se a ação é FORWARD (não DROP ou BUFFER)
4. Verifique os parâmetros de criação de cabeçalho externo

Monitorar Aplicação de QoS:

1. Navegue até Regras → aba QER
2. Verifique se os valores de MBR e GBR correspondem à política
3. Verifique a marcação QFI para fluxos 5G

Rastrear Uso de Dados:

1. Navegue até Regras → aba URR
2. Classifique por volume total para encontrar os maiores usuários
3. Monitore o crescimento do volume ao longo do tempo
4. Verifique a integração de cobrança

Gerenciamento de Buffers

URL: `/buffers`

Recursos

A visualização de Buffers exibe buffers de pacotes mantidos pelo UPF durante eventos de mobilidade ou mudanças de caminho.

Estatísticas Totais

O painel exibe estatísticas agregadas de buffers:

- **Total de Pacotes**: Número de pacotes armazenados em todos os FARs
- **Total de Bytes**: Tamanho total de dados armazenados
- **Total de FARs**: Número de FARs com pacotes armazenados
- **Máximo por FAR**: Máximo de pacotes permitidos por FAR
- **Máximo Total**: Máximo total de pacotes armazenados
- **TTL do Pacote**: Tempo de vida para pacotes armazenados (segundos)

Buffers por FAR

Tabela de todos os FARs com pacotes armazenados:

Coluna	Descrição
FAR ID	Identificador da regra de ação de encaminhamento
Contagem de Pacotes	Número de pacotes armazenados para este FAR
Contagem de Bytes	Total de bytes armazenados para este FAR
Pacote Mais Antigo	Timestamp do pacote armazenado mais antigo
Pacote Mais Novo	Timestamp do pacote armazenado mais novo
Ações	Botões de controle de buffer (estilo pílula)

Ações de Controle de Buffer

Para cada FAR com pacotes armazenados, os seguintes botões de estilo pílula estão disponíveis:

Controle de Buffering:

- **Desativar Buffer** (vermelho): Desligar o buffering para este FAR (atualiza a flag de ação do FAR)
- **Ativar Buffer** (roxo): Ligar o buffering para este FAR

Operações de Buffer:

- **Flush** (azul): Repetir todos os pacotes armazenados usando as regras atuais do FAR
- **Limpar** (cinza): Excluir todos os pacotes armazenados sem encaminhamento

Limpar Todos os Buffers:

- Botão vermelho "Limpar Tudo" no cabeçalho
- Limpa buffers para todos os FARs

- Requer confirmação

Casos de Uso

Monitorar Buffering de Transferência:

1. Durante a transferência, verifique se os pacotes estão sendo armazenados
2. Verifique o status de buffering do FAR (deve estar ativado)
3. Monitore a contagem e a idade dos pacotes

Completar Transferência:

1. Após a mudança de caminho, clique em "Flush" para repetir os pacotes armazenados
2. Verifique se os pacotes são encaminhados para o novo caminho
3. Clique em "Desativar Buffer" para parar o buffering

Limpar Buffers Presos:

1. Identifique FARs com pacotes armazenados antigos (verifique o timestamp mais antigo)
2. Clique em "Limpar" para descartar pacotes obsoletos
3. Ou clique em "Desativar Buffer" para evitar mais armazenamento

Solução de Problemas de Overflow de Buffer:

1. Verifique a contagem total de pacotes vs. máximo total
2. Identifique FARs com buffering excessivo
3. Verifique se o SMF enviou modificação de sessão para desativar o buffering
4. Desative manualmente o buffering se o comando do SMF foi perdido

Atualizações em Tempo Real

A visualização de Buffers atualiza automaticamente a cada 5 segundos para mostrar o status atual do buffer.

Painel de Estatísticas

URL: `/statistics`

Recursos

A visualização de Estatísticas fornece métricas de desempenho em tempo real do datapath do OmniUPF. Para informações detalhadas sobre métricas do Prometheus, consulte a **Referência de Métricas**.

Estatísticas de Pacotes

Contadores agregados de processamento de pacotes:

- **Pacotes RX**: Total de pacotes recebidos em todas as interfaces
- **Pacotes TX**: Total de pacotes transmitidos em todas as interfaces
- **Pacotes Descartados**: Pacotes descartados devido a erros ou políticas
- **Pacotes GTP-U**: Pacotes processados com encapsulação GTP-U

Uso: Monitorar a carga de tráfego geral do UPF e a taxa de queda de pacotes

Estatísticas de Rotas

Métricas de encaminhamento por rota (se disponíveis):

- **Acertos de Rota**: Pacotes correspondidos por cada regra de roteamento
- **Sucesso de Encaminhamento**: Contagem de pacotes encaminhados com sucesso
- **Erros de Encaminhamento**: Tentativas de encaminhamento falhadas

Uso: Identificar rotas ocupadas e erros de encaminhamento

Estatísticas XDP

Métricas de desempenho do eXpress Data Path:

- **XDP Processados**: Total de pacotes processados na camada XDP
- **XDP Passados**: Pacotes enviados para a pilha de rede

- **XDP Descartados**: Pacotes descartados na camada XDP
- **XDP Abortados**: Erros de processamento no programa XDP

Uso: Monitorar o desempenho do XDP e detectar erros de processamento

Causas de Descarte do XDP:

- Formato de pacote inválido
- Falha na consulta de mapa eBPF
- Descartes baseados em políticas
- Exaustão de recursos

Estatísticas de Interface N3/N6

Contadores de tráfego por interface:

Interface N3 (conectividade RAN):

- **RX N3**: Pacotes recebidos do gNB/eNodeB
- **TX N3**: Pacotes transmitidos para gNB/eNodeB

Interface N6 (conectividade da Rede de Dados):

- **RX N6**: Pacotes recebidos da rede de dados (Internet/IMS)
- **TX N6**: Pacotes transmitidos para a rede de dados

Total: Contagem agregada de pacotes através das interfaces

Uso: Monitorar o equilíbrio de tráfego e problemas específicos de interface

Casos de Uso

Monitorar Carga de Tráfego:

1. Verifique as taxas de pacotes RX/TX
2. Verifique se o tráfego está fluindo em ambas as direções
3. Compare o tráfego N3 vs N6 (deve ser aproximadamente igual)

Detectar Quedas de Pacotes:

1. Verifique o contador de pacotes descartados
2. Revise o contador de pacotes descartados do XDP
3. Investigue a causa nos logs se as quedas forem altas

Análise de Desempenho:

1. Monitore a razão de pacotes processados vs. passados
2. Verifique se há abortos do XDP (indica erros)
3. Verifique a distribuição de tráfego das interfaces N3/N6

Planejamento de Capacidade:

1. Rastreie a taxa de pacotes ao longo do tempo
2. Compare com os limites de capacidade do UPF
3. Planeje escalonamento se estiver se aproximando dos limites

Atualizações em Tempo Real

As estatísticas atualizam automaticamente a cada 5 segundos.

Monitoramento de Capacidade

URL: `/capacity`

Recursos

A visualização de Capacidade exibe o uso de mapas eBPF e limites de capacidade para todos os mapas no datapath do UPF.

Tabela de Uso de Mapas eBPF

Tabela de todos os mapas eBPF com informações de uso:

Coluna	Descrição
Nome do Mapa	Nome do mapa eBPF (por exemplo, `uplink_pdr_map`, `far_map`)
Usado	Número de entradas atualmente no mapa
Capacidade	Máximo de entradas permitidas no mapa
Uso	Barra de progresso visual com porcentagem
Tamanho da Chave	Tamanho das chaves do mapa em bytes
Tamanho do Valor	Tamanho dos valores do mapa em bytes

Indicadores de Uso Coloridos

A barra de progresso de uso é colorida com base na utilização:

- **Verde (<50%)**: Operação normal, capacidade ampla
- **Amarelo (50-70%)**: Cuidado, monitorar crescimento
- **Âmbar (70-90%)**: Aviso, planejar aumento de capacidade
- **Vermelho (>90%)**: Crítico, ação imediata necessária

Mapas Críticos para Monitorar

uplink_pdr_map:

- Armazena PDRs de uplink indexados por TEID
- Uma entrada por fluxo de tráfego de uplink
- **Crítico**: A exaustão impede o estabelecimento de novas sessões

downlink_pdr_map / **downlink_pdr_map_ip6**:

- Armazena PDRs de downlink indexados por endereço IP da UE

- Uma entrada por endereço IPv4/IPv6 da UE
- **Crítico**: A exaustão impede o estabelecimento de novas sessões

far_map:

- Armazena regras de ação de encaminhamento indexadas por FAR ID
- Compartilhado entre vários PDRs
- **Alta Prioridade**: Afeta decisões de encaminhamento

qer_map:

- Armazena regras de aplicação de QoS indexadas por QER ID
- **Prioridade Média**: Afeta QoS, mas não conectividade básica

urr_map:

- Armazena regras de relatório de uso indexadas por URR ID
- **Baixa Prioridade**: Afeta cobrança, mas não conectividade

Casos de Uso

Planejamento de Capacidade:

1. Monitore tendências de uso de mapas ao longo do tempo
2. Identifique quais mapas estão crescendo mais rápido
3. Planeje aumentos de capacidade antes de atingir limites

Prevenir Falhas de Estabelecimento de Sessão:

1. Verifique o uso do mapa PDR antes de um aumento esperado de tráfego
2. Aumente a capacidade do mapa se estiver se aproximando dos limites
3. Monitore após o aumento de capacidade para verificar

Solução de Problemas de Falhas de Sessão:

1. Quando o estabelecimento da sessão falhar, verifique a visualização de Capacidade
2. Se os mapas PDR estiverem vermelhos (>90%), a capacidade está esgotada

3. Aumente a capacidade do mapa ou limpe sessões obsoletas

Otimizar Configuração de Mapas:

1. Revise tamanhos de chave e valor
2. Calcule o uso de memória por mapa
3. Otimize tamanhos de mapa com base em padrões de uso reais

Configuração de Capacidade

As capacidades dos mapas eBPF são configuradas na inicialização do UPF no arquivo de configuração do UPF. Valores típicos:

- Implantação pequena: 10.000 - 100.000 entradas por mapa
- Implantação média: 100.000 - 1.000.000 entradas por mapa
- Implantação grande: 1.000.000+ entradas por mapa

Cálculo de Memória:

```
Memória do Mapa = (Tamanho da Chave + Tamanho do Valor) × Capacidade
```

Por exemplo, um mapa PDR com 1 milhão de entradas e valores de 64 bytes usa aproximadamente 64 MB de memória do kernel.

Atualizações em Tempo Real

A visualização de capacidade atualiza automaticamente a cada 10 segundos.

Visualização de Configuração

URL: `/upf_config`

Recursos

A visualização de Configuração exibe parâmetros operacionais do UPF e configuração do dataplane.

Configuração do UPF

Exibe a configuração estática do UPF:

- **Interface PFCP**: Endereço IP e porta para conectividade SMF/PGW-C
- **Interface N3**: Endereço IP para conectividade RAN (gNB/eNodeB)
- **Interface N6**: Endereço IP para conectividade da rede de dados
- **Interface N9**: Endereço IP para comunicação inter-UPF (opcional)
- **Porta da API**: Porta de escuta da API REST
- **Versão**: Versão do software OmniUPF

Configuração do Dataplane (eBPF)

Exibe parâmetros ativos do dataplane em tempo de execução:

- **Endereço N3 Ativo**: Ligação da interface N3 em tempo de execução
- **Endereço N9 Ativo**: Ligação da interface N9 em tempo de execução (se habilitada)

Esses valores refletem a configuração real do datapath eBPF e podem diferir da configuração estática se as interfaces foram alteradas.

Casos de Uso

Verificar Conectividade do UPF:

1. Verifique se o IP da interface N3 corresponde à configuração do gNB
2. Verifique se a interface N6 pode rotear para a rede de dados
3. Confirme se a interface PFCP é acessível a partir do SMF

Solução de Problemas de Problemas de Interface:

1. Compare a configuração estática com os endereços ativos do dataplane

2. Verifique se as interfaces estão vinculadas corretamente
3. Verifique se houve alterações na configuração da interface

Documentação e Auditoria:

1. Registre a configuração do UPF para documentação
2. Verifique se a implantação corresponde às especificações de design
3. Audite as atribuições de interface

Visualização de Rotas

URL: `/routes`

Recursos

A visualização de Rotas fornece monitoramento abrangente das rotas IP do Equipamento do Usuário (UE) e sessões de protocolo de roteamento (OSPF e BGP).

Visão Geral do Status da Rota

O painel exibe estatísticas agregadas de rotas:

- **Status**: Roteamento habilitado ou desabilitado
- **Total de Rotas**: Número total de rotas IP de UE
- **Sincronizado**: Número de rotas sincronizadas com sucesso
- **Falhou**: Número de rotas que falharam ao sincronizar

Rotas IP Ativas de UE

Tabela exibindo todas as rotas IP ativas do Equipamento do Usuário:

Coluna	Descrição
Índice	Número do índice da rota
Endereço IP da UE	Endereço IPv4 ou IPv6 atribuído à UE

Propósito:

- Visualizar todos os endereços IP de UE que têm rotas configuradas
- Verificar a distribuição de rotas para protocolos de roteamento
- Monitorar o status de sincronização de rotas

Vizinhos OSPF

Tabela de vizinhos do protocolo OSPF (Open Shortest Path First):

Coluna	Descrição
Neighbor ID	Identificador do roteador OSPF
Endereço	Endereço IP do vizinho OSPF
Interface	Interface usada para a adjacência OSPF
Estado	Estado da adjacência OSPF (Completo, Inicial, etc.)
Prioridade	Valor de prioridade OSPF
Tempo de Atividade	Duração que o vizinho está ativo
Tempo de Morte	Tempo até que o vizinho seja considerado morto

Estados OSPF:

- **Completo** (verde): Totalmente adjacente e trocando informações de roteamento
- **Outros estados** (amarelo): Adjacência em formação ou incompleta

Pares BGP

Tabela de pares BGP (Border Gateway Protocol):

Coluna	Descrição
Neighbor IP	Endereço IP do par BGP
ASN	Número do Sistema Autônomo do par
Estado	Estado da sessão BGP (Estabelecido, Ocioso, etc.)
Up/Down	Duração do estado atual
Prefixos Recebidos	Número de prefixos de rota recebidos do par
Msg Enviadas	Total de mensagens BGP enviadas ao par
Msg Recebidas	Total de mensagens BGP recebidas do par

Estados BGP:

- **Estabelecido** (verde): Sessão BGP ativa, trocando rotas
- **Outros estados** (vermelho): Sessão inativa ou em estabelecimento

O cabeçalho também exibe o ID do Roteador BGP local e ASN quando o BGP está configurado.

Rotas Redistribuídas OSPF

Tabela mostrando LSAs Externas OSPF (Link State Advertisements) para rotas de UE redistribuídas:

Coluna	Descrição
Link State ID	Identificador LSA (tipicamente o endereço da rede)
Máscara	Máscara de rede para a rota
Roteador Anunciante	ID do roteador que anuncia esta rota externa
Tipo de Métrica	Tipo de métrica externa OSPF (E1 ou E2)
Métrica	Métrica de custo OSPF para a rota
Idade	Tempo desde que a LSA foi originada (segundos)
Seq Number	Número de sequência da LSA para versionamento

Propósito:

- Verificar se as rotas de UE estão sendo redistribuídas no OSPF
- Monitorar qual roteador está anunciando rotas externas
- Rastrear a idade e atualizações da LSA

Ações de Controle de Rotas

Botão Sincronizar Rotas:

- Aciona manualmente a sincronização de rotas para o FRR (Free Range Routing)
- Força a atualização do protocolo de roteamento com as rotas atuais de UE
- Útil após alterações de configuração ou para recuperar de falhas de sincronização

Botão Atualizar:

- Atualiza manualmente todas as informações de rotas

- Atualiza vizinhos OSPF, pares BGP e tabelas de rotas

Casos de Uso

Monitorar Saúde do Protocolo de Roteamento:

1. Navegue até a visualização de Rotas
2. Verifique os estados dos vizinhos OSPF (deve ser "Completo")
3. Verifique se os pares BGP estão "Estabelecidos"
4. Confirme o número esperado de vizinhos/pares

Verificar Distribuição de Rotas de UE:

1. Verifique a tabela de Rotas IP Ativas de UE para uma UE específica
2. Role até a seção de Rotas Redistribuídas OSPF
3. Verifique se a rota de UE aparece nas LSAs externas
4. Confirme se o roteador anunciante corresponde ao UPF esperado

Solução de Problemas de Problemas de Sincronização de Rotas:

1. Verifique os contadores Sincronizados vs. Falhados na visão geral do status
2. Se as rotas estiverem falhando, clique no botão "Sincronizar Rotas"
3. Monitore mensagens de erro na faixa vermelha se a sincronização falhar
4. Verifique mensagens de erro OSPF/BGP nas seções respectivas

Verificar Implantação Multi-UPF:

1. Selecione diferentes instâncias do UPF no menu suspenso
2. Compare contagens de rotas entre instâncias
3. Verifique se os vizinhos OSPF se veem
4. Verifique relacionamentos de peering BGP

Monitorar Escalonamento de Rotas:

1. Rastreie a contagem total de rotas à medida que as sessões de UE aumentam
2. Verifique se as rotas estão sendo distribuídas para os protocolos de roteamento

3. Monitore o crescimento da contagem de LSA OSPF
4. Verifique a contagem de prefixos BGP recebidos pelos pares

Atualizações em Tempo Real

A visualização de Rotas atualiza automaticamente a cada 10 segundos para mostrar o status atual do protocolo de roteamento e as rotas de UE.

Integração de Roteamento

A visualização de Rotas se integra ao FRR (Free Range Routing) em execução no UPF:

- **OSPF**: As rotas são redistribuídas como LSAs Externas do Tipo 2
- **BGP**: As rotas são anunciadas para pares BGP configurados
- **Mecanismo de Sincronização**: Chamadas da API REST acionam comandos vtysh para atualizar o FRR

Visualização de Capacidades XDP

URL: `/xdp_capabilities`

Recursos

A visualização de Capacidades XDP exibe suporte ao modo eXpress Data Path (XDP), capacidades de desempenho e cálculos de throughput para o dataplane do UPF.

Configuração da Interface

Exibe informações sobre a interface de rede e o driver:

Campo	Descrição
Nome da Interface	Interface de rede usada para XDP (por exemplo, eth0, ens1f0)
Driver	Nome do driver de rede (por exemplo, i40e, ixgbe, virtio_net)
Versão do Driver	String da versão do driver
Modo Atual	Modo XDP ativo (DRV, SKB ou NENHUM)
Contagem de Múltiplas Filas	Número de pares de filas NIC para processamento paralelo

Modos XDP

A visualização exibe todos os modos XDP com seu status de suporte e características de desempenho:

XDP_DRV (Modo do Driver):

- **Desempenho**: ~5-10 Mpps (milhões de pacotes por segundo)
- **Descrição**: Suporte nativo ao XDP no driver, maior desempenho
- **Requer**: Driver NIC com suporte nativo ao XDP (i40e, ixgbe, mlx5, etc.)
- **Status**: Suportado se o driver tiver ganchos XDP
- **Indicador**: Marca de verificação verde (✓) se suportado, X vermelho (✗) se não

XDP_SKB (Modo Genérico):

- **Desempenho**: ~1-2 Mpps
- **Descrição**: Modo de fallback usando a pilha de rede do kernel
- **Requer**: Qualquer interface de rede
- **Status**: Sempre suportado
- **Indicador**: Marca de verificação verde (✓)

Indicador de Modo Atual:

- Ponto azul ao lado do modo XDP atualmente ativo
- Mostra qual modo está realmente em uso

Razões para Modos Não Suportados:

- Se um modo não for suportado, o campo "Razão" explica o porquê
- Razões comuns: driver sem suporte ao XDP, incompatibilidade de tipo de interface

Visualização de Capacidades XDP mostrando configuração da interface, modos suportados e o calculador interativo de throughput Mpps

Recomendações

A visualização exibe um banner de recomendação colorido com base na configuração atual:

Verde (Ótimo):

- "✓ Ótimo: Modo XDP_DRV ativado com suporte nativo do driver"
- O modo de maior desempenho está ativo

Amarelo (Aviso):

- "⚠ Considere atualizar para o modo XDP_DRV para melhor desempenho"
- Executando em modo genérico quando o modo do driver está disponível
- "⚠ Aviso: XDP_DRV não suportado por este driver"
- Limitações de hardware impedem desempenho ideal

Azul (Informativo):

- Informações gerais sobre a configuração do XDP

Calculadora de Desempenho Mpps

Calculadora interativa para converter taxa de pacotes (Mpps) em throughput (Gbps):

Parâmetros de Entrada

Taxa de Pacotes (Mpps):

- Faixa: 0,1 - 100 Mpps
- Padrão: Máximo de Mpps para o modo XDP atual
- Representa milhões de pacotes processados por segundo

Tamanho Médio do Pacote (bytes):

- Faixa: 64 - 9000 bytes
- Padrão: 1200 bytes (pacote GTP típico)
- Inclui pacote completo com encapsulação GTP

Botões de Predefinição Rápida:

- **64B (mínimo)**: Tamanho mínimo do quadro Ethernet
- **128B**: Pacotes pequenos
- **256B**: Plano de controle ou sinalização

- **512B**: Pacotes de tamanho médio
- **1024B**: Pacotes grandes
- **1518B (máximo)**: Tamanho máximo do quadro Ethernet sem quadros jumbo

Resultados do Cálculo

Throughput Total (Gbps):

- Throughput de taxa de fio incluindo todos os cabeçalhos
- Fórmula: `Gbps = Mpps × Packet_Size × 8 / 1000`
- Inclui cabeçalhos GTP, UDP, IP e Ethernet

Taxa de Dados do Usuário (Gbps):

- Throughput real da carga útil do usuário
- Exclui ~50 bytes de sobrecarga de encapsulação GTP
- Fórmula: `Gbps = Mpps × (Packet_Size - 50) / 1000`

Taxa de Pacotes:

- Exibe Mpps e pacotes/segundo com separador de milhar
- Exemplo: 10 Mpps = 10.000.000 pacotes/segundo

Exibição da Fórmula:

- Mostra a decomposição do cálculo passo a passo
- Exemplo: `10 Mpps × 1200 bytes × 8 bits/byte ÷ 1000 = 96 Gbps`

Compreendendo Mpps

A visualização inclui uma seção de explicação cobrindo:

O que é Mpps:

- Milhões de Pacotes por Segundo
- Métrica chave para desempenho de processamento de pacotes
- Independente do tamanho do pacote

Relação com Throughput:

- Mesma Mpps com pacotes maiores = maior Gbps
- Mesma Mpps com pacotes menores = menor Gbps
- Throughput depende tanto da taxa quanto do tamanho do pacote

Sobrecarga de Encapsulação GTP:

- Cabeçalho Ethernet: 14 bytes
- Cabeçalho IP: 20 bytes (IPv4) ou 40 bytes (IPv6)
- Cabeçalho UDP: 8 bytes
- Cabeçalho GTP: 8 bytes (mínimo)
- Sobrecarga total típica: ~50 bytes por pacote

Casos de Uso

Avaliar Desempenho do XDP:

1. Navegue até a visualização de Capacidades XDP
2. Verifique o modo XDP atual (deve ser DRV para melhor desempenho)
3. Observe a faixa de desempenho Mpps
4. Revise o banner de recomendação

Calcular Throughput Esperado:

1. Insira a taxa de pacotes esperada em Mpps
2. Insira o tamanho médio do pacote para seu perfil de tráfego
3. Revise o throughput calculado em Gbps
4. Compare com a capacidade do link ou requisitos de desempenho

Otimizar Configuração do XDP:

1. Verifique se o modo XDP_DRV é suportado, mas não está ativo
2. Revise a versão do driver e a compatibilidade
3. Siga a recomendação para atualizar para o modo do driver, se disponível
4. Verifique se a contagem de múltiplas filas corresponde aos núcleos da CPU

Planejamento de Capacidade:

1. Use a calculadora para determinar Mpps necessário para throughput alvo
2. Compare com as capacidades do modo XDP atual
3. Determine se é necessária atualização de hardware
4. Planeje seleção de interface e driver para novas implantações

Solução de Problemas de Problemas de Desempenho:

1. Verifique se o modo XDP é DRV, não SKB
2. Verifique a versão do driver para problemas de desempenho conhecidos
3. Verifique se a contagem de múltiplas filas é suficiente
4. Calcule se o modo atual suporta o throughput necessário

Dicas de Otimização de Desempenho

Modo do Driver (XDP_DRV):

- Use NICs com suporte nativo ao XDP (Intel i40e/ixgbe, Mellanox mlx5)
- Atualize os drivers NIC para a versão mais recente
- Ative múltiplas filas (RSS) para processamento paralelo
- Ajuste os tamanhos do buffer de anel NIC

Modo Genérico (XDP_SKB):

- Aceitável para desenvolvimento e teste
- Não recomendado para produção de alto throughput
- Considere atualização de hardware para implantações de produção

Configuração de Múltiplas Filas:

- O número de filas deve corresponder ou exceder a contagem de núcleos da CPU
- Permite processamento paralelo de pacotes entre núcleos
- Distribui carga via RSS (Receive Side Scaling)

Atualizações em Tempo Real

A visualização de Capacidades XDP atualiza a cada 30 segundos para atualizar o status da interface e informações de modo.

Visualizador de Logs

URL: `/logs`

Recursos

Visualize os logs da aplicação OmniUPF em tempo real a partir do painel de controle.

Recursos:

- Streaming de logs ao vivo via Phoenix LiveView
- Atualizações em tempo real à medida que os logs são gerados
- Histórico de logs rolável
- Útil para solução de problemas durante sessões ativas

Níveis de Log

Os logs do OmniUPF usam níveis padrão do Elixir Logger:

- **DEBUG**: Informações diagnósticas detalhadas
- **INFO**: Mensagens informativas gerais (padrão)
- **WARNING**: Mensagens de aviso para problemas não críticos
- **ERROR**: Mensagens de erro para falhas

Casos de Uso

Solução de Problemas de Estabelecimento de Sessão:

1. Abra a visualização de Logs
2. Inicie o estabelecimento de sessão a partir do SMF

3. Observe os logs de mensagens PFCP e quaisquer erros

Monitorar Comunicação PFCP:

1. Visualize mensagens de configuração de associação PFCP
2. Acompanhe a criação/modificação/exclusão de sessões
3. Verifique mensagens de heartbeat

Depurar Problemas de Encaminhamento:

1. Procure por erros de processamento de pacotes
2. Verifique logs de operação do mapa eBPF
3. Identifique problemas de configuração de FAR/PDR

Melhores Práticas

Diretrizes Operacionais

Monitoramento:

- Verifique regularmente a visualização de Capacidade para evitar exaustão de mapas
- Monitore Estatísticas para padrões de tráfego incomuns ou quedas
- Rastreie o crescimento da contagem de sessões ao longo do tempo
- Fique atento a erros de processamento do XDP

Gerenciamento de Buffers:

- Monitore buffers durante cenários de transferência
- Limpe buffers presos se os pacotes ultrapassarem o TTL
- Verifique se o buffering está desativado após a conclusão da transferência
- Use "Flush" em vez de "Clear" para evitar perda de pacotes

Gerenciamento de Sessões:

- Use filtros para localizar rapidamente sessões específicas de UE
- Expanda sessões para verificar a configuração da regra

- Compare sessões entre várias instâncias do UPF
- Verifique o indicador de saúde antes de solucionar problemas

Solução de Problemas:

- Use Logs para depuração em tempo real
- Verifique a visualização de Sessões para verificar a conectividade da UE
- Verifique a configuração de Regras para fluxos de tráfego
- Monitore Estatísticas para quedas de pacotes ou erros de encaminhamento

Desempenho

- A atualização automática do painel é de 5-10 segundos, dependendo da visualização
- Listas grandes de sessões podem levar tempo para carregar
- A visualização de regras filtra por entradas ativas (volumes diferentes de zero para URRs)
- As operações de buffer são executadas imediatamente no UPF selecionado

Documentação Relacionada

- **Guia de Gerenciamento de Regras** - Configuração de PDR, FAR, QER, URR
- **Guia de Monitoramento** - Estatísticas, métricas e planejamento de capacidade
- **Referência de Métricas** - Referência completa de métricas do Prometheus
- **Códigos de Causa PFCP** - Códigos de erro PFCP e diagnósticos de sessão
- **Documentação da API** - Referência da API REST e paginação
- **Guia de Rotas** - Detalhes sobre roteamento de UE e integração FRR
- **Guia de Modos XDP** - Documentação detalhada sobre modos XDP e informações eBPF
- **Guia de Solução de Problemas** - Problemas comuns e diagnósticos
- **Guia de Operações do UPF** - Operações gerais do UPF e arquitetura

Modos de Anexação XDP para OmniUPF

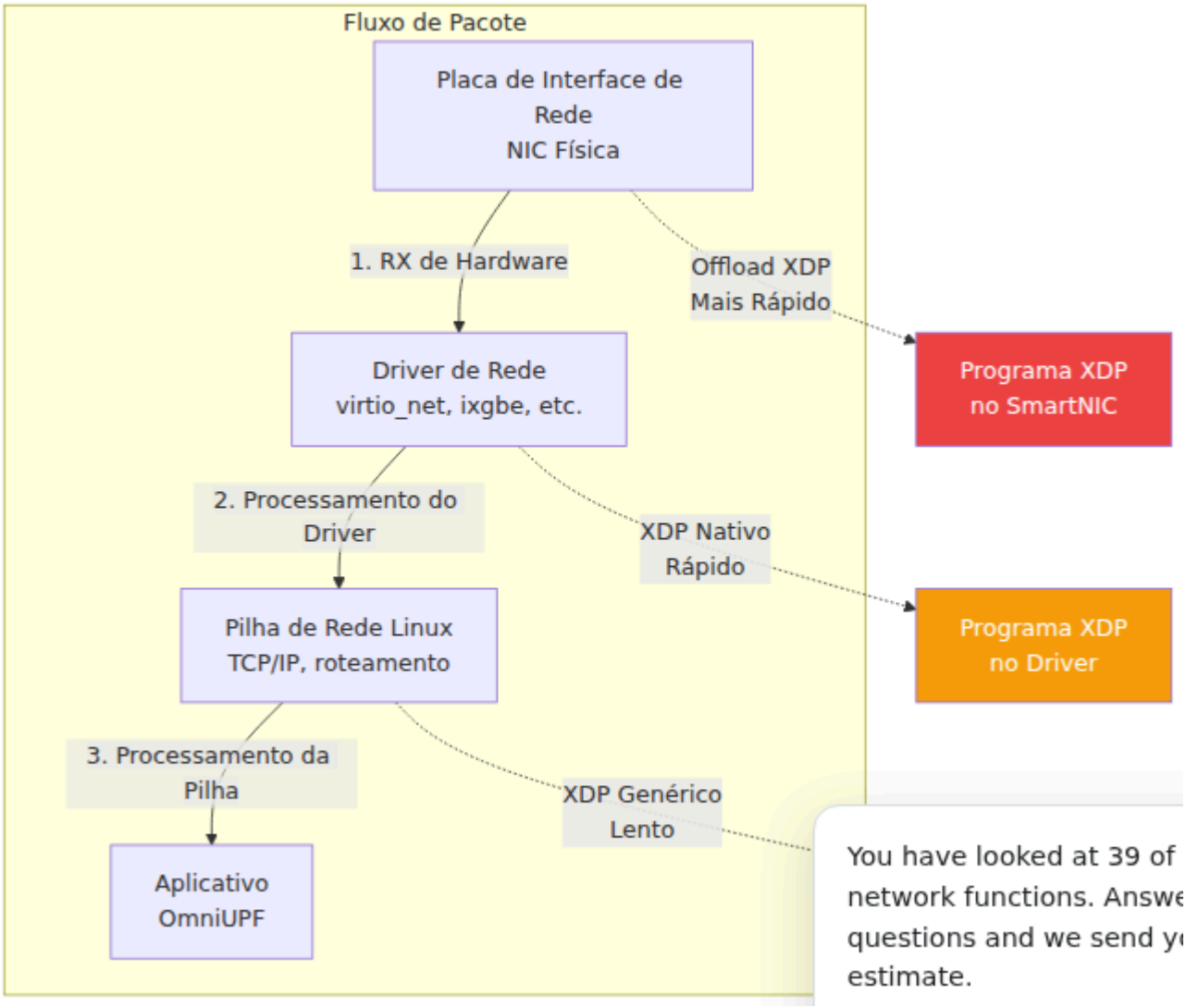
Índice

1. Visão Geral
2. Comparação de Modos XDP
3. Modo Genérico (Padrão)
4. Modo Nativo (Recomendado para Produção)
5. Modo Offload (SmartNIC)
6. Habilitando XDP Nativo no Proxmox VE
7. Habilitando XDP Nativo em Outros Hipervisores
8. Verificando o Modo XDP
9. Resolvendo Problemas de XDP

Visão Geral

OmniUPF utiliza **XDP (eXpress Data Path)** para processamento de pacotes de alto desempenho. XDP é uma tecnologia do kernel Linux que permite que programas de processamento de pacotes (eBPF) sejam executados no ponto mais cedo possível na pilha de rede, proporcionando latência em nível de microssegundos e milhões de pacotes por segundo de throughput.

O modo de anexação XDP determina **onde** no caminho do pacote o programa eBPF é executado:



Escolher o modo XDP correto impacta significativamente o desempenho do OmniUPF e determina se você pode alcançar um processamento de pacotes de nível de produção.

Comparação de Modos XDP

Aspecto	Modo Genérico	Modo Nativo	Modo Offload
Ponto de Anexação	Pilha de rede Linux	Driver de rede	Hardware NIC
Desempenho	~1-2 Mpps	~5-10 Mpps	~10-40 Mpps
Latência	~100 µs	~10 µs	~1 µs
Uso de CPU	Alto	Médio	Baixo
Requisitos de NIC	Qualquer NIC	Driver compatível com XDP	SmartNIC com suporte a XDP
Suporte a Hipervisores	Todos os hipervisores	A maioria (requer multi-queue)	Raro (PCI passthrough)
Caso de Uso	Testes, desenvolvimento	**Produção (recomendado)**	Sites de borda de alto throughput
Configuração	`xdp_attach_mode: generic`	`xdp_attach_mode: native`	`xdp_attach_mode offload`

Recomendação: Use **modo nativo** para implantações em produção. O modo genérico é adequado apenas para testes.

Modo Genérico (Padrão)

Descrição

O XDP genérico executa o programa eBPF na pilha de rede Linux **após** o driver ter processado o pacote. Este é o modo XDP mais lento, mas funciona com

qualquer interface de rede.

Características de Desempenho

- **Throughput**: ~1-2 milhões de pacotes por segundo (Mpps)
- **Latência**: ~100 microsegundos por pacote
- **Sobrecarga de CPU**: Alta (pacote copiado para a pilha do kernel antes do XDP)

Quando Usar

- **Desenvolvimento e testes** apenas
- **Ambientes de laboratório** onde o desempenho não importa
- **Implantação inicial** para verificar a funcionalidade antes de otimizar

Configuração

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"
xdp_attach_mode = "generic"    # Modo padrão
```

Aviso: O modo genérico **não é adequado para produção**. Ele se tornará um gargalo em altas taxas de pacotes e desperdiçará recursos de CPU.

Cuidado com o checksum-offload: No modo genérico, o pacote encaminhado é um `skb` do kernel, portanto, carrega os metadados de checksum-offload TX da NIC. Um pacote de controle **originado em espaço de usuário** (por exemplo, o Anúncio de Roteador IPv6 que responde a uma Solicitação de Roteador de um UE) é marcado como `CHECKSUM_PARTIAL`; quando o caminho de dados o retransmite e modifica, o offload obsoleto `csum_start`/`csum_offset` faz com que a NIC carimbe um checksum UDP **errado** na saída e o receptor o descarta. Isso não afeta o tráfego de plano de usuário nativo do XDP (`CHECKSUM_NONE`), apenas pacotes de controle, e não ocorre no modo nativo/offload (sem skb). Em NICs com checksum-offloading (por exemplo, virtio), desative o offload de checksum TX nas NICs do caminho

de dados (`ethtool -K <iface> tx off`) — veja TROUBLESHOOTING.md ("SLAAC IPv6 intermitente / erros de checksum UDP").

Modo Nativo (Recomendado para Produção)

Descrição

O XDP nativo executa o programa eBPF **dentro do driver de rede**, antes que os pacotes cheguem à pilha de rede Linux. Isso proporciona desempenho quase de hardware, mantendo a flexibilidade em nível de kernel.

Características de Desempenho

- **Throughput**: ~5-10 milhões de pacotes por segundo (Mpps) por núcleo
- **Latência**: ~10 microsegundos por pacote
- **Sobrecarga de CPU**: Baixa (pacote processado no nível do driver)
- **Escalabilidade**: Escalonamento linear com núcleos de CPU e filas de NIC

Quando Usar

- **Implantações em produção** (recomendado)
- **Redes de nível carrier** que requerem alto throughput
- **Cenários de computação de borda** com requisitos de desempenho
- **Qualquer implantação** onde o desempenho é importante

Requisitos do Driver NIC

O XDP nativo requer um driver de rede com suporte a XDP. A maioria das NICs modernas suporta XDP nativo:

NICs Físicas (bare metal):

- Intel: `ixgbe` (10G), `i40e` (40G), `ice` (100G)

- Broadcom: `bnxt_en`
- Mellanox: `mlx4_en`, `mlx5_core`
- Netronome: `nfp` (com suporte a offload)
- Marvell: `mvneta`, `mvpp2`

NICs Virtuais (hipervisores):

- VirtIO: `virtio_net` (KVM, Proxmox, OpenStack) ✓
- VMware: `vmxnet3` ✓
- Microsoft: `hv_netvsc` (Hyper-V) ✓
- Amazon: `ena` (AWS) ✓
- SR-IOV: `ixgbevf`, `i40evf` (PCI passthrough) ✓

Nota: O VirtualBox **não** suporta XDP nativo (use apenas o modo genérico).

Configuração

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"
xdp_attach_mode = "native"
```

Requisito de Multi-Queue: Para desempenho ideal, habilite multi-queue em NICs virtuais (veja a seção do Proxmox abaixo).

Modo Offload (SmartNIC)

Descrição

O XDP offload executa o programa eBPF **diretamente no hardware da NIC** (SmartNIC), contornando completamente a CPU para o processamento de pacotes. Isso proporciona o maior desempenho, mas requer hardware especializado.

Características de Desempenho

- **Throughput**: ~10-40 milhões de pacotes por segundo (Mpps)
- **Latência**: ~1 microsegundo por pacote
- **Sobrecarga de CPU**: Quase zero (processamento na NIC)

Quando Usar

- **Implantações de ultra-alto throughput** (10G+ por instância do UPF)
- **Sites de borda** com aceleração de hardware
- **Implantações sensíveis a custo** (reduzir requisitos de CPU)

Requisitos de Hardware

Apenas as SmartNICs Netronome Agilio atualmente suportam offload XDP:

- Netronome Agilio CX 10G/25G/40G/100G

Nota: O modo offload requer **bare metal** ou **PCI passthrough** - não disponível em configurações padrão de VM.

Configuração

```
# /etc/omniupf/runtime.exs
xdp_interfaces = "eth0"
xdp_attach_mode = "offload"
```

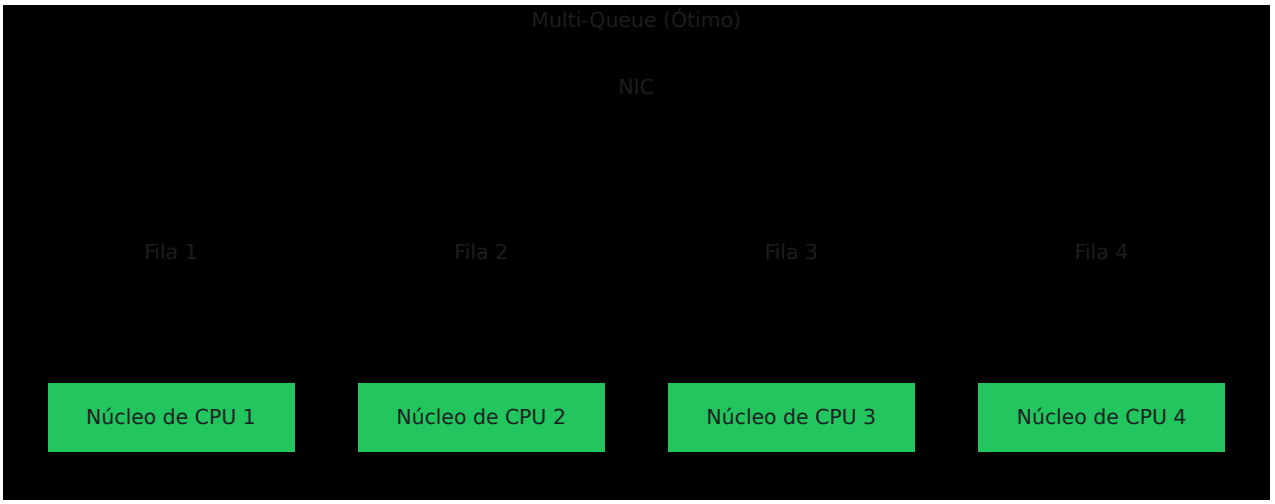
Habilitando XDP Nativo no Proxmox VE

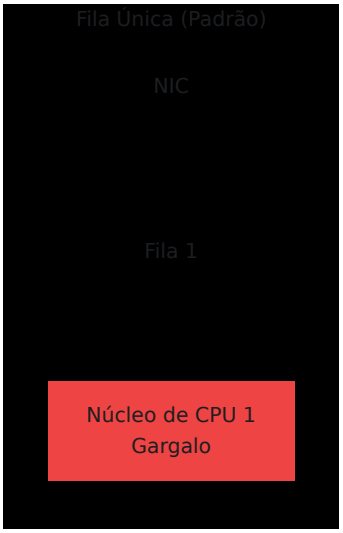
O Proxmox VE utiliza dispositivos de rede **VirtIO** para VMs, que suportam XDP nativo através do driver `virtio_net`. No entanto, você deve habilitar **multi-queue** para desempenho ideal.

Passo 1: Entendendo o Requisito

Por que o Multi-Queue é Importante:

- **Fila única** (padrão): Todo o tráfego de rede processado por um núcleo de CPU → gargalo
- **Multi-queue**: Tráfego distribuído entre múltiplos núcleos de CPU → escalonamento linear





Passo 2: Habilitar Multi-Queue no Proxmox

Opção A: Via Interface Web do Proxmox

1. **Desligue completamente a VM** (não apenas reinicie)

- Selecione sua VM na interface web do Proxmox
- Clique em **Desligar**

2. **Editar Dispositivo de Rede**
   - Vá para a aba **Hardware**
   - Clique no seu dispositivo de rede (por exemplo, `net0`)
   - Clique em **Editar**
3. **Definir Multiqueue**
   - Encontre o campo **"Multiqueue"**
   - Defina como **8** (ou corresponda à sua contagem de vCPU, máximo 16)
   - Clique em **OK**
4. **Iniciar a VM**
   - Clique em **Iniciar**

Opção B: Via Linha de Comando do Proxmox

```
# SSH para o seu host Proxmox

# Encontre o ID da sua VM
qm list

# Defina multi-queue (substitua XXX pelo seu ID da VM)
qm set XXX -net0 virtio=XX:XX:XX:XX:XX:XX,bridge=vmbr0,queues=8

# Exemplo para a VM 191 com MAC BC:24:11:1D:BA:00
qm set 191 -net0 virtio=BC:24:11:1D:BA:00,bridge=vmbr0,queues=8

# Desligue a VM
qm shutdown XXX

# Aguarde o desligamento e, em seguida, inicie
qm start XXX
```

Recomendações de Contagem de Filas:

- **4 filas**: Mínimo para produção (bom para VMs de 2-4 vCPUs)
- **8 filas**: Recomendado para a maioria das implantações (4-8 vCPUs)
- **16 filas**: Máximo para alto desempenho (8+ vCPUs)

Passo 3: Verificar Multi-Queue Dentro da VM

Após a reinicialização da VM, faça SSH na VM e verifique:

```
# Verifique a configuração da fila
ethtool -l eth0

# Saída esperada:
# Parâmetros de canal para eth0:
# Combinado:    8        <-- Deve corresponder ao seu valor
configurado

# Conte as filas reais
ls -1d /sys/class/net/eth0/queues/rx-* | wc -l
ls -1d /sys/class/net/eth0/queues/tx-* | wc -l

# Ambos devem mostrar 8 (ou seu valor configurado)
```

Passo 4: Habilitar XDP Nativo no OmniUPF

Edite a configuração do OmniUPF:

```
# Edite o arquivo de configuração
sudo nano /etc/omniupf/runtime.exs
```

Altere o modo XDP:

```
# Antes
xdp_attach_mode = "generic"

# Depois
xdp_attach_mode = "native"
```

Reinicie o OmniUPF:

```
sudo systemctl restart omniupf
```

Passo 5: Verificar se o XDP Nativo Está Ativo

Verifique os logs:

```
# Veja os logs de inicialização
journalctl -u omniupf --since "1 minute ago" | grep -i
"xdp\|attach"

# Saída esperada:
# xdp_attach_mode:native
# XDPAttachMode:native
# Programa XDP anexado à iface "eth0" (índice 2)
```

Verifique via API:

```
# Consultar configuração
curl -s http://localhost:8080/api/v1/config | grep xdp_attach_mode

# Saída esperada:
# "xdp_attach_mode": "native",
```

Problemas Comuns no Proxmox

Problema: "Falha ao anexar o programa XDP"

Solução:

- Verifique se o multi-queue está habilitado (`ethtool -l eth0`)
- Verifique a versão do kernel: `uname -r` (deve ser ≥ 5.15)
- Certifique-se de que o driver VirtIO está carregado: `lsmod | grep virtio_net`

Problema: Apenas 1 fila apesar da configuração

Solução:

- A VM deve estar **completamente desligada** (não reiniciada) para alterações na fila
- Use `qm shutdown XXX && sleep 5 && qm start XXX`
- Verifique na configuração do Proxmox: `grep net0 /etc/pve/qemu-server/XXX.conf`

Problema: Desempenho não melhora com o modo nativo

Solução:

- Verifique o pinning de CPU (evite oversubscription)
- Monitore `top` - o uso de CPU deve se espalhar entre os núcleos
- Verifique as estatísticas do XDP: `curl http://localhost:8080/api/v1/xdp_stats`

Habilitando XDP Nativo em Outros Hipervisores

VMware ESXi / vSphere

O VMware usa o driver `vmxnet3`, que suporta XDP nativo.

Requisitos:

- ESXi 6.7 ou posterior
- Versão do driver vmxnet3 1.4.16+ na VM
- Versão de hardware da VM 14 ou posterior

Habilitar Multi-Queue:

1. **Desligue a VM**

2. **Editar configurações da VM**:

- Clique com o botão direito na VM → Editar Configurações
- Adaptador de Rede → Avançado
- Defina **Receive Side Scaling** como **Ativado**

3. **Editar arquivo .vmx** (opcional, para mais filas):

```
ethernet0.pnicFeatures = "4"
ethernet0.multiqueue = "8"
```

4. **Inicie a VM e verifique**:

```
ethtool -l ens192  # Verifique a contagem de filas
```

Configurar OmniUPF (em `runtime.exs`):

```
xdp_interfaces = "ens192"    # O VMware geralmente usa ens192
xdp_attach_mode = "native"
```

KVM / libvirt (Raw)

Habilitar Multi-Queue via virsh:

```
# Edite a configuração da VM
virsh edit seu-nome-da-vm
```

Adicione à seção do interface de rede:

```
<interface type='network'>
  <source network='default'/>
  <model type='virtio'/>
  <driver name='vhost' queues='8'/>
</interface>
```

Reinicie a VM e verifique:

```
ethtool -l eth0
```

Microsoft Hyper-V

O Hyper-V usa o driver `hv_netvsc`, que suporta XDP nativo.

Requisitos:

- Windows Server 2016 ou posterior
- Linux Integration Services 4.3+ na VM
- VM de Geração 2

Habilitar Multi-Queue:

PowerShell no host Hyper-V:

```
# Definir VMQ (Virtual Machine Queue) - multi-queue do Hyper-V
Set-VMNetworkAdapter -VMName "SuaVM" -VrssEnabled $true -
VmmqEnabled $true
```

Configurar OmniUPF (em `runtime.exs`):

```
xdp_interfaces = "eth0"
xdp_attach_mode = "native"
```

VirtualBox

Aviso: O VirtualBox **NÃO** suporta XDP nativo.

Razão: Os drivers de rede do VirtualBox (e1000, virtio-net) não implementam hooks de XDP.

Solução alternativa: Use apenas o modo genérico:

```
xdp_attach_mode = "generic"    # Única opção para VirtualBox
```

Verificando o Modo XDP

Após configurar o XDP nativo, verifique se está funcionando corretamente:

1. Verifique os Logs do OmniUPF

```
# Veja os logs recentes
journalctl -u omniupf --since "5 minutes ago" | grep -i xdp

# Procure por:
# ✓ "xdp_attach_mode:native"
# ✓ "Programa XDP anexado à iface"
# ✗ "Falha ao anexar" ou "retornando ao genérico"
```

2. Verifique via API

```
# Consultar endpoint de configuração
curl -s http://localhost:8080/api/v1/config | jq .xdp_attach_mode

# Saída esperada:
# "native"
```

3. Verifique as Estatísticas do XDP

```
# Veja as estatísticas de processamento do XDP
curl -s http://localhost:8080/api/v1/xdp_stats | jq

# Saída de exemplo:
{
  "xdp_aborted": 0,      # Deve ser 0 (erros)
  "xdp_drop": 1234,      # Pacotes descartados
  "xdp_pass": 5678,      # Passados para a pilha
  "xdp_redirect": 9012,  # Pacotes redirecionados
  "xdp_tx": 3456         # Pacotes transmitidos
}
```

4. Verifique o Suporte do Driver

```
# Verifique se o driver suporta XDP
ethtool -i eth0 | grep driver

# Para Proxmox/KVM: Deve mostrar "virtio_net"
# Para VMware: Deve mostrar "vmxnet3"
# Para Hyper-V: Deve mostrar "hv_netvsc"
```

5. Teste de Desempenho

Compare o processamento de pacotes antes e depois:

```
# Monitore a taxa de pacotes
watch -n 1 'curl -s http://localhost:8080/api/v1/packet_stats | jq
.rx_packets'

# Modo genérico: ~1-2 Mpps
# Modo nativo: ~5-10 Mpps (melhoria de 5-10x)
```

Resolvendo Problemas de XDP

Problema: "Falha ao anexar o programa XDP" na Inicialização

Sintomas:

```
Erro: falha ao anexar o programa XDP à interface eth0
```

Diagnóstico:

1. **Verifique o suporte do driver**:

```
ethtool -i eth0 | grep driver

# Se o driver não for virtio_net/vmxnet3/hv_netvsc, o XDP
nativo não funcionará
```

2. **Verifique a versão do kernel**:

```
uname -r

# Deve ser >= 5.15 para suporte confiável ao XDP
```

3. **Verifique se há programas XDP existentes**:

```
ip link show eth0 | grep xdp

# Se outro programa XDP estiver anexado, descarregue-o primeiro
ip link set dev eth0 xdp off
```

Solução:

- Atualize o kernel para 5.15+ se for mais antigo
- Certifique-se de que o driver virtio_net esteja carregado: `modprobe virtio_net`
- Retorne ao modo genérico se o driver não suportar XDP nativo

---

Problema: Modo Nativo Retorna ao Genérico

Sintomas:

```
Aviso: retornando ao modo XDP genérico
```

Diagnóstico:

Verifique `dmesg` para erros de driver:

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

Causas comuns:

1. **Driver não suporta XDP nativo**:
   - Drivers do VirtualBox (sem suporte a XDP nativo)
   - Drivers de NIC mais antigos
2. **Multi-queue não habilitado**:
   - Verifique: `ethtool -l eth0`
   - Deve mostrar > 1 fila combinada
3. **Suporte ao XDP no kernel desabilitado**:

```
# Verifique se o XDP está habilitado no kernel
grep XDP /boot/config-$(uname -r)

# Deve mostrar:
# CONFIG_XDP_SOCKETS=y
# CONFIG_BPF=y
```

Solução:

- Habilite multi-queue (veja a seção do Proxmox)
- Atualize para um driver suportado
- Recompile o kernel com suporte a XDP, se necessário

Problema: Desempenho Não Melhora com o Modo Nativo

Sintomas: Modo nativo habilitado, mas a taxa de pacotes é a mesma do modo genérico

Diagnóstico:

1. **Verifique a distribuição do multi-queue**:

```
# Verifique as estatísticas por fila
ethtool -S eth0 | grep rx_queue

# O tráfego deve ser distribuído entre múltiplas filas
```

2. **Verifique a utilização da CPU**:

```
# Monitore o uso de CPU por núcleo
mpstat -P ALL 1

# Deve ver carga distribuída entre múltiplas CPUs
```

3. **Verifique se o XDP está realmente rodando no modo nativo**:

```
# Verifique bpftool (se disponível)
sudo bpftool net list

# Deve mostrar XDP anexado à interface
```

Solução:

- Aumente a contagem de filas (8-16 filas)
- Habilite o pinning de CPU para evitar migração de núcleo
- Verifique se há oversubscription de CPU no hipervisor

Problema: Programa XDP Abortado (xdp_aborted > 0)

Sintomas:

```
curl http://localhost:8080/api/v1/xdp_stats
{
  "xdp_aborted": 1234,  # Não zero indica erros
  ...
}
```

Diagnóstico:

XDP abortado significa que o programa eBPF encontrou um erro durante a execução.

1. **Verifique os logs do verificador eBPF**:

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

2. **Verifique os limites de tamanho do mapa**:

   ```
   # Os mapas eBPF podem estar cheios
   curl http://localhost:8080/api/v1/map_info

   # Procure mapas em 100% de capacidade
   ```

Solução:

- Aumente os tamanhos dos mapas eBPF na configuração
- Verifique se há pacotes corrompidos causando erros no eBPF
- Verifique se o suporte ao eBPF no kernel Linux está completo

---

Problema: Multi-Queue Não Funciona no Proxmox

Sintomas: `ethtool -l eth0` mostra apenas 1 fila apesar da configuração

Diagnóstico:

1. **Verifique a configuração da VM do Proxmox**:

```
# No host Proxmox
grep net0 /etc/pve/qemu-server/YOUR_VM_ID.conf

# Deve mostrar: queues=8
```

2. **Verifique se a VM estava completamente desligada**:

```
# No host Proxmox
qm status YOUR_VM_ID

# Deve mostrar "status: stopped" antes de iniciar
```

Solução:

```
# No host Proxmox
# Forçar desligamento e reiniciar
qm shutdown YOUR_VM_ID
sleep 10
qm start YOUR_VM_ID

# Em seguida, verifique dentro da VM
ethtool -l eth0
```

Importante: Alterações na contagem de filas requerem um **desligamento completo da VM**, não apenas uma reinicialização de dentro da VM.

Problema: Permissão Negada ao Anexar XDP

Sintomas:

```
Erro: permissão negada ao anexar o programa XDP
```

Diagnóstico:

As operações do XDP requerem as capacidades `CAP_NET_ADMIN` e `CAP_SYS_ADMIN`.

Solução:

1. **Execute o OmniUPF como root** (ou com capacidades):

   ```
   sudo systemctl restart omniupf
   ```

2. **Se estiver usando systemd**, verifique se o arquivo de serviço tem capacidades:

   ```
   # /lib/systemd/system/omniupf.service
   [Service]
   CapabilityBoundingSet=CAP_NET_ADMIN CAP_SYS_ADMIN CAP_NET_RAW
   AmbientCapabilities=CAP_NET_ADMIN CAP_SYS_ADMIN CAP_NET_RAW
   ```

3. **Se estiver usando Docker**, execute com `--privileged`:

   ```
   docker run --privileged -v /sys/fs/bpf:/sys/fs/bpf ...
   ```

Resumo do Impacto no Desempenho

Comparação de desempenho no mundo real para o processamento de pacotes do OmniUPF:

Cenário	Modo Genérico	Modo Nativo	Melhoria
Taxa de Pacotes	1.5 Mpps	8.2 Mpps	**5.5x mais rápido**
Latência	95 µs	12 µs	**8x menor**
Uso de CPU (1 Gbps)	85% (1 núcleo)	15% (distribuído)	**5x mais eficiente**
Throughput Máximo	~1.2 Gbps	~10 Gbps	**8x maior**

Recomendação: Sempre use **modo nativo** com **multi-queue habilitado** para implantações em produção.

Recomendações de Hardware para XDP

⚠ **IMPORTANTE: Antes de comprar qualquer hardware, consulte o suporte da Omnitouch para confirmar que é 100% compatível com sua configuração e requisitos de implantação específicos.**

NICs Conhecidas que Suportam XDP Nativo

Essas NICs estão verificadas para suportar o modo XDP nativo com o OmniUPF:

NICs Intel (Recomendadas para Bare Metal)

Modelo	Velocidade	Driver	Suporte a XDP	Notas
Intel X520	10GbE	ixgbe	Nativo ✓	Comprovada, amplamente disponível, bom custo/desempenho
Intel X710	10/40GbE	i40e	Nativo ✓	Excelente suporte a multi-queue
Intel E810	100GbE	ice	Nativo ✓	Última geração, melhor desempenho
Intel i350	1GbE	igb	Nativo ✓ (kernel 5.10+)	Bom para necessidades de largura de banda mais baixa

Mellanox/NVIDIA NICs (Alto Desempenho)

Modelo	Velocidade	Driver	Suporte a XDP	Notas
ConnectX-4	25/50/100GbE	mlx5	Nativo ✓	Alta largura de banda, bom para computação de borda
ConnectX-5	25/50/100GbE	mlx5	Nativo ✓	Excelente desempenho, aceleração de hardware
ConnectX-6	50/100/200GbE	mlx5	Nativo ✓	Última geração, melhor para ultra-alta largura de banda
BlueField-2	100/200GbE	mlx5	Nativo ✓	SmartNIC com capacidades de DPU

Broadcom NICs

Modelo	Velocidade	Driver	Suporte a XDP	Notas
Série BCM57xxx	10/25/50GbE	bnxt_en	Nativo ✓	Comum em servidores Dell/HP

NICs Virtuais (Implantações de VM)

Plataforma	Tipo de NIC	Driver	Suporte a XDP	Multi-Queue	N
Proxmox/KVM	VirtIO	virtio_net	Nativo ✓	Sim (configurável)	**Melh VMs**
VMware ESXi	vmxnet3	vmxnet3	Nativo ✓	Sim	Requ 6.7+
Hyper-V	NIC Sintético	hv_netvsc	Nativo ✓	Sim	Wind Serve 2016
AWS	ENA	ena	Nativo ✓	Sim	Instâ EC2 r
VirtualBox	Qualquer	vários	Genérico apenas □	Não	Não recon para produ

NICs com Suporte a Offload de Hardware

Offload de hardware XDP verdadeiro (eBPF roda no NIC):

Fornecedor	Modelo	Velocidade	Notas
Netronome	Agilio CX 10G	10GbE	Apenas suporte a offload XDP confirmado
Netronome	Agilio CX 25G	25GbE	Requer firmware especial
Netronome	Agilio CX 40G	40GbE	Muito caro (~$2,500-5,000)
Netronome	Agilio CX 100G	100GbE	Apenas para nível empresarial

Nota: NICs com offload de hardware são raros, caros e requerem implantação bare metal. A maioria das implantações deve usar XDP nativo em vez disso.

Configurações Testadas

Essas configurações foram verificadas com OmniUPF em produção:

Opção Econômica (1-10 Gbps)

- **NIC**: Intel X520 (10GbE de porta dupla)
- **Modo**: XDP Nativo
- **Largura de Banda**: ~8-10 Gbps por instância UPF
- **Custo**: ~$100-200 (usado/refurbished)

Médio Alcance (10-50 Gbps)

- **NIC**: Intel X710 (40GbE) ou Mellanox ConnectX-4 (25GbE)
- **Modo**: XDP Nativo
- **Largura de Banda**: ~25-40 Gbps por instância UPF
- **Custo**: ~$300-800

Alto Desempenho (50-100+ Gbps)

- **NIC**: Mellanox ConnectX-5/6 (100GbE)
- **Modo**: XDP Nativo
- **Largura de Banda**: ~80-100 Gbps por instância UPF
- **Custo**: ~$1,000-2,500

Implantações de VM (Proxmox/KVM)

- **NIC**: VirtIO com 8-16 filas
- **Modo**: XDP Nativo
- **Largura de Banda**: ~5-10 Gbps por instância UPF
- **Custo**: Sem custo adicional de hardware

O QUE NÃO COMPRAR

Evite esses para implantações de OmniUPF em produção:

NIC/Plataforma	Motivo	Alternativa
NICs Realtek	Sem suporte a XDP, drivers Linux ruins	Intel i350 ou melhor
VirtualBox	Sem suporte nativo a XDP	Migrar para Proxmox/KVM
NICs de consumo	Suporte a filas limitado, não confiável	Intel/Mellanox de nível servidor
NICs muito antigas (<2014)	Sem suporte a driver XDP	Intel X520 ou mais recente

Lista de Verificação Pré-Compra

Antes de comprar hardware, verifique:

1. □ **Suporte a Driver**: Verifique se o driver Linux suporta XDP

```
# Em sistema semelhante
modinfo <driver_name> | grep -i xdp
```

2. □ **Versão do Kernel**: Garantir kernel ≥ 5.15 para XDP confiável

```
uname -r
```

3. □ **Multi-Queue**: Verifique se o NIC suporta várias filas (RSS/VMDq)

4. □ **Largura de Banda PCI**: Garantir que o slot PCIe tenha pistas suficientes

   - 10GbE: PCIe 2.0 x4 mínimo
   - 40GbE: PCIe 3.0 x8 mínimo
   - 100GbE: PCIe 3.0 x16 ou PCIe 4.0 x8

5. □ **Tipo de Implantação**:

   - Bare metal: NIC físico necessário
   - VM: Suporte a VirtIO ou SR-IOV necessário
   - Container: Configuração do NIC do host herdada

⚠ **Não compre hardware apenas com base neste guia - sempre confirme com o suporte da Omnitouch primeiro!**

Recursos Adicionais

- **Guia de Configuração**: CONFIGURATION.md - Referência completa de configuração
- **Guia de Solução de Problemas**: TROUBLESHOOTING.md - Diagnóstico abrangente de problemas
- **Guia de Arquitetura**: ARCHITECTURE.md - Detalhes da arquitetura eBPF e XDP
- **Guia de Monitoramento**: MONITORING.md - Monitoramento de desempenho e estatísticas

---

Referência Rápida

Configuração XDP Nativa do Proxmox (TL;DR)

```
# No host Proxmox:
qm set <VM_ID> -net0 virtio=<MAC>,bridge=vmbr0,queues=8
qm shutdown <VM_ID> && sleep 10 && qm start <VM_ID>

# Dentro da VM:
ethtool -l eth0  # Verifique 8 filas
sudo nano /etc/omniupf/runtime.exs  # Defina: xdp_attach_mode =
"native"
sudo systemctl restart omniupf
journalctl -u omniupf --since "1 min ago" | grep xdp  # Verifique
o modo nativo
```

Verifique se o Modo XDP está Ativo

```
# Verifique a configuração
curl -s http://localhost:8080/api/v1/config | grep xdp_attach_mode

# Verifique as estatísticas
curl -s http://localhost:8080/api/v1/xdp_stats | jq

# Verifique as filas
ethtool -l eth0
```

Documentação da API OmniUPF

Visão Geral

A API OmniUPF fornece uma interface RESTful abrangente para gerenciar e monitorar a Função de Plano do Usuário baseada em eBPF. A API permite controle em tempo real e observabilidade de todos os componentes do UPF.

Capacidades da API

Gerenciamento de Sessões:

- **Sessões PFCP**: Consultar sessões ativas, visualizar detalhes da sessão, filtrar por IP do UE ou TEID
- **Associações PFCP**: Monitorar associações e status dos nós do plano de controle

Regras de Tráfego:

- **Regras de Detecção de Pacotes (PDR)**: Inspecionar classificadores de tráfego de uplink e downlink (IPv4/IPv6)
- **Regras de Ação de Encaminhamento (FAR)**: Visualizar políticas de encaminhamento, buffer e descarte
- **Regras de Aplicação de QoS (QER)**: Monitorar limitação de taxa e políticas de QoS
- **Regras de Relatório de Uso (URR)**: Rastrear contadores de volume de dados por sessão

Buffer de Pacotes:

- **Status do Buffer**: Visualizar pacotes armazenados por FAR (`GET /buffer`, `GET /buffer/:far_id`)

- **Operações de Buffer**: Limpar ou esvaziar pacotes armazenados (`POST /buffer/:far_id/flush`, `DELETE /buffer/:far_id`, `DELETE /buffer`)
- **Controle de Buffer**: Acionamento manual de notificações (`POST /buffer/:far_id/notify`)
- **Status de Notificação**: Visualizar estado de notificação DLDR (`GET /buffer/notifications`)

Monitoramento e Estatísticas:

- **Estatísticas de Pacotes**: Contadores de pacotes em tempo real por protocolo (GTP, IP, TCP, UDP, ICMP, ARP)
- **Estatísticas XDP**: Métricas de desempenho do datapath (passar, descartar, redirecionar, abortar)
- **Estatísticas de Interface N3/N6**: Distribuição de tráfego da RAN e da Rede de Dados
- **Estatísticas de Rota**: Desempenho de pesquisa FIB (acertos de cache, pesquisas, erros)

Gerenciamento de Rotas:

- **Rotas do UE**: Consultar tabela de roteamento de IP do UE para gNB (`GET /routes`)
- **Integração FRR**: Sincronizar rotas com o daemon Free Range Routing (`POST /routes/sync`)
- **Sessões de Roteamento**: Visualizar sessões de protocolo de roteamento (`GET /routing/sessions`)
- **Banco de Dados OSPF**: Consultar banco de dados de rotas externas OSPF (`GET /ospf/database/external`)

Configuração:

- **Configuração do UPF**: Recuperar e editar configuração (`GET /config`, `POST /config`)
- **Configuração do Dataplane**: Consultar configuração específica do dataplane (`GET /dataplane_config`)
- **Capacidades XDP**: Consultar suporte ao modo XDP e capacidades da interface (`GET /xdp_capabilities`)

- **Capacidade do Mapa eBPF**: Monitorar utilização de recursos e capacidade (`GET /map_info`)

Integração com a Web UI

A Web UI do OmniUPF é construída sobre esta API e fornece um painel interativo para toda a funcionalidade da API. Veja o Guia da Web UI para capturas de tela e exemplos de uso.

Documentação da API Swagger

A API está totalmente documentada usando a especificação **OpenAPI 3.0 (Swagger)**. A UI interativa do Swagger fornece:

- Documentação completa dos endpoints com esquemas de solicitação/resposta
- Funcionalidade de teste para testar chamadas da API diretamente do navegador
- Definições de esquema para todos os modelos de dados
- Códigos de status HTTP e respostas de erro

UI interativa do Swagger mostrando os endpoints da API OmniUPF com documentação detalhada.

Acessando a UI do Swagger

A documentação do Swagger está disponível em:

```
http://<upf-host>:8080/swagger/index.html
```

Por exemplo: `http://10.98.0.20:8080/swagger/index.html`

Caminho Base da API

Todos os endpoints da API são prefixados com:

```
/api/v1
```

Recursos da API

Paginação

A API OmniUPF suporta paginação para endpoints que retornam grandes conjuntos de dados. A paginação previne timeouts e reduz o uso de memória ao consultar milhares de sessões, PDRs ou URRs.

Estilos de Paginação Suportados:

1. **Paginação baseada em páginas** (recomendada):
   - `page`: Número da página (começando de 1)
   - `page_size`: Itens por página (padrão: 100, máximo: 1000)
2. **Paginação baseada em offset**:
   - `offset`: Número de itens a serem pulados
   - `limit`: Número de itens a serem retornados (máx: 1000)

Exemplos de Solicitações:

```
# Baseada em página: Obter a segunda página com 50 itens por
página
GET /api/v1/pfcp_sessions?page=2&page_size=50

# Baseada em offset: Pular os primeiros 100 itens, retornar os
próximos 50
GET /api/v1/pfcp_sessions?offset=100&limit=50

# Comportamento padrão (sem parâmetros de paginação): Primeiros
100 itens
GET /api/v1/pfcp_sessions
```

Formato de Resposta:

```
{
  "data": [
    { /* objeto de sessão */ },
    { /* objeto de sessão */ },
    ...
  ],
  "pagination": {
    "total": 5432,
    "page": 2,
    "page_size": 50,
    "total_pages": 109
  }
}
```

Endpoints Paginated:

- `/api/v1/pfcp_sessions` - Lista de sessões PFCP
- `/api/v1/pfcp_associations` - Lista de associações PFCP
- `/api/v1/routes` - Rotas de IP do UE
- `/api/v1/uplink_pdr_map` - PDRs de uplink (informações básicas)
- `/api/v1/uplink_pdr_map/full` - PDRs de uplink com detalhes completos do filtro SDF
- `/api/v1/downlink_pdr_map` - PDRs de downlink IPv4 (informações básicas)

- `/api/v1/downlink_pdr_map/full` - PDRs de downlink IPv4 com detalhes completos do filtro SDF
- `/api/v1/downlink_pdr_map_ip6` - PDRs de downlink IPv6 (informações básicas)
- `/api/v1/downlink_pdr_map_ip6/full` - PDRs de downlink IPv6 com detalhes completos do filtro SDF
- `/api/v1/far_map` - Regras de Ação de Encaminhamento
- `/api/v1/qer_map` - Regras de Aplicação de QoS
- `/api/v1/urr_map` - Regras de Relatório de Uso

Endpoints de Gerenciamento de Buffer:

- `GET /api/v1/buffer` - Listar todos os buffers FAR com estatísticas
- `GET /api/v1/buffer/:far_id` - Obter status do buffer para um FAR específico
- `GET /api/v1/buffer/notifications` - Listar status de notificação DLDR
- `DELETE /api/v1/buffer` - Limpar todos os pacotes armazenados
- `DELETE /api/v1/buffer/:far_id` - Limpar buffer para um FAR específico
- `POST /api/v1/buffer/:far_id/flush` - Limpar (repetir) pacotes armazenados
- `POST /api/v1/buffer/:far_id/notify` - Enviar manualmente notificação DLDR

Endpoints de Configuração:

- `GET /api/v1/config` - Obter configuração atual do UPF
- `POST /api/v1/config` - Atualizar configuração do UPF (campos editáveis em tempo de execução)
- `GET /api/v1/dataplane_config` - Obter configuração específica do dataplane

Endpoints de Integração de Roteamento:

- `GET /api/v1/routes` - Listar rotas do UE
- `POST /api/v1/routes/sync` - Acionar sincronização de rotas com FRR

- `GET /api/v1/routing/sessions` - Obter sessões de protocolo de roteamento
- `GET /api/v1/ospf/database/external` - Obter banco de dados de LSA externo OSPF

Melhores Práticas:

- Use `page_size=100` para exibição na Web UI
- Use `page_size=1000` para exportações em massa (limite máximo)
- Consulte `pagination.total_pages` para determinar a contagem de iterações
- Aumente `page_size` para melhor desempenho da API (menos solicitações)

Suporte a CORS

O Compartilhamento de Recursos de Origem Cruzada (CORS) está habilitado por padrão para todos os endpoints da API, permitindo que a Web UI e aplicativos de terceiros consumam a API de diferentes origens.

Métricas Prometheus

Além da API REST, o OmniUPF expõe métricas Prometheus no endpoint `/metrics` (porta padrão `:9090`).

As métricas fornecem:

- Contadores de mensagens PFCP e latência por par
- Estatísticas de pacotes por tipo de protocolo
- Veredictos de ação XDP
- Estatísticas de buffer
- Utilização da capacidade do mapa eBPF
- Rastreamento de volume URR

Veja a Referência de Métricas para documentação completa.

Documentação Relacionada

- **Guia da Web UI** - Painel interativo construído sobre esta API
- **Referência de Métricas** - Documentação de métricas Prometheus
- **Códigos de Causa PFCP** - Códigos de erro PFCP e solução de problemas
- **Guia de Gerenciamento de Regras** - Configuração de PDR, FAR, QER, URR
- **Guia de Gerenciamento de Rotas** - Integração FRR e roteamento do UE
- **Guia de Monitoramento** - Monitoramento de estatísticas e planejamento de capacidade
- **Guia de Configuração** - Opções de configuração do UPF
- **Swagger UI** - Documentação interativa da API (substitua `localhost` pelo seu host UPF)

Gerenciamento de Rota UE

> **Documentação Relacionada**:
>
> - Documentação da API - Referência completa da API incluindo endpoints de gerenciamento de rota
> - Guia de Operações - Operações e monitoramento da interface web

Visão Geral

O UPF (Função de Plano do Usuário) integra-se com **FRR (Free Range Routing)** para gerenciar dinamicamente as rotas IP do Equipamento do Usuário (UE). Essa integração garante que, à medida que as sessões de UE são estabelecidas ou encerradas, a infraestrutura de roteamento se adapte automaticamente para refletir a topologia atual da rede.

O que é FRR?

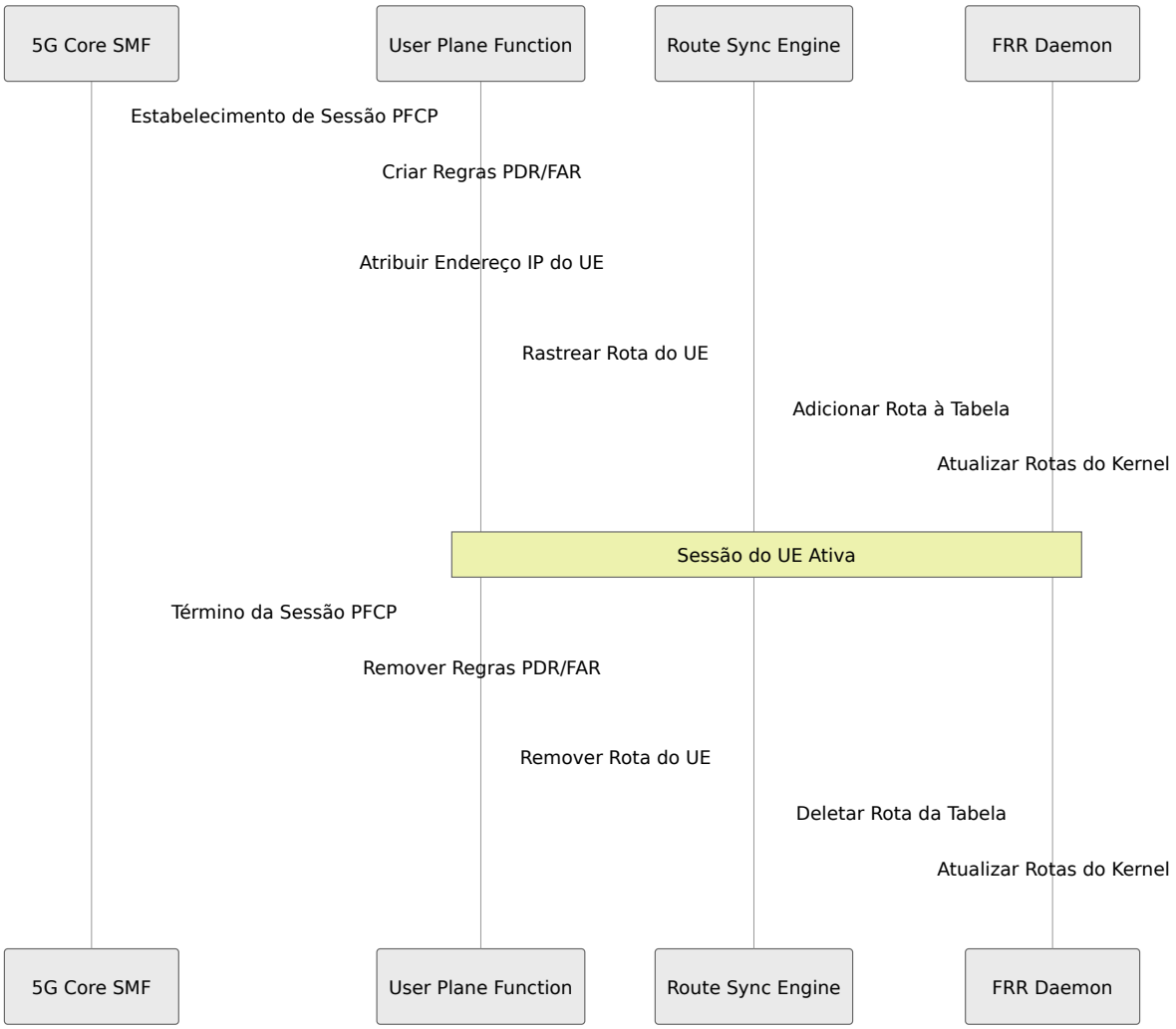
FRR (Free Range Routing) é um robusto conjunto de protocolos de roteamento de código aberto para plataformas Linux e Unix. Ele implementa vários protocolos de roteamento, incluindo BGP, OSPF, RIP e outros. Em nossa implantação, o FRR atua como o daemon de roteamento que mantém a tabela de roteamento do kernel e pode redistribuir rotas para outros elementos da rede.

Arquitetura



Como Funciona a Sincronização de Rotas

Ciclo de Vida da Rota



Sincronização Automática

O UPF mantém um registro interno de todos os endereços IP de UE ativos. Quando habilitado, o sistema de sincronização de rotas:

1. **Monitora Sessões de UE**: Rastreia todas as sessões PFCP ativas e seus endereços IP de UE associados
2. **Mantém Lista de Rotas**: Mantém uma lista atualizada de rotas que precisam estar na tabela de roteamento

3. **Sincroniza com FRR**: Envia automaticamente atualizações de rotas para o daemon FRR via sua API
4. **Lida com Falhas**: Rastreia o status de sincronização (sincronizado/falhou) para cada rota e tenta novamente conforme necessário

Configuração do FRR

Configuração

O FRR é implantado e configurado usando **templates Ansible** para estabelecer os parâmetros básicos de roteamento. Você define a configuração do FRR uma vez como um **template Jinja2** em seu playbook Ansible, e o Ansible a propaga automaticamente para todas as suas instâncias de UPF durante a implantação.

Um template típico de configuração Jinja2 do FRR inclui:

```
frr version 7.2.1
frr defaults traditional
hostname pgw02
log syslog informational
service integrated-vtysh-config
!
ip route {{ hostvars[inventory_hostname]['ansible_default_ipv4']
['gateway'] }}/32 {{ ansible_default_ipv4['interface'] }}
!
interface {{ ansible_default_ipv4['interface'] }}
 ip address ospf router-id {{hostvars[inventory_hostname]
['ansible_host']}}
 ip ospf authentication null
!
router ospf
 ospf router-id {{hostvars[inventory_hostname]['ansible_host']}}
 redistribute static
 network {{ hostvars[inventory_hostname]['ansible_default_ipv4']
['network'] }}/{{ mask_cidr }} area 0
 area 0 authentication message-digest
!
line vty
!
end
```

Modelo de Implantação:

1. **Defina Uma Vez**: Crie o template Jinja2 do FRR em seu papel Ansible (por exemplo, `roles/frr/templates/frr.conf.j2`)
2. **Configure Parâmetros**: Defina variáveis em seu inventário Ansible para cada host UPF
3. **Implante em Todos os Lugares**: Execute o playbook Ansible para implantar a configuração do FRR em todos os nós UPF
4. **Personalização Automática**: O Ansible usa variáveis específicas do host (endereços IP, IDs de roteador, etc.) para personalizar a configuração do FRR de cada UPF

Parâmetros Personalizáveis no template Jinja2:

- **Parâmetros OSPF**: ID do Roteador, configuração de área, métodos de autenticação, anúncios de rede
- **Configuração BGP**: ASN, relacionamentos de vizinhança, políticas de rota, comunidades
- **Redistribuição de Rotas**: Quais rotas redistribuir (por exemplo, `redistribute static` para rotas de UE)
- **Filtragem de Rotas**: Mapas de rota, listas de prefixo, listas de acesso
- **Configurações de Interface**: Parâmetros de interface OSPF/BGP

Integração UPF: Uma vez que a configuração básica do FRR é implantada em cada instância de UPF, o UPF adiciona dinamicamente endereços IP de UE como **rotas de host** (/32 para IPv4, /128 para IPv6) através da interface vtysh do FRR com base nas sessões PFCP ativas. Essas rotas são então:

1. **Adicionadas como rotas estáticas do FRR** pelo mecanismo de sincronização de rotas do UPF (via vtysh)
2. **Capturadas pelo FRR** através da diretiva `redistribute static`
3. **Anunciadas para protocolos de roteamento** (OSPF, BGP) de acordo com sua configuração do FRR
4. **Propagadas para a rede** para que o tráfego de UE possa ser roteado para esta instância de UPF

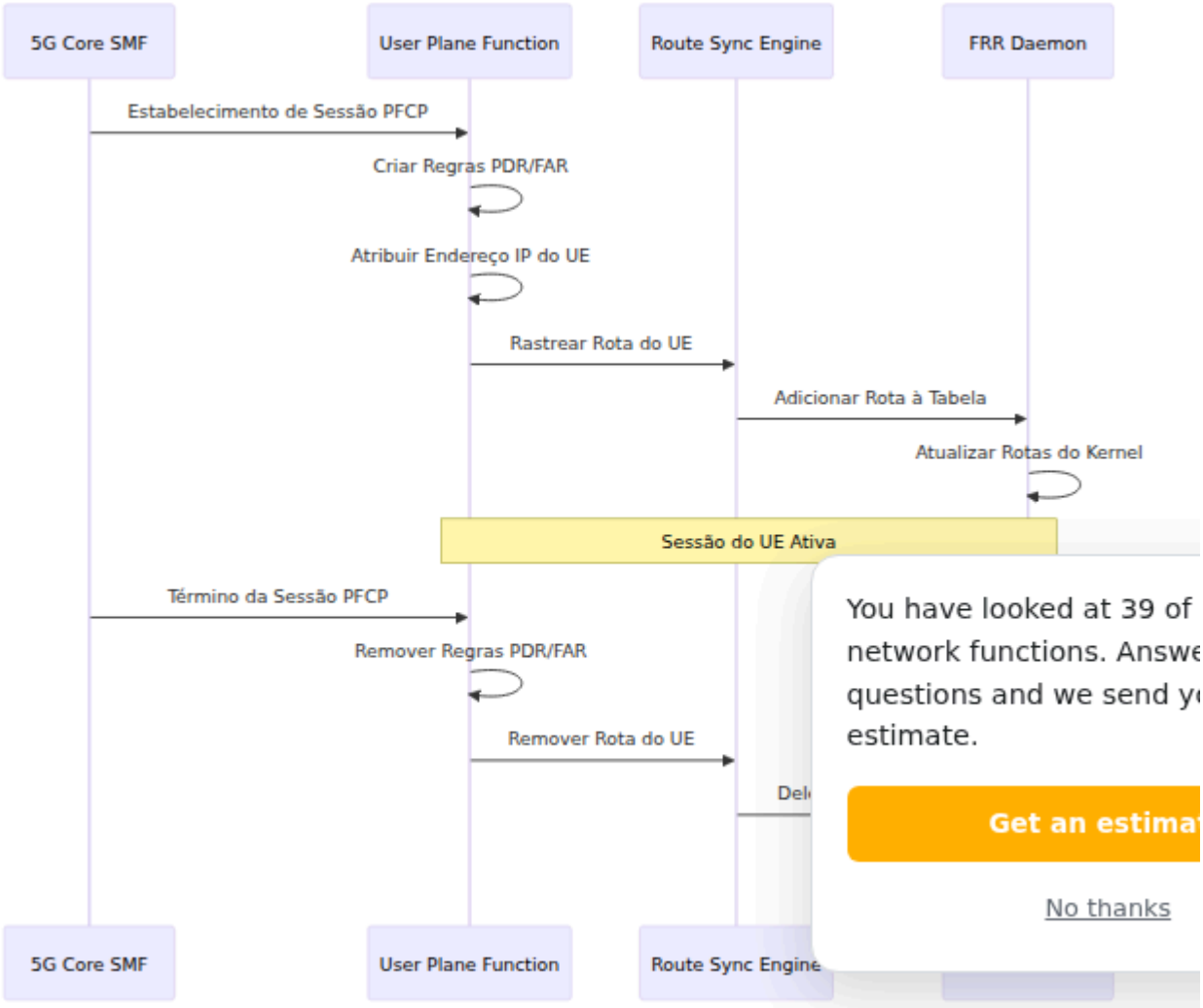
> **Importante**: O UPF adiciona rotas através da interface vtysh do FRR, tornando-as rotas estáticas do FRR (não rotas do kernel). Você deve usar `redistribute static` em sua configuração OSPF/BGP, não `redistribute kernel`.

Pontos-chave:

- **Defina Uma Vez, Implemente em Todos os Lugares**: Defina o template Jinja2 do FRR uma vez no Ansible, e ele será automaticamente implantado em todas as instâncias de UPF
- **Ansible gerencia configuração estática**: O template Jinja2 configura todos os parâmetros do protocolo de roteamento (áreas OSPF, vizinhos BGP, autenticação, políticas de rota, etc.)
- **UPF gerencia rotas dinâmicas**: Cada instância de UPF gerencia dinamicamente apenas as rotas IP /32 do UE com base em suas sessões PFCP ativas

- **Anúncio automático de rotas**: O FRR em cada UPF redistribui automaticamente as rotas locais de UE de acordo com suas políticas configuradas
- **Gerenciamento centralizado**: Atualize o template Ansible e execute novamente o playbook para alterar a configuração de roteamento em todos os UPFs simultaneamente

Anúncio de Rotas



Monitoramento e Gerenciamento

Integração com a Interface Web

O Painel de Controle do UPF fornece uma página de **Rotas** que exibe:

- **Status da Rota**: Se a sincronização de rotas está habilitada ou desabilitada
- **Total de Rotas**: Número de endereços IP de UE sendo rastreados
- **Estatísticas de Sincronização**: Contagem de rotas sincronizadas com sucesso e quaisquer falhas
- **Rotas Ativas**: Lista em tempo real de todos os endereços IP de UE atualmente na tabela de roteamento
- **Vizinhos OSPF**: Status ao vivo das adjacências OSPF com detalhes dos vizinhos
- **Pares BGP**: Status da sessão BGP e estatísticas de prefixo (quando configurado)
- **Rotas Redistribuídas OSPF**: Visão completa dos LSAs externos mostrando como as rotas de UE são anunciadas

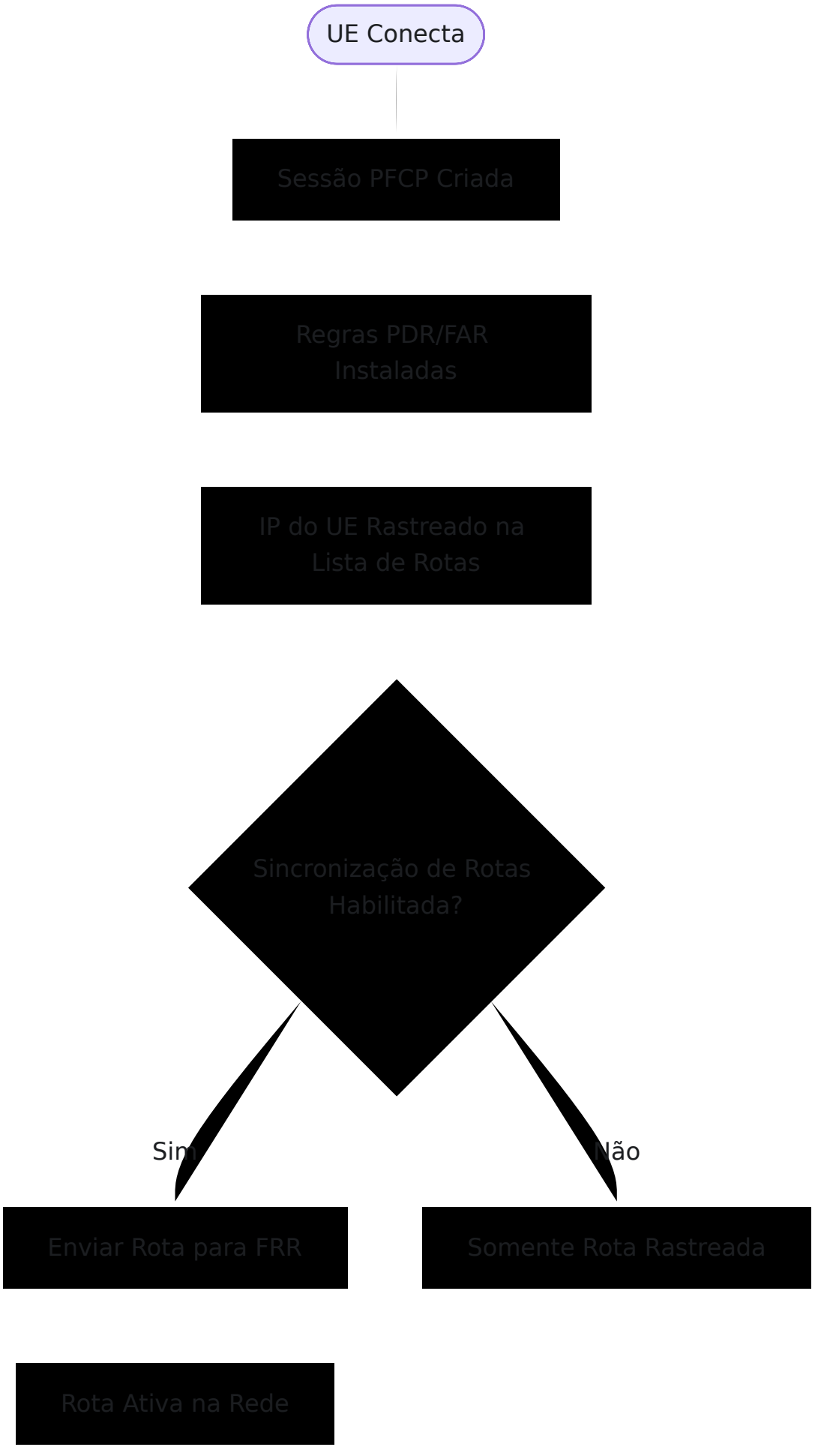
A página de Rotas fornece visibilidade abrangente sobre a sincronização de rotas de UE, vizinhos de protocolos de roteamento e anúncios de rotas redistribuídas.

Operação de Sincronização Manual

Os administradores podem acionar uma sincronização manual de rotas através da interface web usando o botão **Sincronizar Rotas**. Esta operação:

1. Lê novamente a lista atual de sessões de UE ativas do UPF
2. Compara com a tabela de roteamento do FRR
3. Adiciona quaisquer rotas ausentes
4. Remove quaisquer rotas obsoletas
5. Retorna estatísticas de sincronização atualizadas

Fluxo de Rotas


UE Conecta
Sessão PFCP Criada
Regras PDR/FAR Instaladas
IP do UE Rastreado na Lista de Rotas
Sincronização de Rotas Habilitada?
Sim
Não
Enviar Rota para FRR
Somente Rota Rastreada
Rota Ativa na Rede



Benefícios

- **Provisionamento Sem Toque**: As rotas são gerenciadas automaticamente sem intervenção manual
- **Adaptação Dinâmica**: O roteamento da rede se adapta em tempo real à mobilidade do UE e mudanças de sessão
- **Escalabilidade**: Suporta milhares de rotas de UE simultâneas
- **Resiliência**: Operações de sincronização com falha são rastreadas e podem ser tentadas novamente
- **Visibilidade**: Visibilidade total sobre o status da rota através da interface web

Detalhes Técnicos

Endpoints da API

O UPF expõe os seguintes endpoints de gerenciamento de rotas:

- `GET /api/v1/routes` - Lista todas as rotas de UE rastreadas sem sincronização
- `POST /api/v1/routes/sync` - Sincroniza rotas com o FRR e retorna lista atualizada
- `GET /api/v1/route_stats` - Obtém estatísticas detalhadas de roteamento
- `GET /api/v1/routing/sessions` - Obtém sessões de protocolo de roteamento (vizinhos OSPF, pares BGP)
- `GET /api/v1/ospf/database/external` - Obtém o banco de dados OSPF AS-External LSA (rotas redistribuídas)

> **Veja Também**: Documentação da API - Gerenciamento de Rotas para detalhes completos dos endpoints e exemplos

Formato de Rota

As rotas são armazenadas e gerenciadas como endereços IP simples (por exemplo, `100.64.18.5`). O daemon de roteamento lida com todos os detalhes da entrada de rota, incluindo:

- Prefixo/máscara de destino
- Gateway/próximo salto
- Vinculação de interface

- Métrica e distância administrativa

Suporte a IPv6

O gerenciador de rotas suporta endereços de UE tanto IPv4 quanto IPv6:

Tipo de Endereço	Comprimento do Prefixo	Exemplo
IPv4	/32	`100.64.18.5/32`
IPv6	/128	`2001:db8::1/128`

Para IPv6, certifique-se de que sua configuração do FRR inclua a redistribuição OSPFv3 ou BGP IPv6 apropriada:

```
router ospf6
 redistribute static
```

ou para BGP:

```
router bgp <asn>
 address-family ipv6 unicast
  redistribute static
```

Verificação do FRR

Banco de Dados OSPF LSA Externo

Você pode verificar se as rotas de UE estão sendo redistribuídas corretamente no OSPF examinando o Banco de Dados de Estado de Link OSPF do FRR. LSAs externas (Tipo 5) mostram rotas que foram injetadas no OSPF a partir de fontes externas.

Banco de dados OSPF do FRR mostrando LSAs externas incluindo a rota de UE 100.64.18.5/32 sendo anunciada como uma rota E2 (Tipo Externo 2).

No exemplo acima, você pode ver:

- **Network LSA (10.98.0.20)**: O próprio anúncio de rede do UPF
- **Router LSA (192.168.1.1)**: Anúncio do roteador OSPF
- **LSAs Externas**: Incluindo a rota de UE `100.64.18.5` redistribuída no OSPF com métrica tipo E2 (Tipo Externo 2)

Essa verificação confirma que:

1. O UPF está rastreando com sucesso o endereço IP do UE
2. O mecanismo de sincronização de rotas enviou a rota para o FRR
3. O FRR redistribuiu a rota no OSPF
4. Os vizinhos OSPF estão recebendo os anúncios de rota