

# 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

### Bufferização 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 Bufferização:** Acionamento manual de notificação (POST `/buffer/:far_id/notify`)
- **Status de Notificação:** Visualizar estado da 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 da Interface N3/N6:** Distribuição de tráfego da RAN e da Rede de Dados
- **Estatísticas de Rota:** Desempenho de lookup FIB (acertos de cache, buscas, 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. Consulte 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 interface 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

The screenshot displays the Swagger UI for the OmniUPF API. At the top, there is a search bar containing 'doc.json' and an 'Explore' button. Below the search bar, the API title 'OmniUPF API 1.0' is shown, along with the base URL '[ Base URL: /api/v1 ]' and a link to 'doc.json'. The API description is 'OmniUPF - eBPF User Plane Function API'. There are also links for 'API Support - Website', 'Send email to API Support', and 'Apache 2.0'. The main content area is titled 'buffer' and lists five endpoints:

- GET** /buffer: List all packet buffers
- DELETE** /buffer: Clear all buffers
- GET** /buffer/{far\_id}: Get buffer status for a specific FAR
- DELETE** /buffer/{far\_id}: Clear buffer for a specific FAR
- POST** /buffer/{far\_id}/flush: Flush buffer for a specific FAR

*Interface 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 página (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:

```
&#123;
  "data": [
    &#123; /* objeto de sessão */ &#125;,
    &#123; /* objeto de sessão */ &#125;,
    ...
  ],
  "pagination": &#123;
    "total": 5432,
    "page": 2,
    "page_size": 50,
    "total_pages": 109
  &#125;
&#125;
```

### Endpoints Paginação:

- `/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 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 FAR específico
- `POST /api/v1/buffer/:far_id/flush` - Esvaziar (reproduzir) 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 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 do Prometheus

Além da API REST, o OmniUPF expõe métricas do 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

Consulte 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 do 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)

# 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 Buffering](#)
7. [Aplicação de QoS](#)
8. [Características de Desempenho](#)
9. [Escalabilidade e Ajustes](#)

## Visão Geral

OmniUPF aproveita o eBPF (filtro de pacotes Berkeley estendido) e XDP (caminho de dados eXpress) para alcançar desempenho de nível de operadora para processamento de pacotes 5G/LTE. Ao executar a lógica de processamento de pacotes diretamente no kernel do Linux, o OmniUPF elimina a sobrecarga do processamento em espaço de usuário e alcança uma taxa de transferência de múltiplos gigabits com latência de microssegundos.



- 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 a modos nativos do driver ou genéricos

## **Fundação da Tecnologia eBPF**

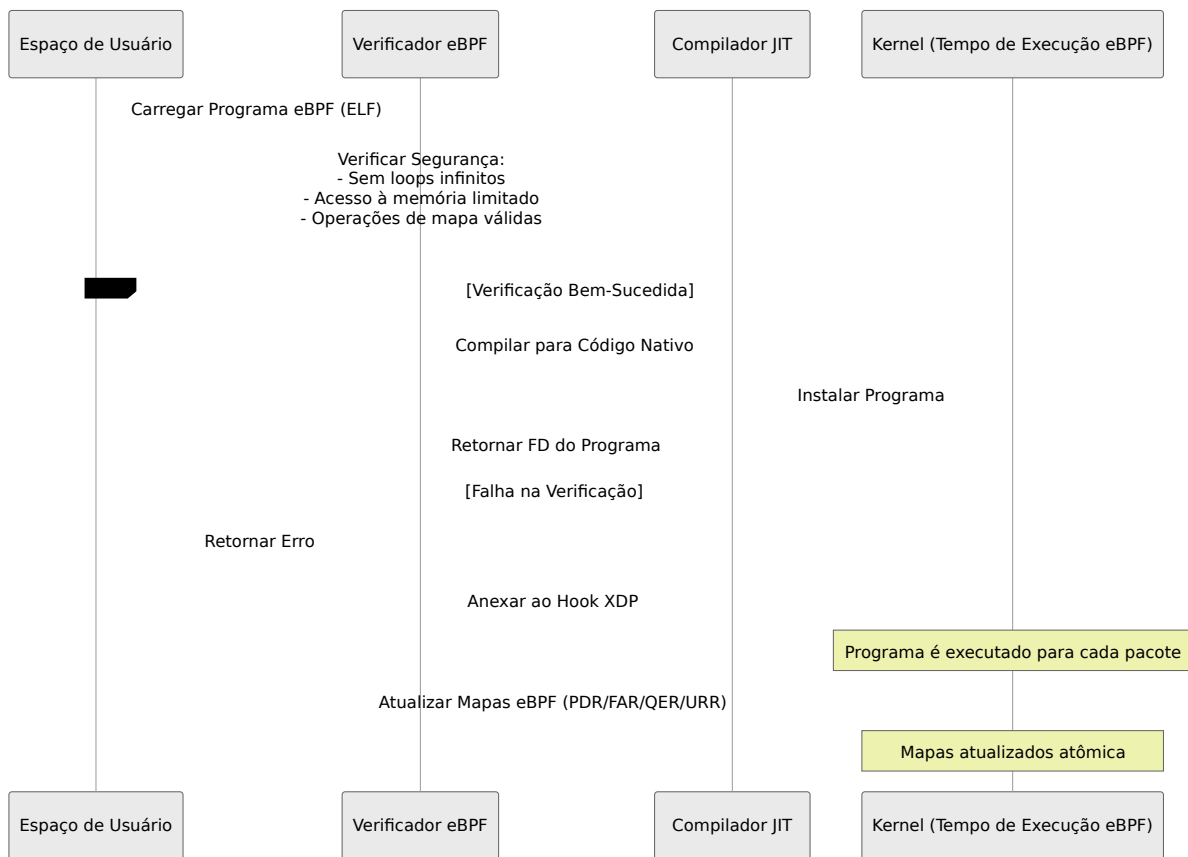
### **O que é eBPF?**

eBPF (filtro de pacotes Berkeley estendido) 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 podem 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 de usuário.

**Tipos de Mapa Usados no OmniUPF:**

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

### Operações de Mapa:

- **Busca:** O(1) busca hash (sub-microsegundo)
- **Atualização:** Atualizações atômicas do espaço de usuário
- **Exclusão:** Remoção imediata de entradas
- **Iterar:** Operações em lote para despejos de mapa

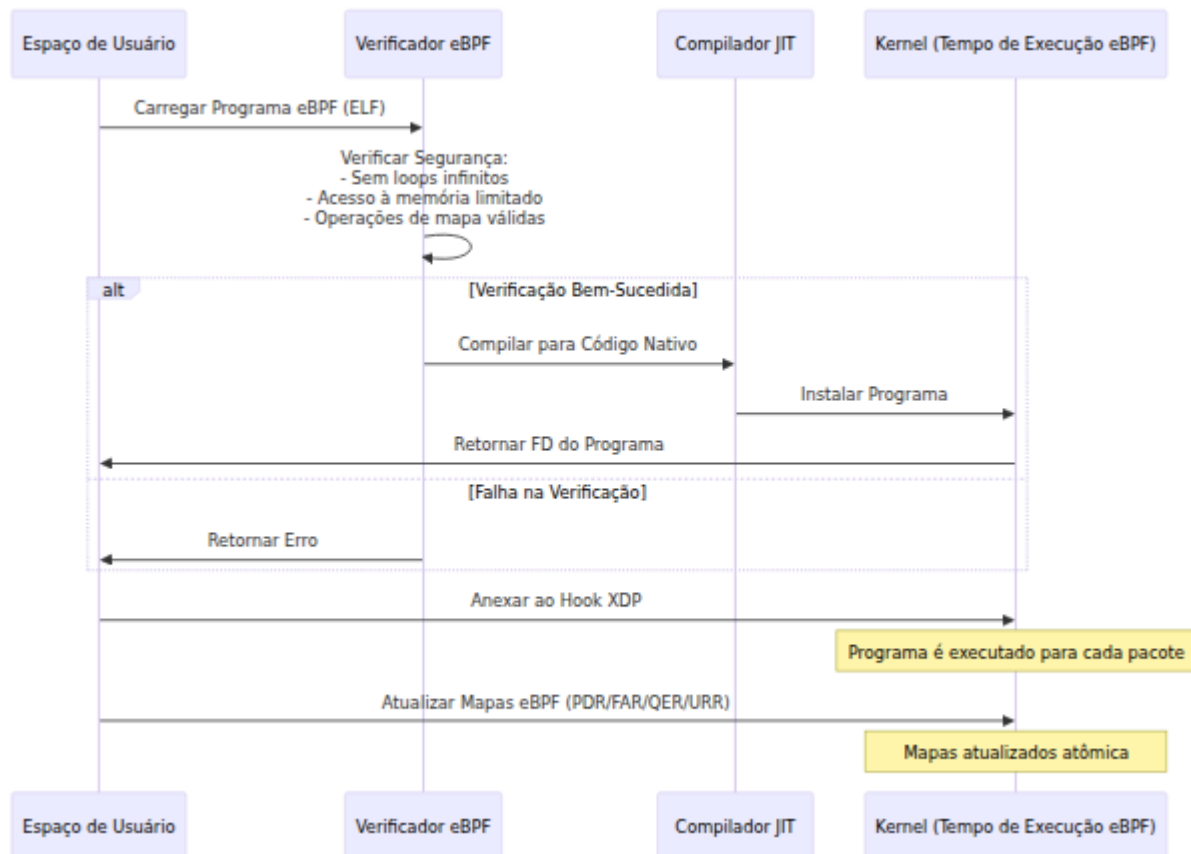
## Caminho de Dados XDP

### Visão Geral do XDP

XDP (Caminho de Dados eXpress) 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 do XDP

OmniUPF suporta três modos de anexação do 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:

```
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 (buffer de socket)
- Alcança 10-40 Gbps por núcleo
- Requer driver com suporte a XDP (a maioria dos drivers modernos)

**Configuração:**

```
xdp_attach_mode: native
```

**Vantagens:**

- Desempenho muito alto (multimilhões 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:

```
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 do 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	Buffering (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 interface diferente	Caminho de encaminhamento principal (N3 ↔ N6)
XDP_ABORTED	Erro de processamento, descartar pacote e registrar	Erros do 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 design modular de pipeline
- Profundidade máxima de 33 chamadas de cauda

## Processamento de Pacotes de Uplink

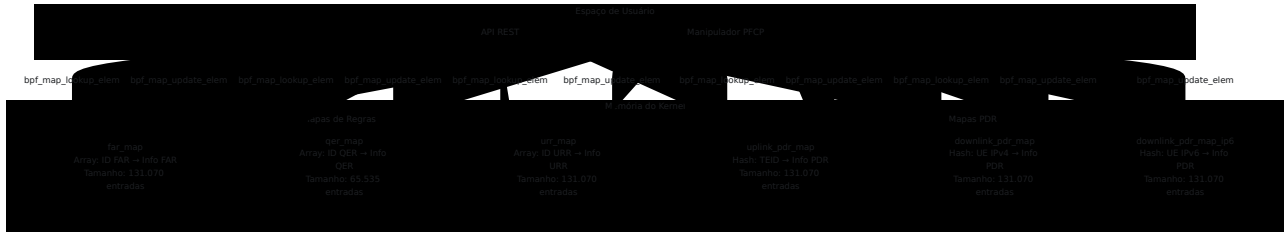


## Processamento de Pacotes de Downlink



# 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 Buffering

## Visão Geral do Buffering

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

## Arquitetura de Buffering

```
Parse error on line 4: ...ionar Cabeçalho UDP (porta 22152)<br/>3. -----  
-----^ Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '-)', 'STADIUMEND',  
'SUBROUTINEEND', 'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND',  
'TRAPEND', 'INVTRAPEND', 'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'PS'
```

Tentar novamente

## Detalhes da Encapsulação do Buffer

Quando o buffering está habilitado (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 em
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:

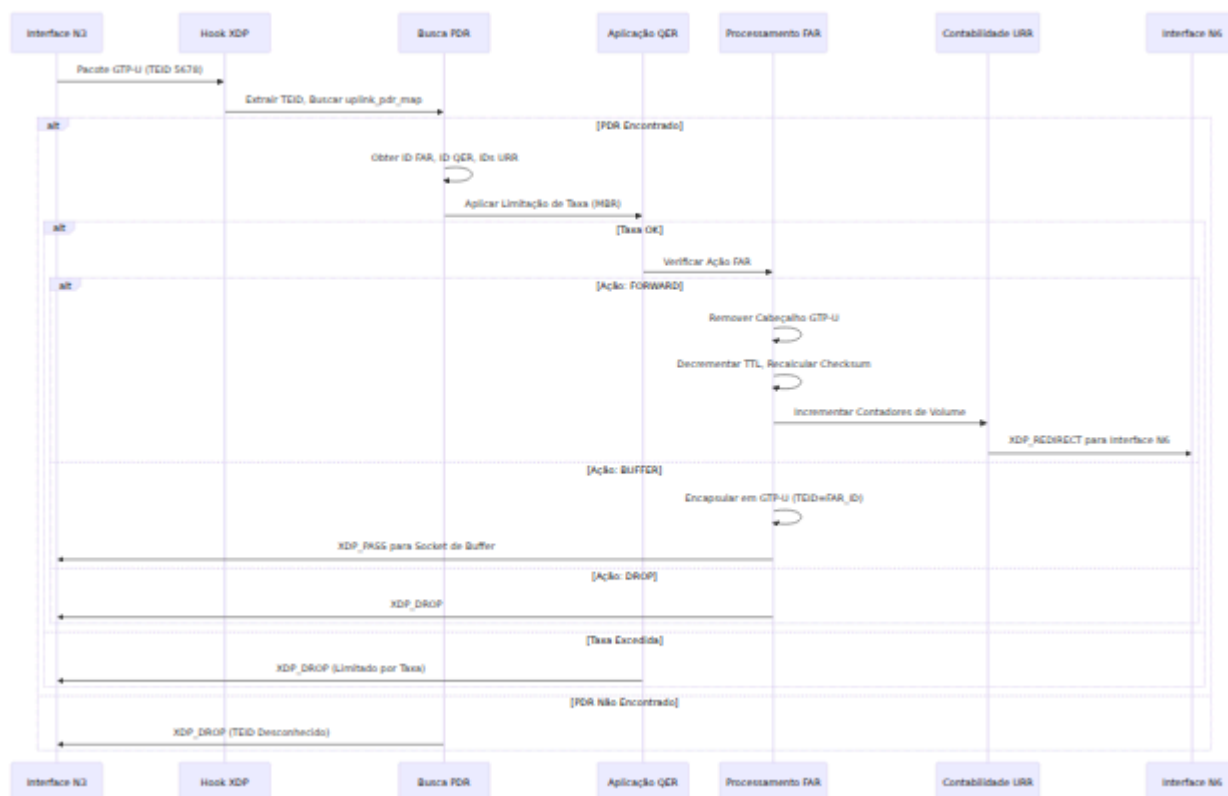
```
gtp->version = 1;
gtp->message_type = GTPU_G_PDU;
gtp->teid = htonl(far_id | (direction << 24)); // Codifica o
ID FAR e a direção
gtp->message_length = htons(orig_packet_len);
```

#### 6. Retorna XDP\_PASS:

- O kernel entrega o pacote ao socket UDP local na porta 22152
- O gerente de buffer em 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. Pacotes em buffer são reproduzidos:



## Parâmetros de Gerenciamento de Buffer

Parâmetro	Padrão	Descrição
<b>Máximo por FAR</b>	10.000 pacotes	Máximo de pacotes em buffer por FAR
<b>Máximo Total</b>	100.000 pacotes	Máximo total de pacotes em buffer
<b>TTL do Pacote</b>	30 segundos	Tempo antes que pacotes em buffer expirem
<b>Porta do Buffer</b>	22152	Porta UDP para entrega de buffer
<b>Intervalo de Limpeza do Buffer</b>	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.

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

Tentar novamente

## Implementação da Janela Deslizante

**Algoritmo** (do `qer.h`):

```

static __always_inline enum xdp_action limit_rate_sliding_window(
    const __u64 packet_size,
    volatile __u64 *window_start,
    const __u64 rate)
{
    static const __u64 NSEC_PER_SEC = 1000000000ULL;
    static const __u64 window_size = 5000000ULL; // 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 (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, redefina-a
    if (start + window_size < now) {
        *window_start = now - window_size + tx_time;
        return XDP_PASS;
    }

    // Atualizar 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

# Exemplo de Cálculo de QoS

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

## 1. Tempo de Transmissão:

$$\begin{aligned} \text{tx\_time} &= 1500 \text{ bytes} \times 8 \text{ bits/byte} \times (1.000.000.000 \text{ ns/sec} \div \\ &100.000.000 \text{ bps}) \\ \text{tx\_time} &= 1500 \times 8 \times 10 = 120.000 \text{ ns} = 120 \mu\text{s} \end{aligned}$$

## 2. Verificação de Taxa:

- Se o último pacote foi transmitido em  $t=0$ , o próximo pacote pode ser transmitido em  $t=120\mu\text{s}$
- Se o pacote chega em  $t=100\mu\text{s}$ , ele é descartado (muito cedo)
- Se o pacote chega em  $t=150\mu\text{s}$ , ele é encaminhado (janela avançada)

## 3. Taxa Máxima de Pacotes:

$$\begin{aligned} \text{Max PPS} &= (100 \text{ Mbps} \div 8) \div 1500 \text{ bytes} = 8.333 \text{ pacotes/segundo} \\ \text{Intervalo entre pacotes} &= 120 \mu\text{s} \end{aligned}$$

# Características de Desempenho

## Taxa de Transferência

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

## Quebra de Latência

**Latência Total de Processamento de Pacotes** (XDP Nativo):

Estágio	Latência	Cumulativa
NIC RX	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
NIC TX	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-core: quase linear até 8 núcleos

### Uso da CPU por Taxa de Pacotes:

Uso da CPU %  $\approx$  (Taxa de Pacotes / 10.000.000)  $\times$  100% por núcleo

**Exemplo:** Tráfego de 2 Mpps usa ~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 Requerida:

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

**Exemplo:** 10 Mpps × (1500 B + 3 acessos ao mapa × 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 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 (Receber Escalonamento Lateral) para distribuir filas RX
3. Fixar programas eBPF em núcleos específicos

### Ajustes de NIC:

1. Aumentar o tamanho do buffer de anel 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 de 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 Requeridos = (PPS Esperados ÷ 10.000.000) × 1.5
(50% de margem)
Memória Requerida = (Sessões Máximas × 212 B × 3) + 100 MB (mapas
eBPF + sobrecarga)
Rede Requerida = (Taxa de Pico × 2) + 10 Gbps (margem)
```

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

- CPU:  $(20 \text{ Gbps} \div 10 \text{ Gbps por núcleo}) \times 1.5 = 3\text{-}4$  núcleos
- Memória:  $(1\text{M} \times 212 \text{ B} \times 3) + 100 \text{ MB} \approx 750 \text{ MB}$

- Rede:  $(20 \text{ Gbps} \times 2) + 10 \text{ Gbps} = 50 \text{ Gbps}$  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 Configuração do OmniUPF

## Índice

1. [Visão Geral](#)
  2. [Modos de Operação](#)
  3. [Modos de Anexação XDP](#)
  4. [Parâmetros de Configuração](#)
  5. [Métodos de Configuração](#)
  6. [Compatibilidade com Hypervisor](#)
  7. [Compatibilidade com NIC](#)
  8. [Exemplos de Configuração](#)
  9. [Dimensionamento de Mapas e Planejamento de Capacidade](#)
- 

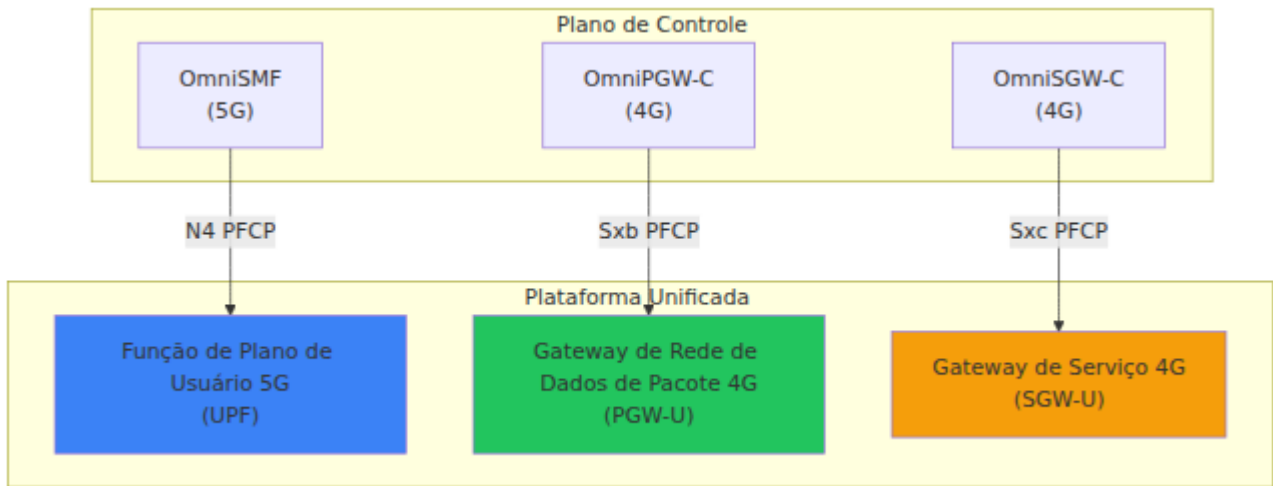
## Visão Geral

OmniUPF é uma função de plano de usuário versátil que pode operar em múltiplos modos para suportar redes centrais 4G (EPC) e 5G. A configuração é gerenciada através de arquivos de configuração YAML.

---

## 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

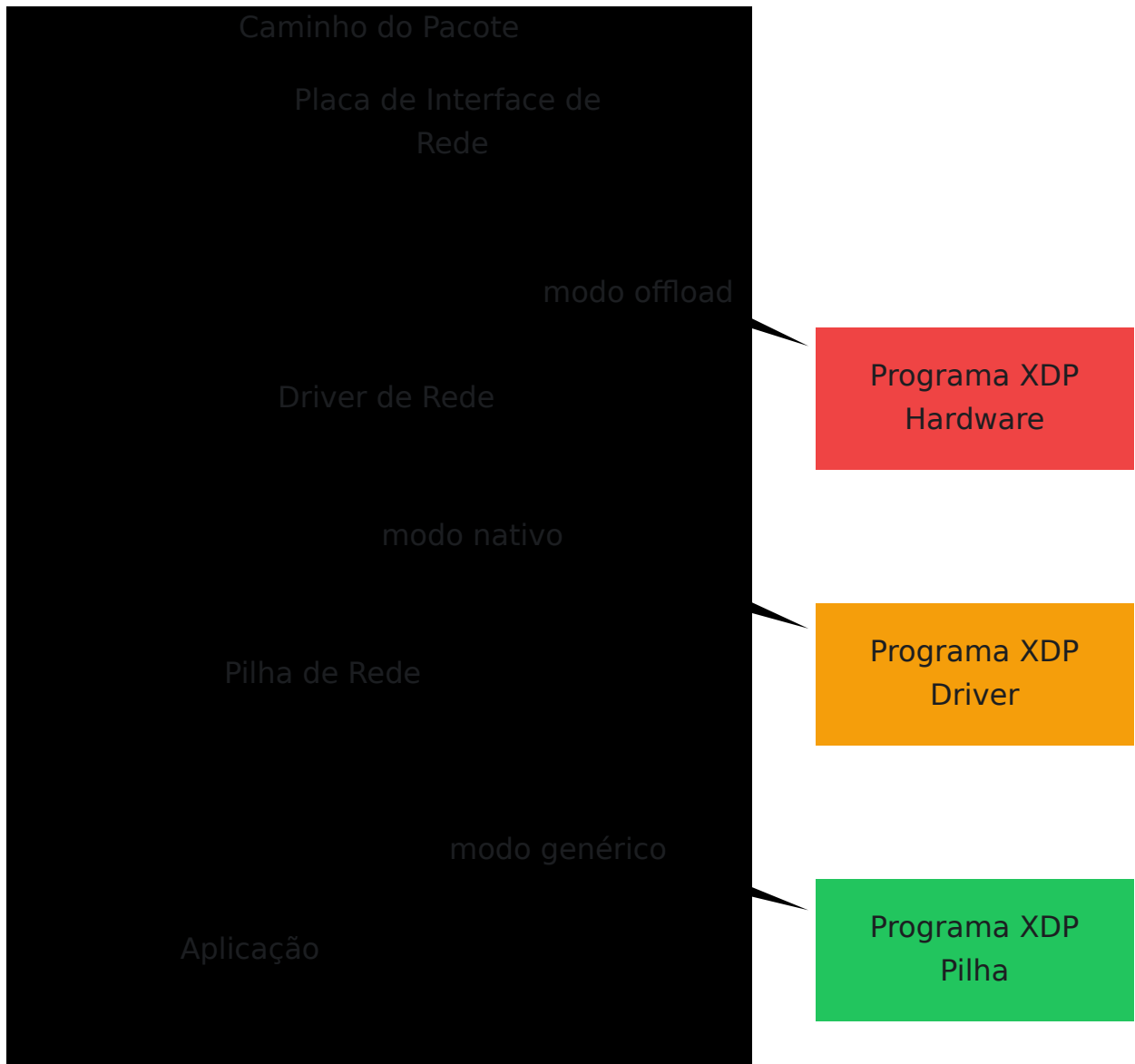
## Modos de Anexação XDP

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, consulte o [Guia de Modos XDP](#).**

## Comparação de Modos

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

```
xdp_attach_mode: generic
```

### Melhor para:

- Máquinas virtuais sem SR-IOV
  - Ambientes de teste 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 no 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:

```
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:

```
xdp_attach_mode: offload
```

### Melhor para:

- Implantações de produção de ultra-alto desempenho
- Computação de borda com requisitos de latência rigorosos
- Ambientes onde os recursos da 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
<code>interface_name</code>	Interfaces de rede para tráfego N3/N6/N9 (pontos de anexação XDP)	Lista	<code>[lo]</code>
<code>n3_address</code>	Endereço IPv4 para interface N3 (GTP-U da RAN)	IP	<code>127.0.0.1</code>
<code>n9_address</code>	Endereço IPv4 para interface N9 (UPF para UPF para ULCL)	IP	Mesmo que <code>n3_address</code>

## Exemplo:

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

---

## Configuração PFCP

Parâmetro	Descrição	Tipo	Padrão
<code>pfcp_address</code>	Endereço local para servidor PFCP (interface N4/Sxb/Sxc)	Host:Port	<code>:8805</code>
<code>pfcp_node_id</code>	ID do Nó Local para o protocolo PFCP	IP	<code>127.0.0.1</code>
<code>pfcp_remote_node</code>	Pares PFCP remotos (SMF/PGW-C/SGW-C) para conectar	Lista	<code>[]</code>
<code>association_setup_timeout</code>	Tempo limite entre Solicitações de Configuração de Associação (segundos)	Inteiro	<code>5</code>
<code>heartbeat_retries</code>	Número de tentativas de heartbeat antes de declarar o par como morto	Inteiro	<code>3</code>
<code>heartbeat_interval</code>	Intervalo de heartbeat PFCP (segundos)	Inteiro	<code>5</code>
<code>heartbeat_timeout</code>	Tempo limite de heartbeat PFCP	Inteiro	<code>5</code>

Parâmetro	Descrição	Tipo	Padrão
	(segundos)		

### Exemplo:

```

pfcf_address: :8805
pfcf_node_id: 10.100.50.241
pfcf_remote_node:
  - 10.100.50.10 # OmniSMF
  - 10.100.60.20 # OmniPGW-C
heartbeat_interval: 10
heartbeat_retries: 5

```

## API e Monitoramento

Parâmetro	Descrição	Tipo	Padrão
<code>api_address</code>	Endereço local para servidor REST API	Host:Port	<code>:8080</code>
<code>metrics_address</code>	Endereço local para endpoint de métricas Prometheus (veja <a href="#">Referência de Métricas</a> )	Host:Port	<code>:9090</code>
<code>logging_level</code>	Nível de log ( <code>trace</code> , <code>debug</code> , <code>info</code> , <code>warn</code> , <code>error</code> )	String	<code>info</code>

### Exemplo:

```

api_address: :8080
metrics_address: :9090
logging_level: debug

```

# Gerenciamento de Caminho GTP

Parâmetro	Descrição	Tipo	Padrão
<code>gtp_peer</code>	Lista de pares GTP para keepalives de Echo Request	Lista	<code>[]</code>
<code>gtp_echo_interval</code>	Intervalo entre GTP Echo Requests (segundos)	Inteiro	<code>10</code>

## Exemplo:

```
gtp_peer:  
- 10.100.50.50:2152 # gNB  
- 10.100.50.60:2152 # Outro UPF para N9  
gtp_echo_interval: 15
```

---

## Capacidade do Mapa eBPF

Parâmetro	Descrição	Tipo	Padrão	Auto-calculado
<code>max_sessions</code>	Número máximo de sessões simultâneas	Inteiro	<code>65535</code>	Usado para calcular tamanhos de mapa
<code>pdr_map_size</code>	Tamanho do mapa eBPF PDR	Inteiro	<code>0</code>	$\text{max\_sessions} \times 2$
<code>far_map_size</code>	Tamanho do mapa eBPF FAR	Inteiro	<code>0</code>	$\text{max\_sessions} \times 2$
<code>qer_map_size</code>	Tamanho do mapa eBPF QER	Inteiro	<code>0</code>	<code>max_sessions</code>
<code>urr_map_size</code>	Tamanho do mapa eBPF URR	Inteiro	<code>0</code>	$\text{max\_sessions} \times 2$

**Nota:** Definir tamanhos de mapa como `0` (padrão) habilita o auto-cálculo com base em `max_sessions`. Substitua por valores específicos se dimensionamento personalizado for necessário.

### Exemplo:

```
max_sessions: 100000
# Os mapas serão dimensionados automaticamente:
# PDR: 200.000 entradas
# FAR: 200.000 entradas
# QER: 100.000 entradas
# URR: 200.000 entradas
```

### Exemplo de dimensionamento personalizado:

```
max_sessions: 50000
pdr_map_size: 131070 # Tamanho personalizado
far_map_size: 131070
qer_map_size: 65535
urr_map_size: 131070
```

## Configuração de Buffer

Parâmetro	Descrição	Tipo	Padrão
<code>buffer_port</code>	Porta UDP para pacotes armazenados do eBPF	Inteiro	<code>22152</code>
<code>buffer_max_packets</code>	Número máximo de pacotes a serem armazenados por FAR	Inteiro	<code>10000</code>
<code>buffer_max_total</code>	Número máximo total de pacotes armazenados (0=ilimitado)	Inteiro	<code>100000</code>
<code>buffer_packet_ttl</code>	TTL para pacotes armazenados em segundos (0=sem expiração)	Inteiro	<code>30</code>
<code>buffer_cleanup_interval</code>	Intervalo de limpeza do buffer em segundos (0=sem limpeza)	Inteiro	<code>60</code>

**Exemplo:**

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

## Flags de Recursos

Parâmetro	Descrição	Tipo	Padrão
<code>feature_ueip</code>	Habilitar alocação de IP UE pelo OmniUPF	Booleano	<code>false</code>
<code>ueip_pool</code>	Pool de IP para alocação de IP UE (requer <code>feature_ueip</code> )	CIDR	<code>10.60.0.0/24</code>
<code>feature_ftup</code>	Habilitar alocação de F-TEID pelo OmniUPF	Booleano	<code>false</code>
<code>teid_pool</code>	Tamanho do pool de TEID para alocação de F-TEID (requer <code>feature_ftup</code> )	Inteiro	<code>65535</code>

### Exemplo (alocação de IP UE):

```
feature_ueip: true
ueip_pool: 10.45.0.0/16 # Alocar IPs UE deste pool
```

### Exemplo (alocação de F-TEID):

```
feature_ftup: true
teid_pool: 1000000 # Permitir até 1M alocações de TEID
```

# Configuração do Gerenciador de Rotas

Para sincronização de rotas UE com o daemon FRR (Free Range Routing). Consulte o [Guia de Gerenciamento de Rotas](#) para detalhes.

Parâmetro	Descrição	Tipo	Padrão
<code>route_manager_enabled</code>	Habilitar sincronização automática de rotas UE	Booleano	<code>false</code>
<code>route_manager_type</code>	Tipo de daemon de roteamento ( <code>frr</code> suportado)	String	<code>frr</code>
<code>route_manager_vtysh_path</code>	Caminho para o comando vtysh	String	<code>/usr/bin/vtysh</code>
<code>route_manager_nexthop</code>	IP do próximo salto para rotas UE	Endereço IP	<code>``</code> (vazio)

## Exemplo:

```
route_manager_enabled: true
route_manager_type: frr
route_manager_vtysh_path: /usr/bin/vtysh
route_manager_nexthop: 10.0.1.1 # Próximo salto para rotas UE
```

## Quando Habilitar:

- Implantações Multi-UPF que requerem anúncio de rotas

- Integração com protocolos de roteamento OSPF ou BGP
  - Requer daemon FRRouting instalado e configurado
- 

# Métodos de Configuração

## Arquivo de Configuração YAML (Recomendado)

Arquivo: `config.yml`

```
# Configuração de Rede
interface_name: [eth0]
n3_address: 10.100.50.233
n9_address: 10.100.50.233
xdp_attach_mode: native

# Configuração PFCP
pfcip_address: :8805
pfcip_node_id: 10.100.50.241
pfcip_remote_node:
  - 10.100.50.10

# API e Monitoramento
api_address: :8080
metrics_address: :9090
logging_level: info

# Capacidade
max_sessions: 100000

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

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

# Bufferização
buffer_max_packets: 15000
buffer_packet_ttl: 45
```

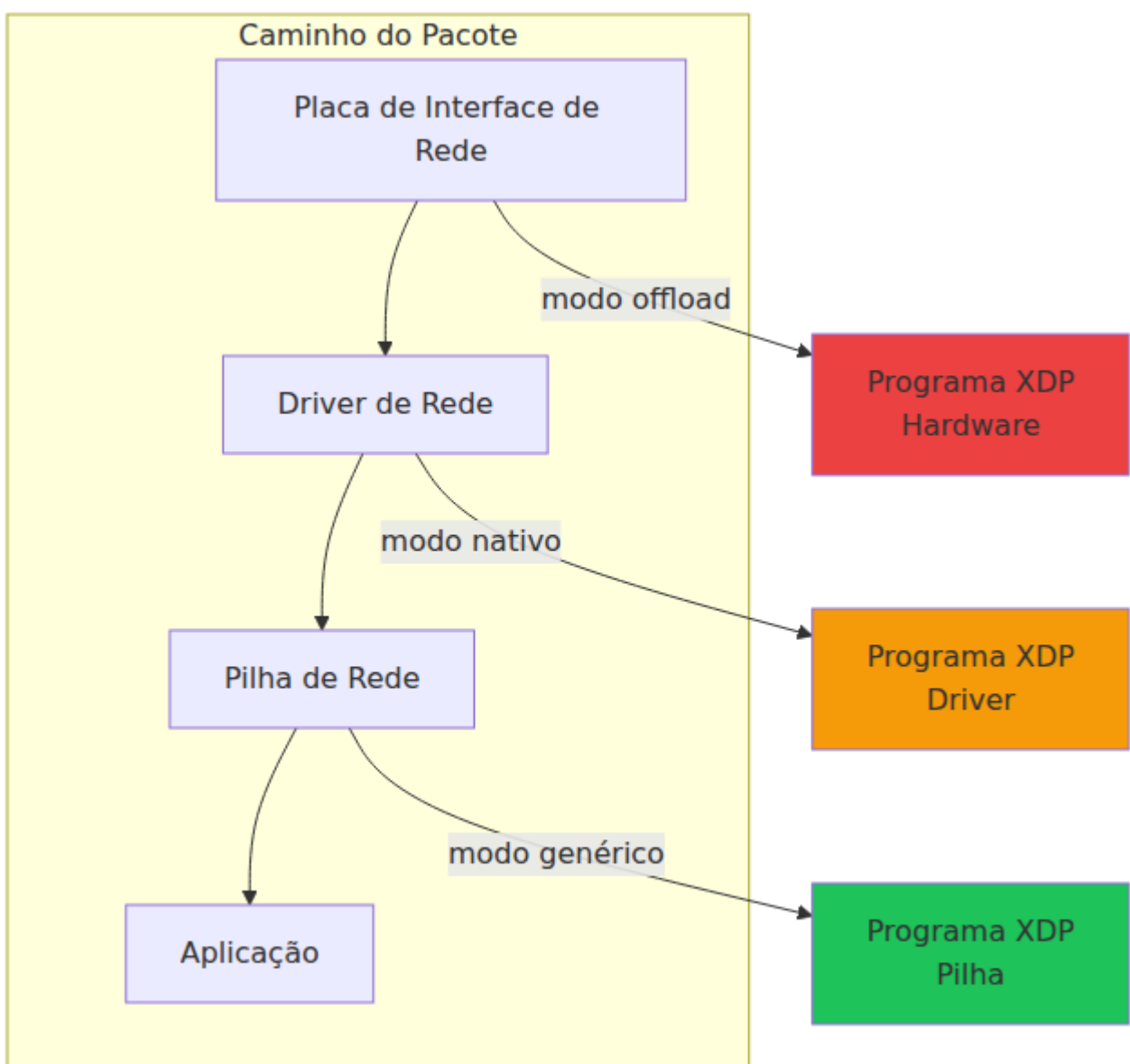
---

# 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 XDP nativo no Proxmox e em 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: `generic`
- Desempenho: ~1-2 Mpps

#### Configurações da VM Proxmox:

```
Dispositivo de Rede: net0  
Modelo: VirtIO (paravirtualizado)  
Bridge: vubr0
```

#### Configuração do OmniUPF:

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

---

### 2. Passthrough SR-IOV (XDP Nativo)

**Caso de uso:** Produção de alto desempenho

#### Configuração:

- Dispositivo de Rede: Função Virtual SR-IOV
- Modo XDP: `native`
- 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:

```
interface_name: [ens1f0] # Nome da VF SR-IOV
xdp_attach_mode: native
```

---

### 3. Passthrough PCI (XDP Nativo)

**Caso de uso:** NIC dedicada para uma única VM

#### Configuração:

- Toda a NIC física passada para a VM
- Modo XDP: `native` 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:

```
interface_name: [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 \  
  ...
```

#### Passthrough SR-IOV:

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

---

## VMware ESXi

### vSwitch Padrão (XDP Genérico):

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

### SR-IOV (XDP Nativo):

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

## Microsoft Hyper-V

### Switch Virtual (XDP Genérico):

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

### SR-IOV (XDP Nativo):

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

# VirtualBox

## Modo NAT/Bridge (apenas XDP Genérico):

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

# Compatibilidade com 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

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

### Sobrecarga de Encapsulamento 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:**

$$65 \text{ bytes} \times 1.000.000 \text{ pps} \times 8 \text{ bits/byte} = 520 \text{ Mbps}$$

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

- **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:**

$$1536 \text{ bytes} \times 1.000.000 \text{ pps} \times 8 \text{ bits/byte} = 12.288 \text{ Mbps} \approx 12.3 \text{ Gbps}$$

#### **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:**

$$29 \text{ bytes} \times 1.000.000 \text{ pps} \times 8 \text{ bits/byte} = 232 \text{ Mbps}$$

#### **Pacote N6 Máximo (MTU de 1500):**

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

#### **Throughput a 1 Mpps com pacotes N6 máximos:**

$1500 \text{ bytes} \times 1.000.000 \text{ pps} \times 8 \text{ bits/byte} = 12.000 \text{ Mbps} = 12 \text{ Gbps}$

## 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	<b>0.8-1.5 Gbps</b>	Voz AMR-WB, G.711
Web leve (N3)	400-600 bytes	436-636 bytes	<b>3.5-5.1 Gbps</b>	HTTP/HTTPS, mensagens
<b>Móvel moderno (N3)</b>	<b>1200 bytes</b>	<b>1236 bytes</b>	<b>9.9 Gbps</b>	<b>Mistura típica de tráfego em 2024</b>
Streaming de vídeo (N3)	1400-1450 bytes	1436-1486 bytes	<b>11.5-11.9 Gbps</b>	Fragmentos de vídeo HD/4K
MTU Máximo (N3)	1500 bytes	1536 bytes	<b>12.3 Gbps</b>	Grandes downloads de TCP

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

<b>Padrão de Tráfego</b>	<b>Tamanho do Pacote</b>	<b>Throughput a 10 Mpps</b>	<b>Caso de Uso Típico</b>
Pacotes VoIP	65-150 bytes	<b>0.5-1.2 Gbps</b>	Fluxos RTP de voz
Web leve	400-600 bytes	<b>3.2-4.8 Gbps</b>	Solicitações HTTP
<b>Móvel moderno</b>	<b>1200 bytes</b>	<b>9.6 Gbps</b>	<b>Tráfego típico em 2024</b>
Streaming de vídeo	1400-1450 bytes	<b>11.2-11.6 Gbps</b>	Downloads de vídeo
MTU Máximo	1500 bytes	<b>12.0 Gbps</b>	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.

**Por que isso importa para redes móveis:**

O tráfego móvel é **altamente variável** em tamanho de pacote e a sobrecarga GTP-U (36 bytes) impacta significativamente o desempenho de pacotes pequenos:

**Tamanho do pacote interno (dados reais do usuário):**

- **VoIP (codec AMR-WB):** 65-80 bytes → Com GTP-U: 101-116 bytes
- **Dados de sensores IoT:** 50-200 bytes → Com GTP-U: 86-236 bytes
- **Navegação na web (HTTP/3):** 400-800 bytes → Com GTP-U: 436-836 bytes
- **Streaming de vídeo:** 1200-1450 bytes → Com GTP-U: 1236-1486 bytes
- **Grandes downloads:** 1500 bytes → Com GTP-U: 1536 bytes

**Impacto da sobrecarga GTP-U:**

- Pacotes pequenos (< 200 bytes): **~35-70% de sobrecarga** - Mpps é o fator limitante
- Pacotes médios (200-800 bytes): **~5-20% de sobrecarga** - Limitação mista
- Pacotes grandes (> 1200 bytes): **~3% de sobrecarga** - A velocidade do link é o fator limitante

### Planejamento de Desempenho:

Uma NIC classificada em **10 Mpps** alcançará na interface N3:

- **Tráfego pesado de VoIP** (pacotes internos de 100 bytes): ~1.0 Gbps (sobrecarga GTP-U domina)
- **Mistura móvel moderna** (pacotes internos de 1200 bytes): ~9.9 Gbps
- **Tráfego pesado de vídeo** (pacotes internos de 1400 bytes): ~11.5 Gbps
- **Throughput máximo** (pacotes internos de 1500 bytes): ~12.3 Gbps

**Na interface N6** (sem sobrecarga GTP-U):

- **Mistura móvel moderna** (pacotes de 1200 bytes): ~9.6 Gbps a 10 Mpps
- **Throughput máximo** (pacotes de 1500 bytes): ~12.0 Gbps a 10 Mpps

### Regra Geral para UPF Móvel:

- **Tráfego de pacotes pequenos** (VoIP, IoT, sinalização): Mpps é limitante - planeje para 1-2 Gbps por 10 Mpps
- **Tráfego móvel moderno** (média de 1200 bytes): Planeje para ~9-10 Gbps por 10 Mpps de capacidade
- **Tráfego pesado de vídeo** (streaming, downloads): Planeje para ~10-12 Gbps por 10 Mpps de capacidade
- **Sempre considere tanto N3 quanto N6** - N3 tem sobrecarga GTP-U, N6 não

### 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)	Cenário de Implantação Realista
<b>1 Mpps</b>	~1.0 Gbps	~1.0 Gbps	Pequeno site celular, gateway IoT
<b>5 Mpps</b>	~4.9 Gbps	~4.8 Gbps	Site celular médio, empresa
<b>10 Mpps</b>	~9.9 Gbps	~9.6 Gbps	Grande site celular, pequena cidade
<b>20 Mpps</b>	~19.7 Gbps	~19.2 Gbps	Área metropolitana, cidade média
<b>40 Mpps</b>	~39.4 Gbps	~38.4 Gbps	Grande metrópole, hub regional

**Nota:** Essas estimativas assumem um tamanho médio de carga útil de 1200 bytes, que é representativo do tráfego móvel moderno dominado por streaming de vídeo, redes sociais e aplicações em nuvem. O throughput real variará com a mistura de tráfego.

## Drivers de Rede Compatíveis com XDP

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

### NICs Intel

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

### **NICs Mellanox/NVIDIA**

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

## NICs Broadcom

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

## Outros Fornecedores

Modelo	Driver	Suporte a XDP	Modo	Desempenho
<b>Netronome Agilio CX</b>	nfp	Sim	Offload	~30 Mpps
<b>Amazon ENA</b>	ena	Sim	Nativo	~5 Mpps
<b>Solarflare SFC9xxx</b>	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):

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

### **Estimativas de Throughput:**

- Baseado em tamanho médio de pacote de 1200 bytes com encapsulamento GTP-U (1236 bytes na N3)
- Throughput N6 ligeiramente inferior (~9.6 Gbps por 10 Mpps) devido à ausência de sobrecarga GTP-U

- O desempenho real varia com a mistura de tráfego - redes pesadas em VoIP terão throughput menor
- 

## 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
interface_name: [eth0]
xdp_attach_mode: generic
api_address: :8080
pfcg_address: :8805
pfcg_node_id: 127.0.0.1
n3_address: 127.0.0.1
metrics_address: :9090
logging_level: debug
max_sessions: 1000
```

---

## Exemplo 2: Produção Bare Metal (Modo Nativo)

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

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

---

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

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

```
# Configuração Proxmox SR-IOV
interface_name: [ens1f0] # VF SR-IOV
xdp_attach_mode: native
api_address: :8080
pfcf_address: 192.168.100.10:8805
pfcf_node_id: 192.168.100.10
n3_address: 192.168.100.10
metrics_address: :9090
logging_level: info
max_sessions: 100000
gtp_peer:
  - 192.168.100.50:2152
gtp_echo_interval: 15
pfcf_remote_node:
  - 192.168.100.20 # SMF
```

---

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

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

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

---

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

**Cenário:** OmniUPF servindo simultaneamente redes 5G e 4G

```
# Configuração de múltiplos modos
interface_name: [eth0, eth1]
xdp_attach_mode: native
api_address: :8080
pfcf_address: :8805
pfcf_node_id: 10.50.1.100
n3_address: 10.50.1.100
n9_address: 10.50.1.101
metrics_address: :9090
logging_level: info
max_sessions: 300000
gtp_peer:
  - 10.50.2.10:2152 # gNB 5G
  - 10.50.2.20:2152 # eNodeB 4G (via SGW-U)
gtp_echo_interval: 15
pfcf_remote_node:
  - 10.50.3.10 # OmniSMF (5G)
  - 10.50.3.20 # OmniPGW-C (4G)
heartbeat_interval: 10
feature_ueip: true
ueip_pool: 10.60.0.0/16
```

---

## Exemplo 6: Modo Offload SmartNIC

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

```
# Configuração de offload SmartNIC
interface_name: [enp1s0np0] # Interface SmartNIC
xdp_attach_mode: offload
api_address: :8080
pfc_p_address: 10.10.1.50:8805
pfc_p_node_id: 10.10.1.50
n3_address: 10.10.1.50
metrics_address: :9090
logging_level: warn # Reduzir sobrecarga
max_sessions: 1000000
pdr_map_size: 2000000
far_map_size: 2000000
qer_map_size: 1000000
gtp_peer:
  - 10.10.2.10:2152
  - 10.10.2.20:2152
  - 10.10.2.30:2152
gtp_echo_interval: 30
pfc_p_remote_node:
  - 10.10.3.10
heartbeat_interval: 15
buffer_max_packets: 1000000
buffer_max_total: 1000000
```

---

# Dimensionamento de Mapas e Planejamento de Capacidade

## Auto-Dimensionamento (Recomendado)

Defina `max_sessions` e deixe o OmniUPF calcular os tamanhos dos mapas automaticamente:

```
max_sessions: 100000
# Tamanhos auto-calculados:
# PDR: 200.000 entradas (2 × max_sessions)
# FAR: 200.000 entradas (2 × max_sessions)
# QER: 100.000 entradas (1 × max_sessions)
# URR: 200.000 entradas (2 × max_sessions)
```

**Uso de memória:** ~91 MB para 100K sessões

---

## Dimensionamento Manual

Substitua o auto-cálculo por requisitos personalizados:

```
max_sessions: 100000
pdr_map_size: 300000 # Suportar mais PDRs por sessão
far_map_size: 200000
qer_map_size: 150000 # Mais QERs do que o padrão
urr_map_size: 200000
```

---

## Estimativa de Capacidade

**Calcule o número 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 =  $\min(100.000, 100.000, 100.000) = \mathbf{100.000}$

---

## Requisitos de Memória

### Uso de memória por sessão:

- PDR:  $2 \times 212 \text{ B} = 424 \text{ B}$
- FAR:  $2 \times 20 \text{ B} = 40 \text{ B}$
- QER:  $1 \times 36 \text{ B} = 36 \text{ B}$
- URR:  $2 \times 20 \text{ B} = 40 \text{ B}$
- **Total:**  $\sim 540 \text{ B}$  por sessão

**Para 100K sessões:**  $\sim 52 \text{ MB}$  de memória do kernel

**Recomendação:** Certifique-se de que o limite de memória bloqueada permita  $2\times$  o uso estimado:

```
# Verifique o limite atual
ulimit -l

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

---

## Documentação Relacionada

- **Guia de Arquitetura** - Detalhes técnicos e otimização de desempenho do eBPF/XDP
- **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 Prometheus
- **Guia de UI Web** - Operações do painel de controle

- **Guia de Operações** - Visão geral da arquitetura e implantação do UPF

# Referência de Métricas

Este documento descreve todas as métricas do Prometheus expostas pelo OmniUPF no endpoint `/metrics`.

## Categorias de Métricas

1. **Métricas de mensagens PFCP** - Contadores de mensagens do protocolo de controle e latência por par
2. **Métricas de Ação XDP** - Veredictos de pacotes do plano de dados (descartar, passar, redirecionar, etc.)
3. **Métricas de Pacotes** - Contadores de pacotes recebidos por tipo de protocolo
4. **Métricas de Sessão e Associação PFCP** - Contagens de sessões e associações por par
5. **Métricas URR** - Contadores de volume de tráfego agregados por par PFCP
6. **Métricas de Bufferização de Pacotes** - Estado do buffer de pacotes, capacidade e taxa de transferência
7. **Métricas de Relatório de Dados de Downlink (Notificação)** - Notificações de Solicitação de Relatório de Sessão PFCP e rastreamento de índice FAR
8. **Métricas de Capacidade do Mapa eBPF** - Utilização e capacidade do mapa eBPF

## Referência de Métricas

### Métricas de mensagens PFCP

Métricas para rastrear mensagens do protocolo PFCP entre o UPF e os **nós** do plano de controle.

Nome da Métrica	Tipo	Rótulos	Descrição
upf_pfcpx_rx	Counter	message_name, peer_address	Número total de mensagens PFCP recebidas por tipo de mensagem e par
upf_pfcpx_tx	Counter	message_name, peer_address	Número total de mensagens PFCP transmitidas por tipo de mensagem e par
upf_pfcpx_rx_errors	Counter	message_name, cause_code, peer_address	Número total de mensagens PFCP rejeitadas com causa de erro por tipo de mensagem e par
upf_pfcpx_rx_latency	Summary	message_type, peer_address	Duração do processamento da mensagem PFCP em microsegundos (quantis p50, p90, p99) por tipo de mensagem e par

**Nota:** Todos os contadores rastreiam mensagens por par PFCP para visibilidade granular no comportamento do nó do plano de controle.

## Métricas de Ação XDP

Contadores de pacotes por ação/veredicto do programa XDP. Essas métricas rastreiam a decisão do plano de dados para cada pacote.

Nome da Métrica	Tipo	Rótulos	Descrição
upf_xdp_aborted	Counter	none	Número total de pacotes abortados (XDP_ABORTED)
upf_xdp_drop	Counter	none	Número total de pacotes descartados (XDP_DROP)
upf_xdp_pass	Counter	none	Número total de pacotes passados para o kernel (XDP_PASS)
upf_xdp_tx	Counter	none	Número total de pacotes transmitidos (XDP_TX)
upf_xdp_redirect	Counter	none	Número total de pacotes redirecionados (XDP_REDIRECT)

## Métricas de Pacotes

Contadores para pacotes recebidos por tipo de protocolo. Todas as métricas usam o rótulo `packet_type`.

Nome da Métrica	Tipo	Rótulos	Descrição
upf_rx	Counter	<code>packet_type</code>	Número total de pacotes recebidos por tipo
upf_route	Counter	<code>packet_type</code>	Número total de pacotes roteados pelo resultado da busca

**Valores de `packet_type` de `upf_rx`:**

- `arp` - Pacotes ARP
- `icmp` - Pacotes ICMP
- `icmp6` - Pacotes ICMPv6
- `ip4` - Pacotes IPv4
- `ip6` - Pacotes IPv6
- `tcp` - Pacotes TCP
- `udp` - Pacotes UDP
- `other` - Outros tipos de pacotes
- `gtp-echo` - Solicitação/resposta de eco GTP
- `gtp-pdu` - PDU GTP-U (dados de usuário encapsulados)
- `gtp-other` - Outros tipos de mensagens GTP
- `gtp-unexp` - Pacotes GTP inesperados/malformados

### Valores de `packet_type` de `upf_route`:

- `ip4-cache` - Acertos de cache de rota IPv4
- `ip4-ok` - Sucesso na busca FIB IPv4
- `ip4-error-drop` - Falha na busca FIB IPv4, pacote descartado
- `ip4-error-pass` - Falha na busca FIB IPv4, pacote passado para o kernel
- `ip6-cache` - Acertos de cache de rota IPv6
- `ip6-ok` - Sucesso na busca FIB IPv6
- `ip6-error-drop` - Falha na busca FIB IPv6, pacote descartado
- `ip6-error-pass` - Falha na busca FIB IPv6, pacote passado para o kernel

## Métricas de Sessão e Associação PFCP

Métricas para rastrear sessões e associações PFCP entre o UPF e os nós do plano de controle.

Nome da Métrica	Tipo	Rótulos	Descrição
upf_pfcps_sessions	Gauge	none	Número total de sessões PFCP atualmente estabelecidas (todos os pares)
upf_pfcps_associations	Gauge	none	Número total de associações PFCP atualmente estabelecidas (todos os pares)
upf_pfcps_association_status	Gauge	node_id, address	Status da associação PFCP por par (1=ativo, 0=inativo)
upf_pfcps_sessions_per_node	Gauge	node_id, address	Número de sessões PFCP ativas por nó do plano de controle

## Métricas URR (Regra de Relatório de Uso)

Métricas de volume de tráfego agregadas por par PFCP. O volume de cada par representa a soma de todos os contadores URR em todas as sessões daquele nó do plano de controle.

Nome da Métrica	Tipo	Rótulos	Descrição
upf_urr_uplink_volume_bytes	Gauge	peer_address	Volume total de tráfego uplink em bytes para todas as sessões deste par
upf_urr_downlink_volume_bytes	Gauge	peer_address	Volume total de tráfego downlink em bytes para todas as sessões deste par
upf_urr_total_volume_bytes	Gauge	peer_address	Volume total de tráfego em bytes (uplink + downlink) para todas as sessões deste par

**Nota:** Os volumes são agregados por par PFCP para evitar problemas de alta cardinalidade. Estatísticas URR individuais estão disponíveis via API REST em `/api/v1/urr_map`.

## Métricas de Bufferização de Pacotes

Métricas para rastrear o estado e o desempenho do buffer de pacotes. O UPF pode armazenar pacotes de downlink quando um UE está em estado ocioso, retendo-os até que o UE seja chamado e transite para o estado conectado.

Nome da Métrica	Tipo	Rótulos	
upf_buffer_packets_total	Counter	none	M c a b a
upf_buffer_packets_dropped	Counter	reason	M c c b
upf_buffer_packets_flushed	Counter	none	M c e b
upf_buffer_packets_current	Gauge	none	M c b
upf_buffer_bytes_total	Counter	none	M c a b a
upf_buffer_bytes_current	Gauge	none	E r
upf_buffer_fars_active	Gauge	none	M c p a

Nome da Métrica	Tipo	Rótulos	
upf_buffer_listener_packets_received_total	Counter	none	T P r l b c e
upf_buffer_listener_packets_buffered_total	Counter	none	T P a c P
upf_buffer_listener_errors_total	Counter	type	E P c l b
upf_buffer_listener_error_indications_sent_total	Counter	remote_peer	T n l E e T c P
upf_buffer_flush_success_total	Counter	none	T c e c s

Nome da Métrica	Tipo	Rótulos	
upf_buffer_flush_errors_total	Counter	reason	T C E C f.
upf_buffer_flush_packets_sent_total	Counter	none	T P E C C E

#### Valores de reason de upf\_buffer\_packets\_dropped:

- `expired` - Pacotes descartados devido à expiração do TTL
- `global_limit` - Descartados devido ao limite total do buffer alcançado
- `far_limit` - Descartados devido ao limite de buffer por FAR alcançado
- `cleared` - Pacotes manualmente limpos do buffer

#### Valores de type de upf\_buffer\_listener\_errors\_total:

- `read_error` - Erro ao ler do socket do buffer
- `too_small` - Pacote muito pequeno para o cabeçalho GTP
- `invalid_gtp_type` - Tipo de mensagem GTP não-G-PDU
- `unknown_teid` - Nenhum PDR/FAR encontrado para TEID
- `not_buffering_far` - FAR não tem ação BUFF
- `truncated_ext` - Cabeçalhos de extensão GTP truncados
- `no_payload` - Pacote GTP sem carga útil
- `buffer_full` - Capacidade do buffer excedida

#### Valores de reason de upf\_buffer\_flush\_errors\_total:

- `far_lookup_failed` - Falha ao buscar informações do FAR no mapa eBPF

- `no_forw_action` - FAR não tem ação FORW definida
- `connection_failed` - Falha ao criar conexão UDP para esvaziamento

## **Métricas de Relatório de Dados de Downlink (Notificação)**

Métricas para notificações de Solicitação de Relatório de Sessão PFCP enviadas ao plano de controle quando pacotes são armazenados. Essas notificações acionam o plano de controle para chamar o UE.

Nome da Métrica	Tipo	Rótulos	De
upf_dldr_sent_total	Counter	none	Núm de noti de F de L Dov (DLI env SMF
upf_dldr_send_errors	Counter	none	Núm de e env noti de F de L Dov
upf_dldr_active_notifications	Gauge	none	Núm de F noti DLC (ain limp
upf_far_index_size	Gauge	none	Núm de F regi no F para noti DLC
upf_far_index_registrations_total	Counter	none	Núm de r

Nome da Métrica	Tipo	Rótulos	De
			de F Farl
upf_far_index_unregistrations_total	Counter	none	Nún de desi de F Farl
upf_buffer_notify_to_flush_duration_seconds	Histogram	pfcp_peer	Tem o er noti DLD esva dos arm

### upf\_buffer\_notify\_to\_flush\_duration\_seconds:

- Baldes de histograma: 0.01, 0.05, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0, 60.0 segundos
- Rótulo pfcp\_peer: Endereço SMF/PGW-C (por exemplo, 10.100.50.241)
- Mede a latência entre o UPF enviando a notificação ao SMF e o SMF respondendo com a modificação da sessão para esvaziar pacotes
- Útil para monitorar a capacidade de resposta do plano de controle durante transições de ocioso para conectado

## Métricas de Indicação de Erro GTP-U

Métricas para rastrear mensagens de Indicação de Erro GTP-U enviadas e recebidas. Indicações de Erro são enviadas quando um par recebe pacotes para TEIDs desconhecidos, indicando incompatibilidades de estado de túnel (frequentemente devido a reinicializações de pares).

Nome da Métrica	Tipo	Ró
upf_buffer_listener_error_indications_sent_total	Counter	node_ peer_
upf_buffer_listener_error_indications_received_total	Counter	node_ peer_
upf_buffer_listener_error_indication_sessions_deleted_total	Counter	node_ peer_

### Definições de Rótulos:

- `node_id`: ID do Nó PFCP da associação (por exemplo, "pgw-u-1", "smf-1"). Definido como "unknown" se não existir associação PFCP para aquele par.
- `peer_address`: Endereço IP do par remoto (por exemplo, "192.168.50.10")

### Quando as Indicações de Erro São Enviadas:

- O UPF recebe um pacote GTP-U para um TEID que não existe (por exemplo, após a reinicialização do UPF, a sessão já foi deletada)
- O remetente (eNodeB, gNodeB, UPF a montante) está encaminhando para um túnel obsoleto/deletado
- O UPF envia uma Indicação de Erro para informar ao remetente para parar de enviar

## Quando as Indicações de Erro São Recebidas:

- O UPF encaminha um pacote GTP-U para um par a jusante (PGW-U, SGW-U, UPF) para um TEID desconhecido
- O par remoto não reconhece o TEID de destino (por exemplo, o par foi reiniciado e perdeu o estado do túnel)
- O UPF automaticamente deleta sessões afetadas para parar de encaminhar para túneis mortos

## Casos de Uso:

- Detectar reinicializações de pares (alta taxa de Indicações de Erro indica perda de estado)
- Identificar incompatibilidades de configuração (problemas de alocação de TEID)
- Monitorar a saúde da sincronização de túneis entre elementos de rede
- Alertar sobre exclusões inesperadas de sessões

## Exemplos de Consultas PromQL:

```
# Taxa de Indicações de Erro recebidas por par (por segundo)
rate(upf_buffer_listener_error_indications_received_total[5m])

# Total de sessões deletadas devido a Indicações de Erro de um par es
upf_buffer_listener_error_indication_sessions_deleted_total{peer_addr

# Pares enviando TEIDs desconhecidos para este UPF
sum by (node_id, peer_address) (upf_buffer_listener_error_indications
```

## Métricas de Capacidade do Mapa eBPF

Métricas para rastrear a utilização do mapa eBPF. Essas métricas ajudam a monitorar o uso de recursos e detectar potenciais problemas de capacidade.

Nome da Métrica	Tipo	Rótulos	Descrição
upf_ebpf_map_capacity	Gauge	map_name	Capacidade máxima do mapa eBPF
upf_ebpf_map_used	Gauge	map_name	Número atual de entradas no mapa eBPF

### Valores comuns de map\_name:

- pdr\_map - Mapa de Regras de Detecção de Pacotes
- far\_map - Mapa de Regras de Ação de Encaminhamento
- qer\_map - Mapa de Regras de Aplicação de QoS
- session\_map - Mapa de busca de sessão
- teid\_map - Mapeamento de TEID para sessão
- ue\_ip\_map - Mapeamento de endereço IP de UE para sessão

# Usando Métricas do Prometheus

## Acessando Métricas

As métricas são expostas no endpoint `/metrics` no endereço especificado por `metrics_address` no arquivo de configuração (padrão `:9090`):

```
# Ver métricas brutas
curl http://localhost:9090/metrics

# Exemplo de saída
upf_pfcps_sessions 42
upf_pfcps_associations 2
upf_urr_total_volume_bytes{peer_address="10.100.50.241"}
1048576000
```

# Configuração do Prometheus

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

```
scrape_configs:  
  - job_name: 'omniupf'  
    static_configs:  
      - targets: ['localhost:9090']
```

## Painéis do Grafana

Importe métricas para o Grafana para visualização:

- Contagens e tendências de sessões
- Volume de tráfego por par PFCP
- Taxas de processamento de pacotes
- Utilização de buffer
- Monitoramento da capacidade do mapa eBPF

## Documentação Relacionada

- **Guia de Monitoramento** - Monitoramento de estatísticas, planejamento de capacidade e alertas
- **Guia de Configuração** - Configurar `metrics_address` e outras opções do UPF
- **Guia da Interface Web** - Visualizar métricas na página de Estatísticas
- **Guia de Arquitetura** - Caminho de dados eBPF e otimização de desempenho
- **Guia de Gestão de Regras** - Compreendendo métricas PDR, FAR, QER, URR
- **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 Limites](#)
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.

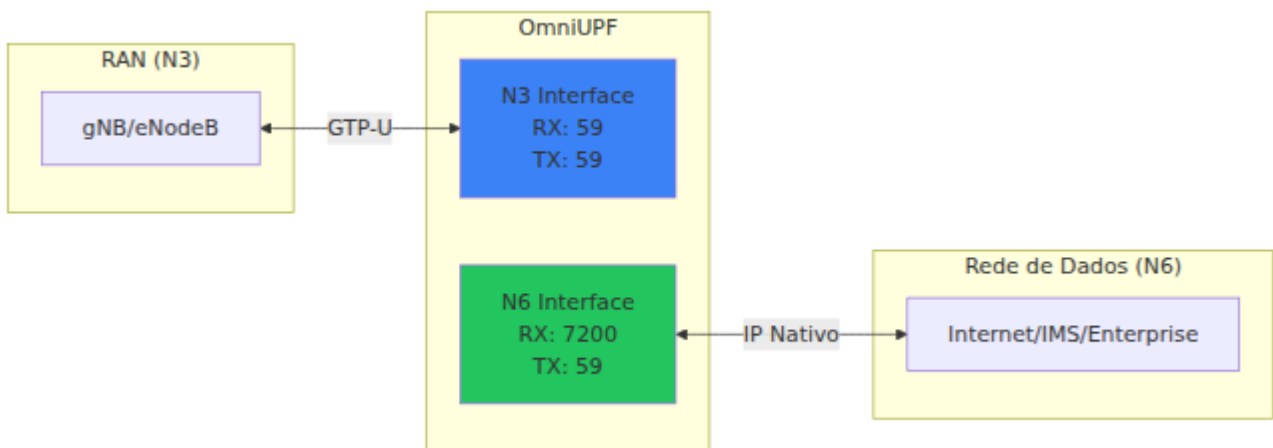
# Categorias de Monitoramento

<b>Categoria</b>	<b>Propósito</b>	<b>Frequência de Atualização</b>	<b>Métricas Chave</b>
<b>Estatísticas de Pacotes</b>	Acompanhar taxas de processamento de pacotes e erros	Em tempo real	Pacotes RX/TX, descartes, detalhamento de protocolo
<b>Estatísticas de Interface</b>	Monitorar a distribuição de tráfego N3/N6	Em tempo real	N3 RX/TX, N6 RX/TX
<b>Estatísticas de XDP</b>	Acompanhar o desempenho do caminho de dados do kernel	Em tempo real	XDP processados, passados, descartados, abortados
<b>Estatísticas de Roteamento</b>	Monitorar decisões de roteamento de pacotes	Em tempo real	Consultas FIB, acertos/miss de cache
<b>Capacidade do Mapa eBPF</b>	Prevenir exaustão de recursos	A cada 10s	Percentuais de uso do mapa, usado vs. capacidade
<b>Estatísticas de Buffer</b>	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

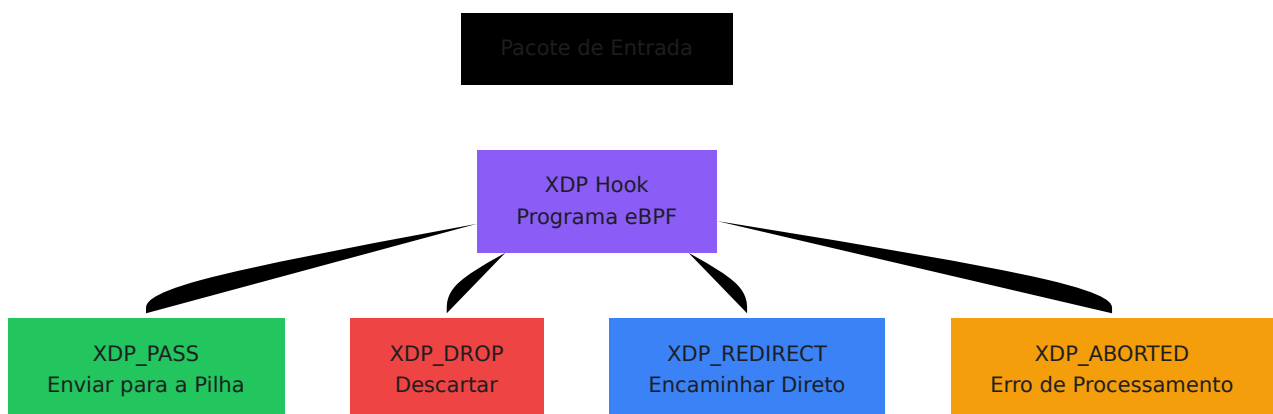
### Comportamento Esperado:

- **RX N3  $\approx$  TX N6:** Pacotes de uplink fluem da RAN para a Rede de Dados
- **RX N6  $\approx$  TX N3:** Pacotes de downlink fluem da Rede de Dados para a RAN
- Um desequilíbrio significativo pode indicar:
  - Tráfego assimétrico (downloads  $\gg$  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 descartados intencionalmente 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:** Descartes 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 OTHER:** Outras mensagens de controle GTP
- **RX GTP PDU:** Dados de usuário encapsulados em GTP-U (tráfego principal)
- **RX GTP UNEXP:** Tipos de pacotes GTP inesperados
- **RX ICMP:** Protocolo de Mensagens de Controle da Internet (ping, erros)
- **RX ICMP6:** Pacotes ICMPv6
- **RX IP4:** Pacotes IPv4
- **RX IP6:** Pacotes IPv6
- **RX OTHER:** Outros protocolos
- **RX TCP:** Pacotes do Protocolo de Controle de Transmissão
- **RX UDP:** Pacotes do Protocolo de Datagramas do Usuário

### Casos de Uso:

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

## Estatísticas de Roteamento

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

### Consulta FIB IPv4:

- **Cache:** Consultas de roteamento em cache (caminho rápido)
- **OK:** Consultas de roteamento bem-sucedidas

### Consulta FIB IPv6:

- **Cache:** Consultas de roteamento IPv6 em cache
- **OK:** Consultas de roteamento IPv6 bem-sucedidas

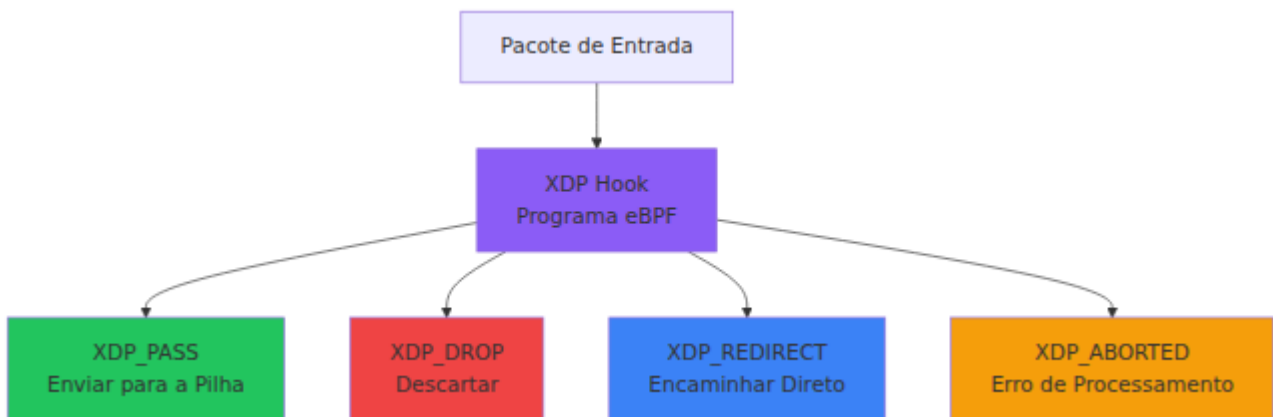
### Indicadores de Desempenho:

- **Alta Taxa de Acerto de Cache:** Indica bom desempenho do cache de roteamento
  - **Alta Contagem de OK:** Confirma que as tabelas de roteamento estão configuradas corretamente
  - **Baixas ou Zero Consultas:** 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 - A falha na estabelecimento de sessões ocorre se estiver cheia

### **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 - A falha na estabelecimento de sessões IPv6 ocorre se estiver 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 - O tráfego de uplink falha se estiver 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

## Limites de Capacidade

Limite	Ação Necessária
<b>0-50% (Verde)</b>	Operação normal - Nenhuma ação necessária
<b>50-70% (Amarelo)</b>	Cuidado - Monitorar tendências de crescimento, planejar aumento de capacidade
<b>70-90% (Âmbar)</b>	Aviso - Agendar aumento de capacidade dentro de 1 semana
<b>90-100% (Vermelho)</b>	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 novos tamanhos de mapa
3. Reiniciar o serviço OmniUPF
4. Verificar nova capacidade na visualização de Capacidade
5. Monitorar para estabelecimento bem-sucedido de sessões

**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:

- **Pequeno UPF:** 10.000 - 100.000 pps
- **Médio UPF:** 100.000 - 1.000.000 pps
- **Grande UPF:** 1.000.000 - 10.000.000 pps

### Indicadores de Gargalo:

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

## Cálculo de Throughput

### Cálculo:

Throughput (Mbps) =  $(\text{Delta de Contagem de Bytes} \times 8) / (\text{Delta de Tempo em segundos} \times 1.000.000)$

### Exemplo:

- Bytes RX iniciais: 500 MB
- Após 60 segundos: 800 MB
- Throughput =  $(300 \text{ MB} \times 8) / (60 \times 1.000.000) = 40$  Mbps

### 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 o throughput de pico para margem
-

# Taxa de Descarte

## Cálculo:

$$\text{Taxa de Descarte (\%)} = \left( \frac{\text{Pacotes Descartados}}{\text{Total de Pacotes RX}} \right) \times 100$$

## Limites 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 do QER (MBR excedido)
  - Falhas de consulta do mapa eBPF
  - TEIDs ou IPs de UE inválidos
  - Erros de roteamento
- 

# Alertas e Limites

## 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 mais antigos que 30s)

#### **Alertas Informativos** (Monitorar tendências):

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

## Configuração de Alertas

Os alertas podem ser configurados via:

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

## Planejamento de Capacidade

### Estimativa de Capacidade de Sessão

Calcular sessões máximas:

```
Max Sessions = 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

Max Sessions =  $\min(131.070 / 2, 131.070 / 2, 65.535) = \mathbf{65.535}$  sessões

## Capacidade de Memória

### Calcular a memória total do mapa eBPF:

```
Memória =  $\Sigma$  (Capacidade do Mapa  $\times$  (Tamanho da Chave + Tamanho do Valor))
```

### Exemplo de Configuração:

- Mapas PDR:  $3 \times 131.070 \times 212 \text{ B} = 83,3 \text{ MB}$
- Mapa FAR:  $131.070 \times 20 \text{ B} = 2,6 \text{ MB}$
- Mapa QER:  $65.535 \times 36 \text{ B} = 2,3 \text{ MB}$
- Mapa URR:  $131.070 \times 20 \text{ B} = 2,6 \text{ MB}$
- **Total:** ~91 MB de memória do kernel

### Considerações sobre Memória do Kernel:

- Garantir limite de memória bloqueada suficiente (`ulimit -l`)
- Reservar 2x o 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 de pico diária
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:  $(65.535 - 30.000) / 2.000 = \mathbf{17,8 \text{ semanas}}$

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

---

# 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 do QER:** Verificar valores de MBR do 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 o overflow for 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 de 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 obsoletas (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 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 do 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 Web UI** - 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

# N9 Loopback: Executando SGWU e PGWU na Mesma Instância

## Visão Geral

OmniUPF suporta a execução das funções **SGWU (Serving Gateway User Plane)** e **PGWU (PDN Gateway User Plane)** na **mesma instância** com **loopback N9 de zero latência**. Este modo de implantação é ideal para:

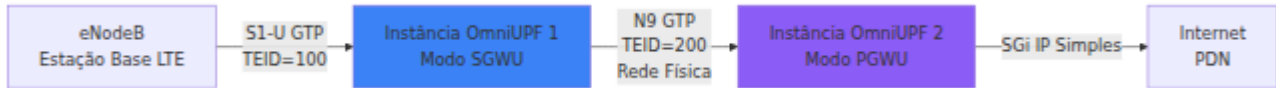
- **Implantações simplificadas de EPC 4G** - Uma única instância de UPF em vez de duas
- **Otimização de custos** - Redução da complexidade de infraestrutura e operacional
- **Computação de borda** - Minimizar a latência para cenários de breakout local
- **Ambientes de laboratório/teste** - Plano de usuário EPC completo em um único servidor

Quando configurado com o mesmo endereço IP para as interfaces N3 e N9, o OmniUPF **detecta automaticamente** o tráfego fluindo entre os papéis SGWU e PGWU e o processa **totalmente em eBPF** sem nunca enviar pacotes para a interface de rede.

---

# Como Funciona

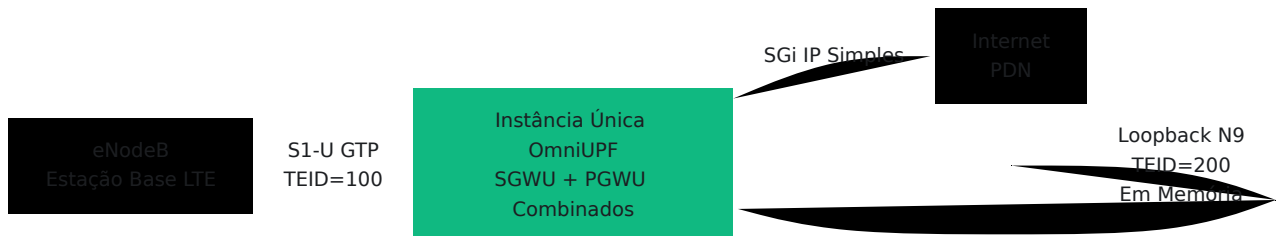
## Implantação Tradicional (Duas Instâncias)



### Fluxo de Pacotes:

1. eNodeB → SGWU: Pacote GTP (TEID=100) chega no S1-U
2. SGWU: Combina PDR de uplink, encapsula em novo túnel GTP (TEID=200)
3. **Pacote enviado pela rede física N9** para a instância PGWU
4. PGWU: Recebe GTP (TEID=200), decapsula, encaminha para a Internet
5. **Total: 2 passes XDP + 1 salto de rede**

## Implantação de Loopback N9 (Uma Única Instância)



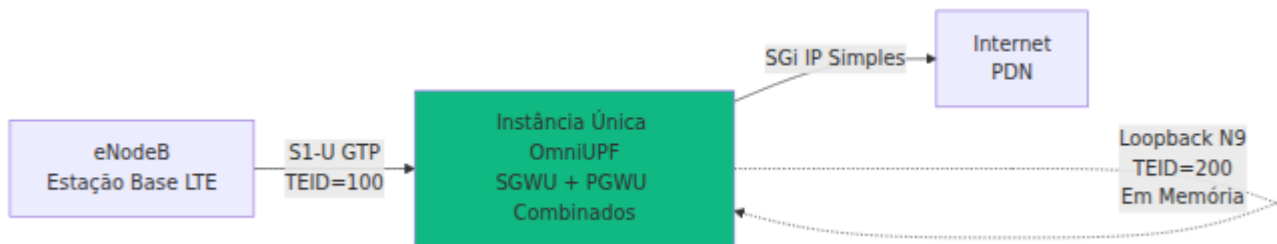
### Fluxo de Pacotes com Loopback N9:

1. eNodeB → papel SGWU: Pacote GTP (TEID=100) chega no S1-U
2. Papel SGWU: Combina PDR de uplink
3. **Deteção de loopback:** IP de destino = IP local (10.0.1.10)
4. **Processamento in-place:** Atualiza GTP TEID para 200 (sessão PGWU)
5. Papel PGWU: Decapsula, encaminha para a Internet
6. **Total: 1 passe XDP, zero saltos de rede**

**Benefício de desempenho:** Encaminhamento interno sub-microsegundo vs milissegundos para ida e volta na rede

# Detalhes do Processamento de Pacotes

## Fluxo de Uplink: eNodeB → SGWU → PGWU → Internet

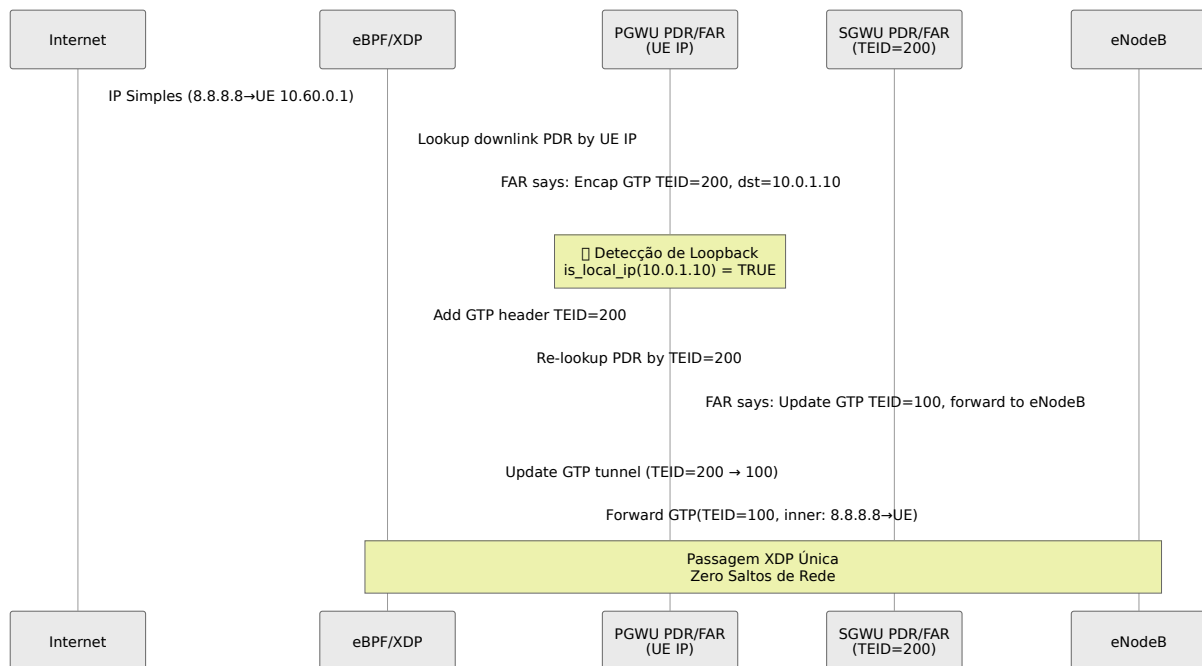


**Caminho do Código:** `cmd/ebpf/xdp/n3n6_entrypoint.c` linhas 349-403

### Principais Etapas:

1. **Receber:** Pacote GTP do eNodeB com TEID=100
  2. **Combinação de PDR:** Lookup PDR de uplink para sessão SGWU (TEID=100)
  3. **Ação FAR:** Encapsular em GTP com TEID=200, encaminhar para 10.0.1.10
  4. **Verificação de Loopback:** `is_local_ip(10.0.1.10)` retorna TRUE
  5. **Atualizar TEID:** Mudar `ctx->gtp->teid` de 100 para 200 (na memória do kernel)
  6. **Reprocessar:** Lookup PDR para TEID=200 (sessão PGWU)
  7. **Ação FAR:** Remover cabeçalho GTP, encaminhar para a Internet
  8. **Rota:** Enviar pacote IP simples para a interface N6
-

# Fluxo de Downlink: Internet → PGWU → SGWU → eNodeB



**Caminho do Código:** `cmd/ebpf/xdp/n3n6_entrypoint.c` linhas 137-194 (IPv4), 265-322 (IPv6)

## Principais Etapas:

1. **Receber:** Pacote IP simples da Internet destinado ao UE (10.60.0.1)
  2. **Combinação de PDR:** Lookup PDR de downlink pelo IP do UE (sessão PGWU)
  3. **Ação FAR:** Encapsular em GTP com TEID=200, encaminhar para 10.0.1.10
  4. **Verificação de Loopback:** `is_local_ip(10.0.1.10)` retorna TRUE
  5. **Adicionar GTP:** Encapsular pacote com TEID=200
  6. **Reprocessar:** Lookup PDR para TEID=200 (sessão SGWU)
  7. **Ação FAR:** Atualizar túnel GTP para eNodeB TEID=100
  8. **Rota:** Enviar pacote GTP para a interface S1-U (eNodeB)
-

# Configuração

## Requisitos

### Plano de Controle:

- **SGWU-C:** Deve conectar à interface PFCP do OmniUPF (por exemplo, `192.168.1.10:8805`)
- **PGWU-C:** Deve conectar à **mesma** interface PFCP do OmniUPF

### Rede:

- **Um único endereço IP** para as interfaces N3 e N9
  - **Endereços IP diferentes** para SGWU-C e PGWU-C (se executando no mesmo host, use portas diferentes)
- 

## Configuração do OmniUPF

**config.yml:**

```

# Interfaces de rede
interface_name: [eth0]           # Interface única para S1-U e
N9                               # Use nativo para melhor
xdp_attach_mode: native         # desempenho

# Interface PFCP
pfcip_address: ":8805"          # Ouvir em todas as
interfaces, porta 8805
pfcip_node_id: "192.168.1.10"  # ID do Nó PFCP do OmniUPF

# Interfaces do Plano do Usuário
n3_address: "10.0.1.10"        # IP da interface S1-U/N3
n9_address: "10.0.1.10"        # IP da interface N9 (IGUAL ao
N3)

# APIs
api_address: ":8080"           # API REST
metrics_address: ":9090"       # Métricas do Prometheus (veja
o documento de Referência de Métricas)

# Pools de Recursos
ueip_pool: "10.60.0.0/16"      # Pool de endereços IP do UE
teid_pool: 65535               # Pool de alocação de TEID

# Capacidade
max_sessions: 100000           # Máximo de sessões UE
concorrentes

```

## Configuração Chave:

- `n3_address` e `n9_address` **DEVEM ser idênticos** para habilitar o loopback
- `pfcip_address` Endereço PFCP único para ambos os planos de controle
- `max_sessions` Suficiente para carga combinada de SGWU + PGWU

## Configuração do Plano de Controle

### Configuração do SGWU-C

```
# Apontar para a interface PFCP do OmniUPF
upf_pfcip_address: "192.168.1.10:8805"

# Interface S1-U (mesma que o n3_address do OmniUPF)
sgwu_s1u_address: "10.0.1.10"

# Interface N9 para encaminhamento para PGWU (mesma que o OmniUPF)
sgwu_n9_address: "10.0.1.10"
```

## Configuração do PGWU-C

```
# Apontar para a MESMA interface PFCP do OmniUPF
upf_pfcip_address: "192.168.1.10:8805"

# Interface N9 (recebe do SGWU)
pgwu_n9_address: "10.0.1.10"

# Interface SGi para conectividade com a Internet
pgwu_sgi_address: "192.168.100.1"
```

### Importante:

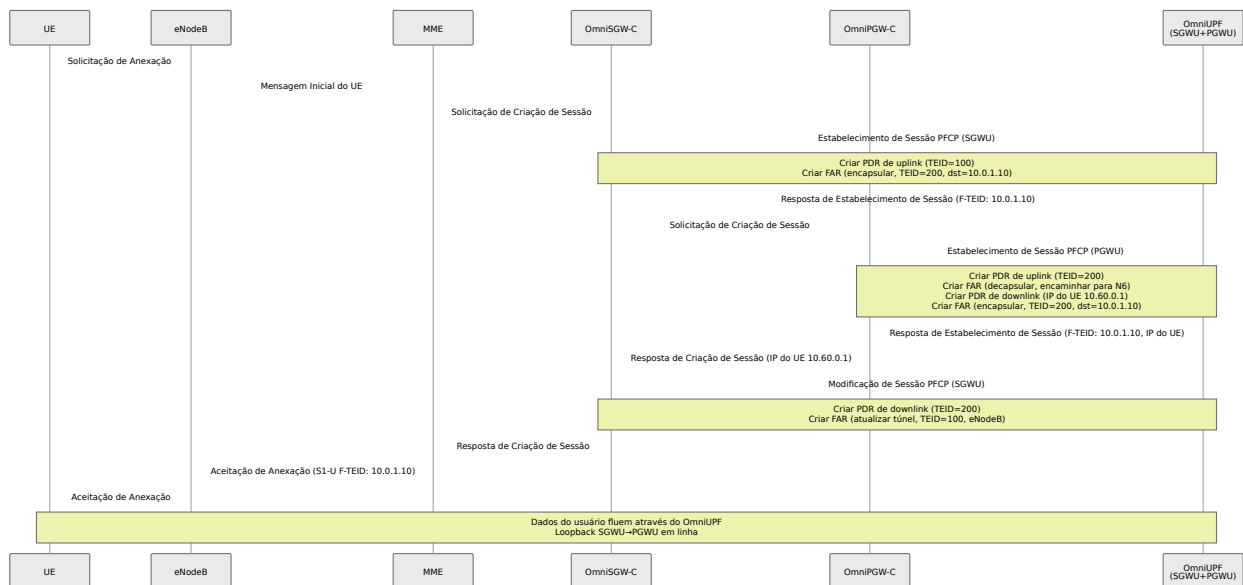
- Ambos os planos de controle conectam-se ao **mesmo ponto de extremidade PFCP** (:8805)
- OmniUPF cria **associações PFCP separadas** para SGWU-C e PGWU-C
- As sessões são isoladas por plano de controle (controladas pelo ID do Nó)

---

# Exemplo de Fluxo de Sessão

## Anexação do UE e Estabelecimento da Sessão PDU

**Cenário:** UE se conecta à rede, estabelece sessão de dados



## Sessões PFCP Criadas:

### Sessão SGWU (do OmniSGW-C):

- **PDR de Uplink:** Combinar TEID=100 (do eNodeB) → FAR: Encapsular TEID=200, dst=10.0.1.10
- **PDR de Downlink:** Combinar TEID=200 (do PGWU) → FAR: Atualizar túnel TEID=100, encaminhar para eNodeB

### Sessão PGWU (do OmniPGW-C):

- **PDR de Uplink:** Combinar TEID=200 (do SGWU) → FAR: Decapsular, encaminhar para a Internet
- **PDR de Downlink:** Combinar IP do UE=10.60.0.1 → FAR: Encapsular TEID=200, dst=10.0.1.10

# Monitoramento e Verificação

## Verificar se o Loopback N9 está Ativo

Verifique os Logs do XDP:

```
# Veja a saída de depuração eBPF em tempo real
sudo cat /sys/kernel/debug/tracing/trace_pipe | grep loopback
```

### Saída esperada:

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

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

---

## Monitorar Sessões via API REST

### Listar Associações PFCP:

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

### Saída esperada:

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

**Verifique duas associações separadas** (uma para SGWU-C, uma para PGWU-C)

---

### Listar Sessões Ativas:

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

### Saída esperada:

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

**Cada UE tem DUAS sessões:**

- Sessão do SGWU-C (TEID=100, interface S1-U)
  - Sessão do PGWU-C (TEID=200, interface N9)
- 

## Métricas de Desempenho

### Verifique as Estatísticas de Pacotes:

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

### Métricas chave:

- `xdp_processed`: Total de pacotes processados em eBPF
- `xdp_pass`: Pacotes passados para a pilha de rede (deve ser zero para tráfego de loopback)
- `xdp_redirect`: Pacotes encaminhados via redirecionamento XDP
- `xdp_tx`: Pacotes transmitidos (tráfego de loopback usa isso)

### Para tráfego de loopback N9:

- `xdp_pass` deve ser **mínimo** (apenas tráfego não-loopback)
  - `xdp_tx` ou `xdp_redirect` conta o encaminhamento de loopback
- 

## Solução de Problemas

### Tráfego N9 Indo para a Rede em vez de Loopback

**Sintoma:** Pacotes enviados para a interface de rede, alta latência

**Causa Raiz:** `n3_address`  $\neq$  `n9_address`

**Solução:**

```
# ERRADO:
n3_address: "10.0.1.10"
n9_address: "10.0.1.20" # IP diferente, sem loopback!

# CORRETO:
n3_address: "10.0.1.10"
n9_address: "10.0.1.10" # Mesmo IP, habilita loopback
```

## Verificação:

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

Deve mostrar:

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

---

## PDR Não Encontrado Após Loopback

**Sintoma:** Logs mostram `[n9-loopback]` no PDR for destination TEID

**Causa Raiz:** Sessão PGWU não criada ou incompatibilidade de TEID

### Diagnóstico:

#### 1. Verifique as Sessões PFCP:

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

#### 2. Verifique a Configuração do FAR:

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

**Solução:** Garantir que o PGWU-C crie a sessão com TEID correspondente que o SGWU-C usa para o encaminhamento N9

---

## Alta Utilização de CPU

**Sintoma:** Utilização de CPU maior que o esperado

**Causa Raiz:** Programa eBPF processando pacotes várias vezes ou buscas excessivas em mapas

**Diagnóstico:**

```
# Verifique os padrões de acesso ao mapa eBPF  
sudo bpftool map dump name pdr_map_teid_ip4 | wc -l  
sudo bpftool map dump name far_map | wc -l
```

**Solução:**

- Aumentar `max_sessions` se o mapa estiver cheio (causa falhas de lookup)
  - Verifique se a limitação de taxa QER não está causando perdas e retransmissões
  - Verifique se há buffering excessivo de pacotes
- 

## Perda de Pacotes Durante a Transferência

**Sintoma:** Pacotes descartados durante a transferência do eNodeB

**Causa Raiz:** Buffering não configurado ou limites de buffer insuficientes

**Configuração:**

```
buffer_port: 22152
buffer_max_packets: 20000 # Aumentar para redes de alta
mobilidade
buffer_max_total: 100000
buffer_packet_ttl: 30 # Ajustar com base no tempo de
transferência
```

### Verificação:

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

## Benefícios do Loopback N9

### Desempenho

Métrica	Duas Instâncias	Uma Única Instância (Loopback N9)	Melhoria
Latência	1-5 ms	< 1 $\mu$ s	<b>1000x mais rápido</b>
Throughput	Limitado pela rede	Limitado pela CPU/memória	<b>2-3x maior</b>
Uso de CPU	2x passes XDP + pilha de rede	1x passe XDP	<b>Redução de 40-50%</b>
Perda de Pacotes	Risco durante congestionamento da rede	Zero (em memória)	<b>Eliminado</b>

# Operacional

- **Implantação Simplificada:** Uma única instância OmniUPF em vez de duas
- **Infraestrutura Reduzida:** Metade dos servidores, portas de rede, endereços IP
- **Menor Complexidade:** Uma única configuração, um único ponto de monitoramento
- **Economia de Custos:** Redução de hardware, energia, refrigeração, manutenção
- **Facilidade de Solução de Problemas:** Rastreamento de pacotes único, saída de depuração eBPF única

## Casos de Uso

### Ideal Para:

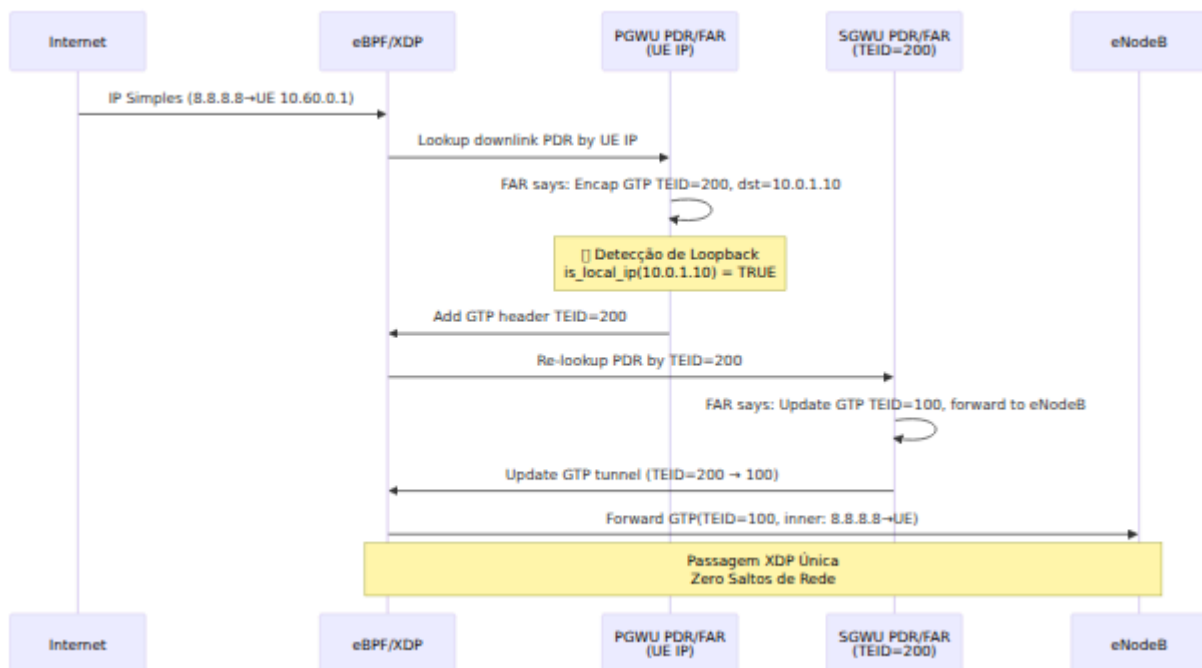
- **Computação de Borda:** Minimizar latência para breakout local
- **Implantações Pequenas/Médias:** < 100K assinantes
- **Laboratório/Teste:** Plano de usuário EPC completo em uma única VM
- **Com Orçamento Limitado:** Orçamento de hardware restrito

### Não Recomendado Para:

- **Redundância Geográfica:** SGWU e PGWU em diferentes data centers
  - **Escala Maciça:** > 1M assinantes (considere escalonamento horizontal)
  - **Requisitos Regulatórios:** Separação mandatória de SGW e PGW
-

# Comparação com Outros Modos de Implantação

## Uma Única Instância (Loopback N9) vs. Instâncias Separadas



## Resumo

O Loopback N9 permite **plano de usuário EPC 4G de grau de operadora em uma única instância OmniUPF** processando o tráfego SGWU→PGWU totalmente em eBPF sem saltos de rede. Isso proporciona:

- **Latência sub-microsegundo** para encaminhamento entre gateways
- **Redução de CPU de 40-50%** em comparação com instâncias separadas
- **Operações simplificadas** - instância única, configuração, monitoramento
- **Custo mais baixo** - metade da infraestrutura
- **Total conformidade com 3GPP** - protocolos PFCP, GTP-U padrão

**A configuração é automática** quando `n3_address == n9_address` - nenhuma bandeira ou configuração especial necessária. O caminho de dados eBPF do OmniUPF detecta condições de loopback e processa pacotes em linha.

Para mais informações:

- **Configuração:** [CONFIGURATION.md](#)
- **Arquitetura:** [ARCHITECTURE.md](#)
- **Referência de Métricas:** [METRICS.md](#)
- **Monitoramento:** [MONITORING.md](#)
- **Operações:** [OPERATIONS.md](#)
- **Solução de Problemas:** [TROUBLESHOOTING.md](#)

# Referência de Códigos de Causa PFCP

## Visão Geral

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

<b>Código</b>	<b>Nome</b>	<b>Quando Ocorre</b>
		analisável, valor de RecoveryTimeStamp inválido, F-SEID malformado.
72	<b>NoEstablishedPFCPAssociation</b>	Operação de sessão tentada sem associação ativa. Deve-se estabelecer a associação PFCP antes de criar sessões.
73	<b>RuleCreationModificationFailure</b>	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	<b>PFCPEntityInCongestion</b>	UPF enfrentando alta carga ou esgotamento de recursos. Temporariamente incapaz de processar solicitações.
75	<b>NoResourcesAvailable</b>	Recursos insuficientes para atender à solicitação. Exemplos: capacidade do mapa eBPF esgotada, falha na alocação de memória, pool de TEID esgotado.
77	<b>SystemFailure</b>	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	<b>InvalidLength</b>	O campo de comprimento do IE não corresponde ao comprimento real dos dados. Atualmente não utilizado no OmniUPF.
70	<b>InvalidForwardingPolicy</b>	Política de encaminhamento não suportada pelo UPF. Atualmente não utilizada no OmniUPF.
71	<b>InvalidFTEIDAllocationOption</b>	Opção de alocação de F-TEID não suportada. Atualmente não utilizada no OmniUPF.
76	<b>ServiceNotSupported</b>	Serviço ou funcionalidade solicitada não implementada. Atualmente não utilizada no OmniUPF.
78	<b>RedirectionRequested</b>	UPF solicita redirecionamento para outra instância do UPF. Atualmente não utilizada no OmniUPF.

## Cenários e Causas Comuns

### Falhas na Configuração da Associação

**Cenário: NodeID Ausente**

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

**Resolução:** Certifique-se de que o SMF inclua o IE NodeID em todas as Solicitações de Configuração de Associação.

### **Cenário: Formato de NodeID Inválido**

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

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

### **Cenário: Timestamp de Recuperação Ausente**

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

**Resolução:** Inclua o RecoveryTimeStamp na Solicitação de Configuração de Associação.

## **Falhas na Estabelecimento de Sessão**

### **Cenário: Nenhuma Associação Estabelecida**

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

**Resolução:** Estabeleça a associação PFCP antes de criar sessões.

### **Cenário: Falha na Criação de Regras**

```
SMF → UPF: Solicitação de Estabelecimento de Sessão
UPF processa FARs, QERs, URRs com sucesso
UPF falha ao criar PDR (mapa eBPF cheio)
UPF → SMF: Resposta de Estabelecimento de Sessão (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: Solicitação de Estabelecimento de Sessão (sem CP F-
SEID)
UPF → SMF: Resposta de Estabelecimento de Sessão (Causa:
MandatoryIEMissing)
```

**Resolução:** Inclua o CP F-SEID na Solicitação de Estabelecimento de Sessão.

## Falhas na Modificação de Sessão

### Cenário: SEID Desconhecido

```
SMF → UPF: Solicitação de Modificação de Sessão (SEID=12345)
UPF não tem sessão com SEID 12345
UPF → SMF: Resposta de Modificação de Sessão (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 que está usando a instância correta do UPF (cenários de loopback N9)

# Falhas na Exclusão de Sessão

## Cenário: SEID Desconhecido

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

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

# Solução de Problemas com Códigos de Causa

## Usando Métricas do Prometheus

Consulta ao 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 esgotamento 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 do mapa 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 do heartbeat
3. Revise os logs de configuração da associação
4. Certifique-se de que SMF e UPF podem se alcançar

### `SessionContextNotFound` na Modificação:

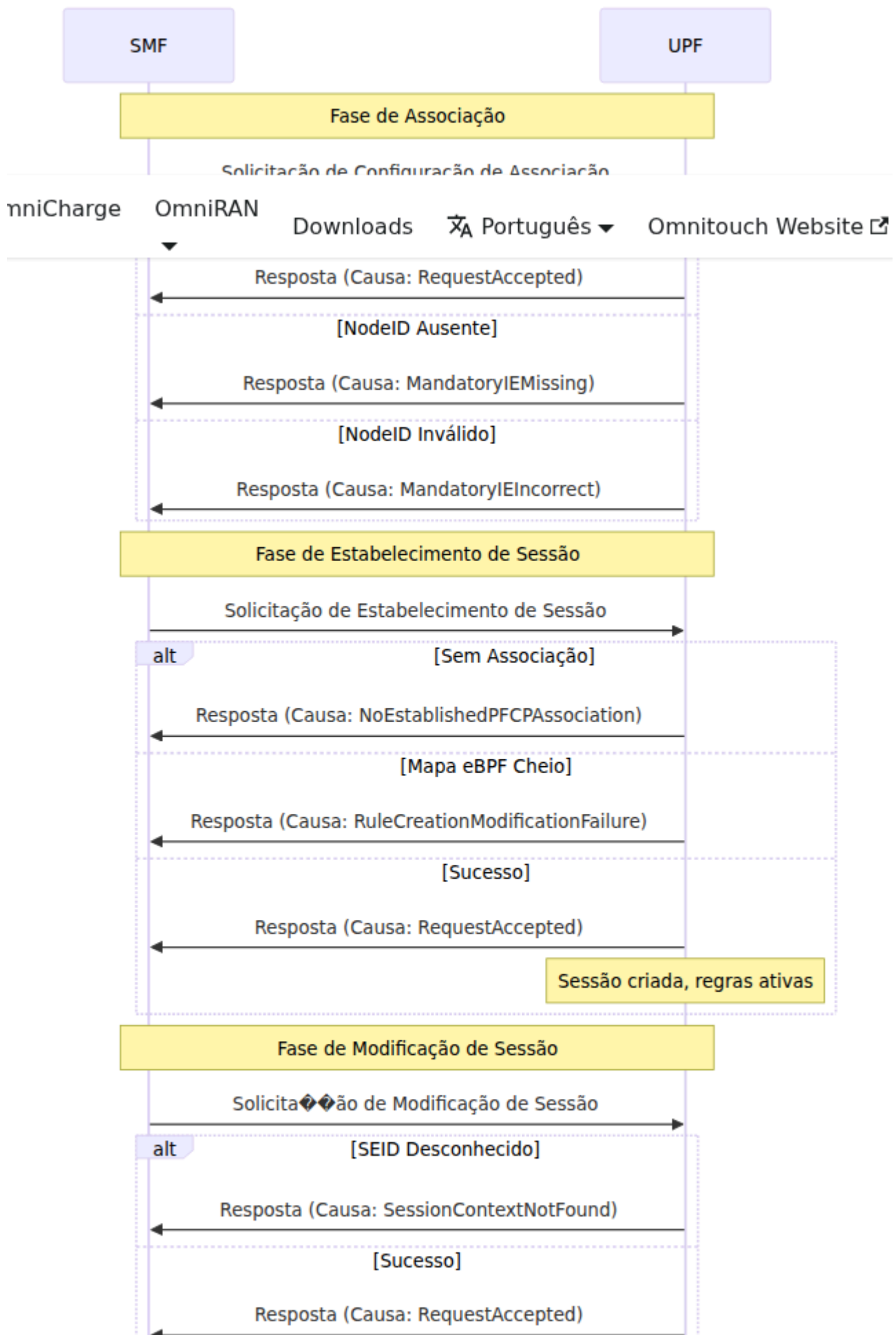
1. Verifique o SEID da resposta de estabelecimento de sessão
2. Verifique se a sessão foi excluída

3. Para loopback N9: Certifique-se de que está usando 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**





## Métricas e Alertas

### Alertas Recomendados:

```
# Crítico: Alta taxa de rejeição
- alert: PfcphighRejectionRate
  expr: |
    rate(upf_pfcpx_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_pfcpx_errors{cause_code="RuleCreationModificationFailure"}
[5m]) > 0
  annotations:
    summary: "Falhas na criação de regras PFCP detectadas"

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

rate(upf_pfcpx_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 manipulação de mensagens
- **Referência de Métricas** - Documentação da métrica upf\_pfcpx\_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

# 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 do 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 sessões de UE são estabelecidas ou encerradas, a infraestrutura de roteamento se adapta automaticamente para refletir a topologia atual da rede.

## O que é FRR?

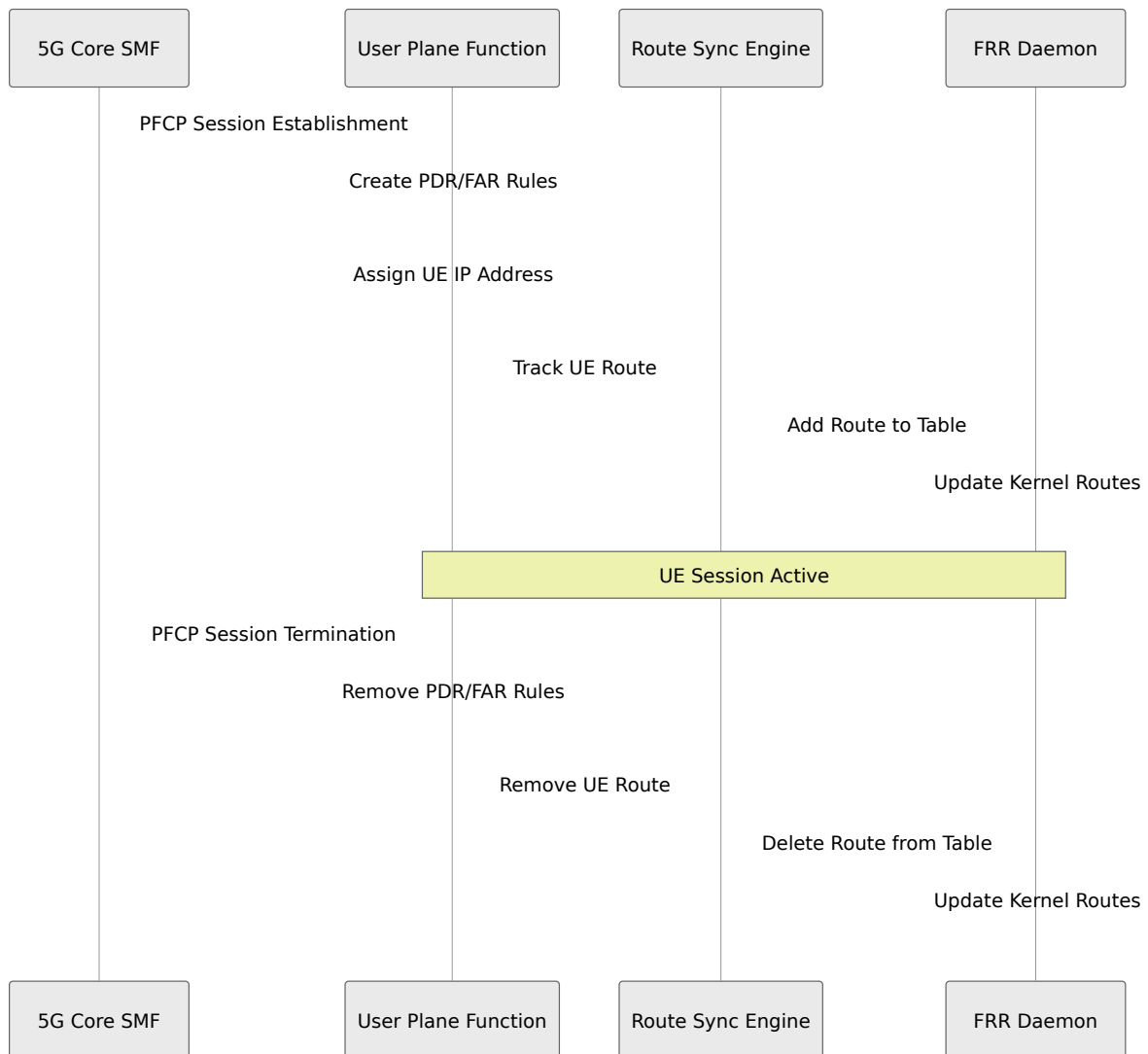
**FRR (Free Range Routing)** é um robusto conjunto de protocolos de roteamento de código aberto para plataformas Linux e Unix. Ele implementa vários protocolos de roteamento, incluindo BGP, OSPF, RIP e outros. Em nossa implantação, o FRR atua como o daemon de roteamento que mantém a tabela de roteamento 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:** Acompanha 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. **Gerencia Falhas:** Acompanha 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 **modelos Ansible** para estabelecer os parâmetros básicos de roteamento. Você define a configuração do FRR uma vez como um **modelo Jinja2** em seu playbook Ansible, e o Ansible a propaga automaticamente para todas as suas instâncias de UPF durante a implantação.

Um modelo 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 modelo 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 modelo Jinja2:

- **Parâmetros OSPF:** ID do roteador, configuração da 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 rotas, 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) via a 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** via a 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

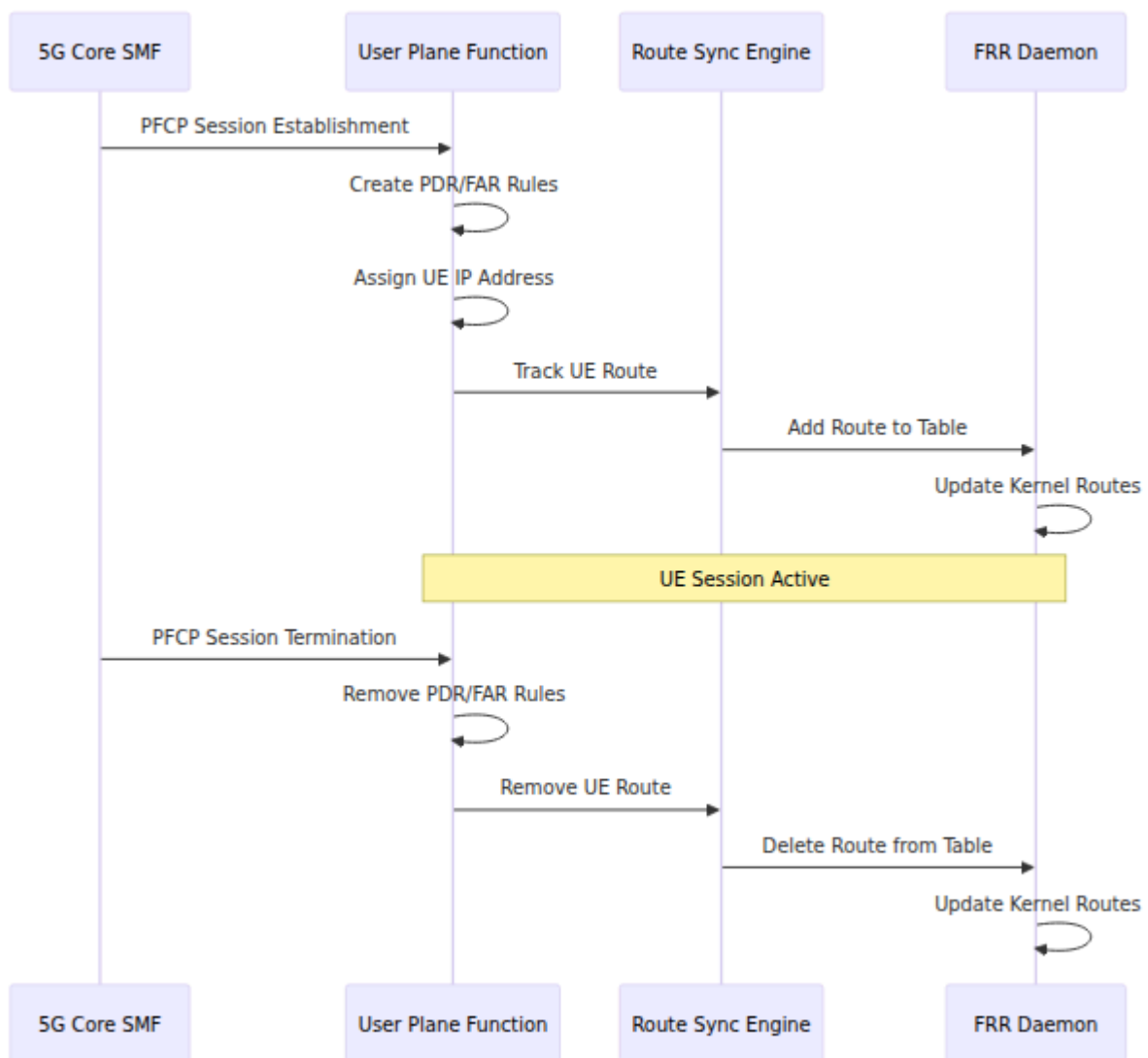
**Importante:** O UPF adiciona rotas através da interface vtysh do FRR, tornando-as rotas estáticas do FRR (não rotas do kernel). Você deve usar `redistribute static` em sua configuração OSPF/BGP, não `redistribute kernel`.

### Pontos Chave:

- **Defina Uma Vez, Implemente em Todos os Lugares:** Defina o modelo Jinja2 do FRR uma vez no Ansible, e ele é automaticamente implantado em todas as instâncias de UPF
- **Ansible gerencia configuração estática:** O modelo 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 de UE /32 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 modelo Ansible e execute novamente o playbook para alterar a configuração de roteamento em todos os UPFs simultaneamente

## Anúncio de Rotas



# Monitoramento e Gerenciamento

## Integração da 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 monitorados
- **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 do protocolo 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

PFCP Session Created

PDR/FAR Rules Installed

UE IP Tracked in Route List

Route Sync Enabled?

Yes

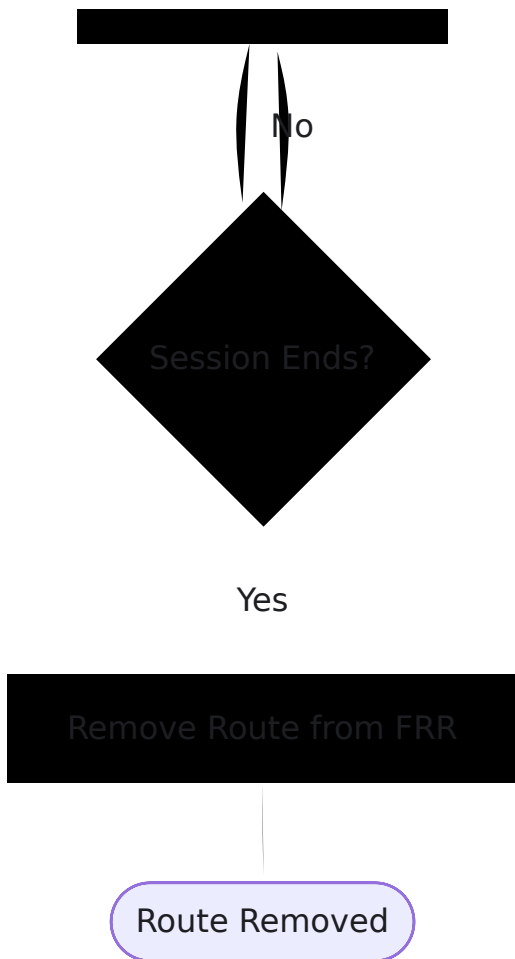
No

Push Route to FRR

Route Tracked Only

Route Active in Network

UE Traffic Flows



## Benefícios

- **Provisionamento Zero Touch:** As rotas são gerenciadas automaticamente sem intervenção manual
- **Adaptação Dinâmica:** O roteamento da rede se adapta em tempo real à mobilidade e mudanças de sessão de UE
- **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 das rotas 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 monitoradas sem sincronização
- `POST /api/v1/routes/sync` - Sincroniza rotas para o FRR e retorna a 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 da Rota

As rotas são armazenadas e gerenciadas como endereços IP simples (por exemplo, `100.64.18.5`). O daemon de roteamento gerencia todos os detalhes de entrada da 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 corretamente redistribuídas 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 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 tipo de métrica E2 (Tipo Externo 2)

Essa verificação confirma que:

1. O UPF está rastreando com sucesso o endereço IP de 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

# 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 usa um conjunto de regras interconectadas para classificar, encaminhar, modelar 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
<b>PDR</b> (Regra de Detecção de Pacotes)	Classificar pacotes em fluxos	TEID ou IP do UE	SMF via Estabelecimento/Modificação de Sessão PFCP
<b>FAR</b> (Regra de Ação de Encaminhamento)	Determinar ação de encaminhamento	ID do FAR	SMF via Estabelecimento/Modificação de Sessão PFCP
<b>QER</b> (Regra de Aplicação de QoS)	Aplicar limites de largura de banda e marcação	ID do QER	SMF via Estabelecimento/Modificação de Sessão PFCP
<b>URR</b> (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

PDR de Downlink

Chave: Endereço IP do  
UE  
IPv4 ou IPv6

ID do FAR  
ID do QER  
IDs do URR  
Modo SDF  
Filtros SDF

PDR de Uplink

Chave: TEID  
inteiro de 32 bits

ID do FAR  
ID do QER  
IDs do URR  
Remoção do Cabeçalho  
Externo

# 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 sem sinal de 32 bits
- Atribuído pelo SMF e sinalizado para 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 o 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**: Apenas o tráfego que combina com SDF é encaminhado
- **SDF + Padrão**: O tráfego que combina com SDF usa regras específicas, o 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 o 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:

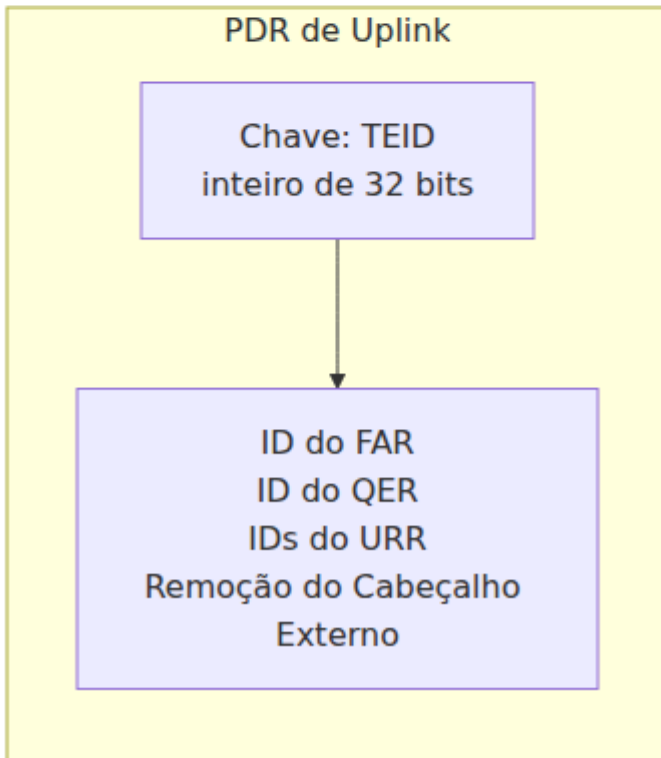
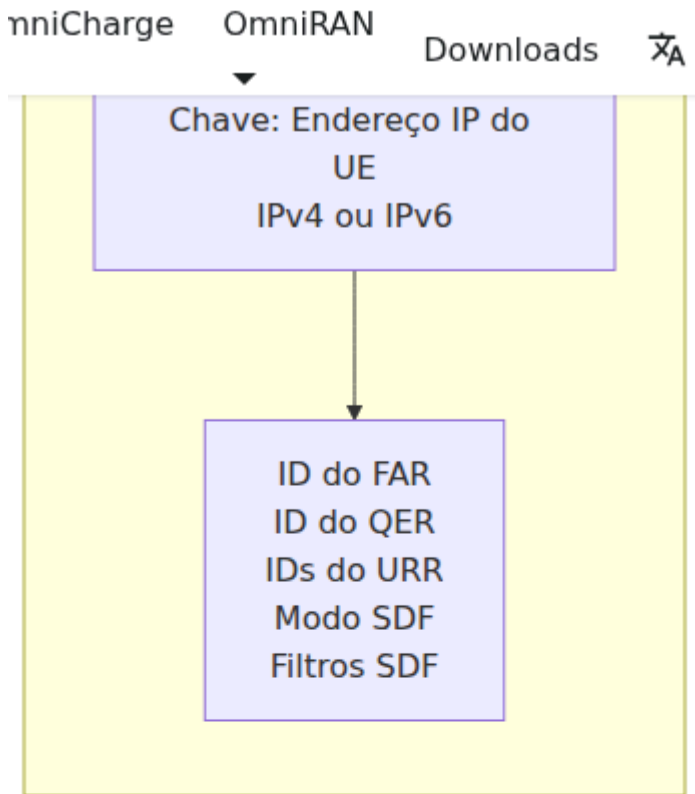
- Protocolo: UDP, Portas: 5060-5061 → ID do FAR 5 (FAR VoIP)
- Protocolo: TCP, Porta: 443 → ID do FAR 1 (FAR Padrão)

# 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 bitwise que podem ser combinadas:

Flag	Bit	Valor	Descrição
<b>ENCAMINHAR</b>	1	2	Encaminhar pacote para o destino
<b>BUFFER</b>	2	4	Armazenar pacote no buffer
<b>DESCARTAR</b>	0	1	Descartar pacote
<b>NOTIFICAR</b>	3	8	Enviar notificação para o plano de controle
<b>DUPLICAR</b>	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) - Apenas armazenar (durante a troca de caminho)
- Ação: 1 (DESCARTAR) - Descartar pacote (raro, geralmente para aplicação de políticas)

## Controle de Bufferização

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

### Quando a Bufferização é Usada

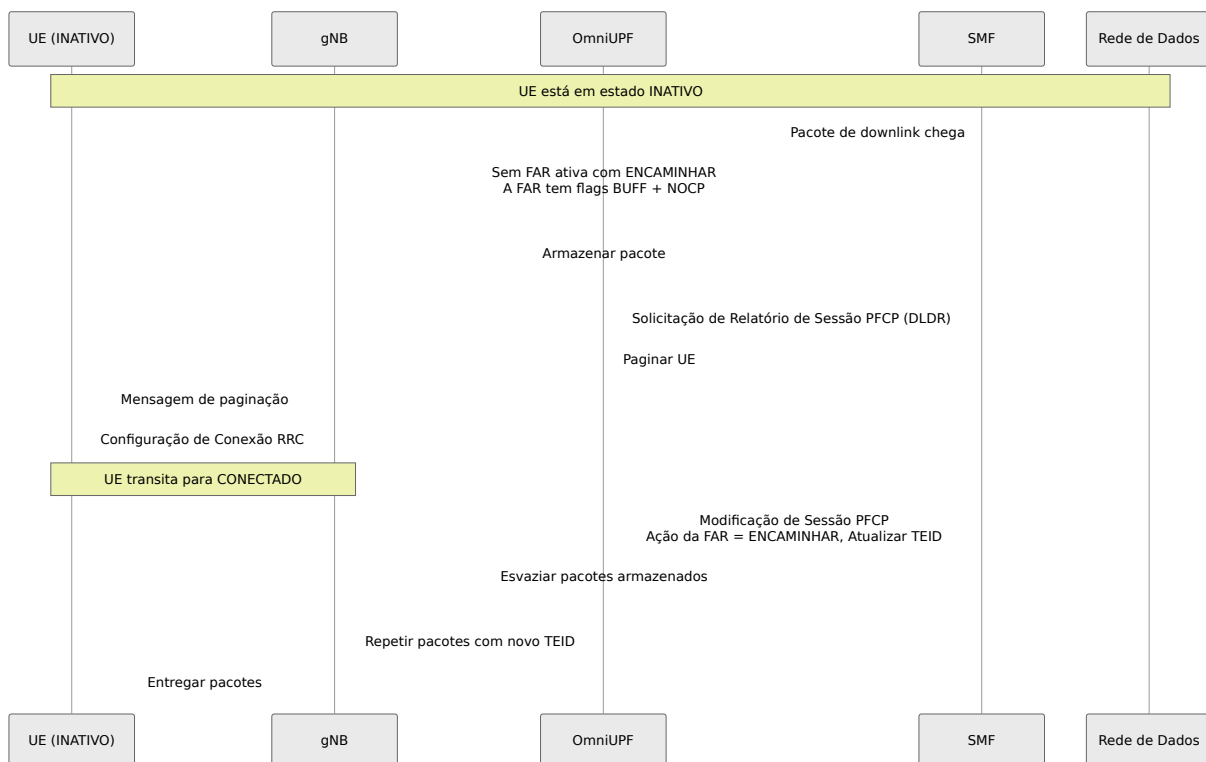
**Transição de Inativo para Conectado:** Quando pacotes de downlink chegam para um UE em estado INATIVO (não conectado ao gNB), o UPF:

1. Armazena os pacotes

2. Envia uma Notificação de Dados de Downlink (DLDR) para o SMF
3. O SMF pagina 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 um novo TEID e ação de ENCAMINHAR
6. O UPF esvazia os pacotes para o novo gNB



## Capacidade e Limites de Buffer

### Limites de Buffer Global:

- **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 um único 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 a bufferização
- **Informações de Serviço de Dados de Downlink:** QFI opcional, Indicador de Política de Paginação

#### **Ações do SMF no DLDR:**

1. Pagar 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 Bufferização

### Habilitar Bufferização (Definir flag BUFF):

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

### Modo Apenas Buffer (BUFF sem FORW):

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

### Desabilitar Bufferização (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 atualizado
- 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
- Usado quando a transferência falha ou a sessão é excluída

- Métricas rastreiam com `reason="cleared"`

## Monitorando Pacotes Armazenados

**Página de Buffers** (Interface 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 bufferização por FAR
- Operações de esvaziamento ou limpeza

### Indicadores Chave:

- **Pacotes > 10 segundos antigos:** Potencial atraso de paginação
- **Pacotes > 30 segundos antigos:** Provável falha de paginação, limpar buffer
- **Alta contagem de pacotes:** Verificar sessões travadas ou falhas de paginação

### 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 Bufferização

### Cenário 1: UE INATIVO Dados de Downlink

Estado Inicial:

- UE em modo INATIVO (sem conexão gNB)
- Ação da FAR: 0x04 (apenas 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 pagina 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 (apenas 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, 200.198.5.10)
- TEID: ID do túnel para tráfego do UE
- Ação: Adicionar cabeçalho GTP-U, encaminhar para gNB

## Busca da FAR na Interface 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 os detalhes da FAR

**Informações Exibidas:**

- ID do FAR
- Ação (numérica + flags decodificadas)
- Status de bufferização (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  
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 QoS):

- Identificador de 6 bits para fluxos 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

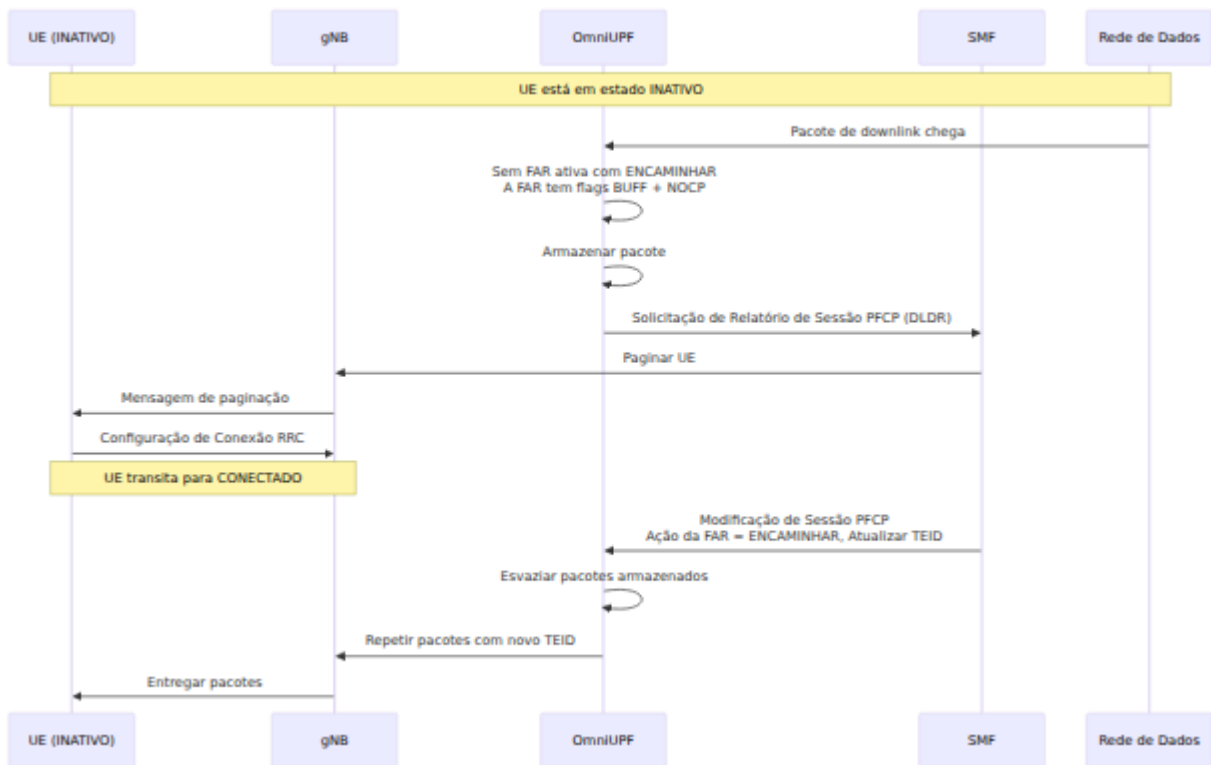
MBR Uplink: 10000 kbps (10 Mbps)

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 for **FECHADO** (não zero), descartar o pacote imediatamente
2. **Verificação de MBR:** Se  $MBR = 0$ , ignorar a limitação de taxa (largura de banda ilimitada)
3. **Cálculo do Tempo de Transmissão:**

```
tx_time = (tamanho_do_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 o pacote
5. **Avanço da Janela:** Se o pacote for permitido, avançar a janela por `tx_time`

### Exemplo de Cálculo:

Suponha:

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

Passo 1: Calcular o 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 se encaixa na janela

```
tempo_atual = 1000000000 ns  
início_da_janela = 999990000 ns  
if (início_da_janela + tx_time > tempo_atual):  
    DESCARTAR pacote (excederia o limite de taxa)
```

Passo 3: Se permitido, avançar a janela

```
início_da_janela = início_da_janela + 120 ns  
PACOTE ACEITO
```

## Comportamento da Janela Deslizante

### Tamanho da Janela de 5ms:

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

## Tratamento de Burst:

- Pequenos bursts são permitidos 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 de forma independente

## 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 aprovado, 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 aprovado, 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 verificado 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
- **Varição 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 ociosos de 5ms permitem breves bursts 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:  $(\text{volume\_delta\_bytes} \times 8) / \text{time\_delta\_seconds} / 1000 = \text{kbps}$
3. Compare com o MBR configurado na QER

## GBR (Taxa Garantida de Bits)

**Importante:** O OmniUPF **não** aplica atualmente os 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 metadado para rastreamento do tipo de fluxo (melhor esforço vs. garantido)

#### **Aprimoramento Futuro:**

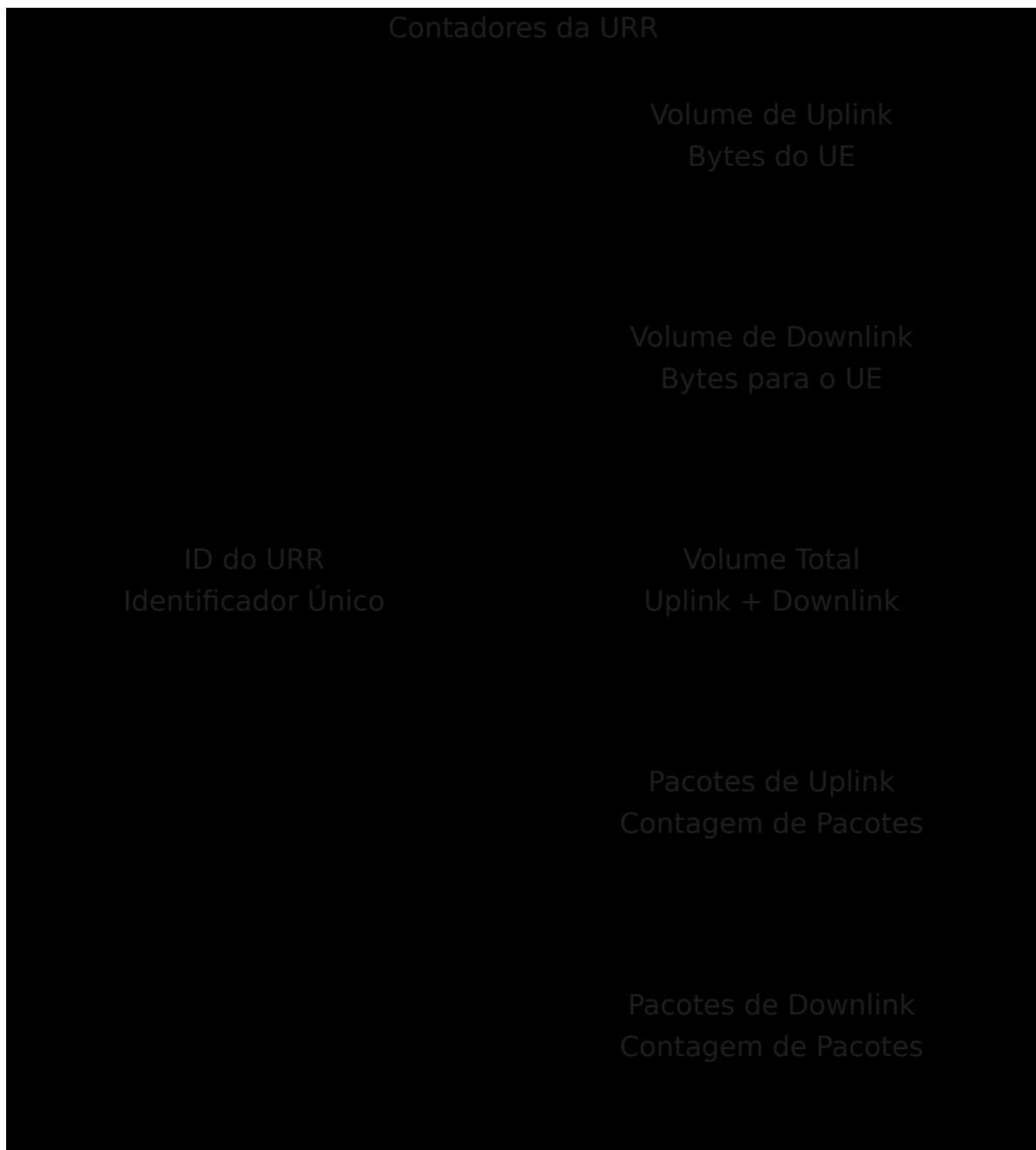
- A aplicação de GBR requer agendamento de tráfego ou enfileiramento ponderado
- Pode ser implementado usando capacidades de QoS eBPF em versões futuras

## **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 excede 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 alteração de QoS
- Relatar na transferência

## **Formatação de Exibição de Volume**

A interface Web formata automaticamente o volume em unidades legíveis:

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

## Parâmetros da QER

QFI

nniCharge

OmniRAN

Downloads

Português

OmniTouch Website

ID do QER  
Identificador Único

Estado do Portão UL  
Aberto/Fechado

Estado do Portão DL  
Aberto/Fechado

MBR Uplink  
Taxa Máxima de Bits

MBR Downlink  
Taxa Máxima de Bits

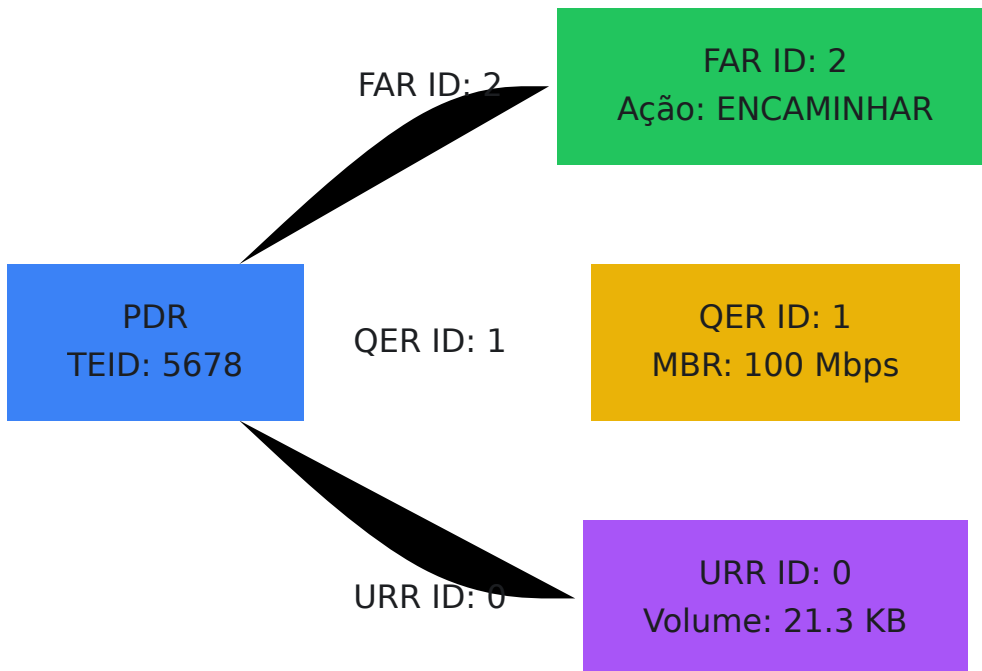
GBR Uplink  
Taxa Garantida de Bits

GBR Downlink  
Taxa Garantida de Bits

# 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: 200.198.5.10  
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 Bufferização

**Cenário:** Durante a transferência, bufferizar pacotes para prevenir perda

### Passos:

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

### Alternativa via Página de Buffers:

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

# Monitorar Conformidade com 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:

$\text{Throughput (kbps)} = (\text{Delta de Volume em bytes} \times 8) / (\text{Delta de Tempo em segundos} \times 1000)$

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

# Solução de Problemas

## Sem Fluxo de Tráfego

### 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 apenas 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 a bufferização não está presa em modo apenas BUFFER

## **Problemas de Bufferização**

#### **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 a bufferização na FAR

### **Overflow do buffer:**

1. Verifique o total de pacotes 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 a bufferização não foi desabilitada

## **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 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 Interface 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 Hypervisor
  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 uma solução de problemas mais 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 de 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 os detalhes do mapa:**

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

### **Conte as 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 os detalhes do programa XDP:**

```
bpftool prog show
```

### **Despeje as estatísticas do XDP:**

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

---

## **Depuração de Rede**

### **Capture o 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 o 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 do switch/SPAN para copiar o tráfego N3
# 3. Espelhamento de porta do switch virtual para VM de analisador

# 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 os 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

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

# Verifique o montado
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 o 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 YAML no arquivo de configuração

**Resolução:**

```
# Valide a sintaxe YAML
cat config.yml | python3 -c "import yaml, sys;
yaml.safe_load(sys.stdin)"

# Problemas comuns:
# - Indentação incorreta (use espaços, não tabs)
# - Dois pontos ausentes após chaves
# - Strings não entre aspas com caracteres especiais
# - Itens de lista sem hífen

# Exemplo de YAML correto:
cat > config.yml <<EOF
interface_name: [eth0]
xdp_attach_mode: generic
api_address: :8080
pfc_address: :8805
EOF
```

---

## 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 config.yml:
interface_name: [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 na configuração
api_address: :8081
pfcg_address: :8806
metrics_address: :9091
```

## Problema: ID do nó PFCP inválido

### Sintomas:

```
ERRO[0000] id do nó pfcf inválido: deve ser um endereço IPv4 válido
```

**Causa:** ID do nó PFCP não é um endereço IPv4 válido

### Resolução:

```
# Correto: Use endereço IP (não nome de host)
pfcf_node_id: 10.100.50.241

# Incorreto:
# pfcf_node_id: localhost
# pfcf_node_id: upf.example.com
```

---

## Problemas de Associação PFCP

### Problema: Nenhuma associação PFCP estabelecida

#### Sintomas:

- A interface da Web mostra "Sem associações"
- 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 e Resoluções:

### Firewall bloqueando PFCP

#### Resolução:

```
# Permita o tráfego PFCP (UDP 8805)
sudo ufw allow 8805/udp
sudo iptables -A INPUT -p udp --dport 8805 -j ACCEPT
```

### ID do nó PFCP errado

#### Resolução:

```
# Defina o ID do nó PFCP para o IP correto da interface N4
pfcpc_node_id: 10.100.50.241 # Deve corresponder ao IP na rede N4
```

### Rede inacessível para o 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 ausente
sudo ip route add <SMF_NETWORK>/24 via <GATEWAY>
```

## SMF configurado com IP do UPF errado

### Resolução:

- Verifique a configuração do SMF para o endereço do UPF
- Certifique-se de que o SMF tem o IP `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 e 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:

```
# Aumente o intervalo de heartbeat
heartbeat_interval: 30 # Aumente de 5 para 30 segundos
heartbeat_retries: 5 # Aumente as tentativas
heartbeat_timeout: 10 # 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:**

```
# Atualize config.yml com a interface correta
interface_name: [ens1f0] # Use o nome real da interface do 'ip
link show'
```

---

## Problema: Pacotes recebidos, mas não encaminhados (alta taxa de perda)

### Sintomas:

- Contadores RX aumentando, mas contadores TX não
- Taxa de perda > 1%

### Diagnóstico:

```
# Verifique as estatísticas de perda
curl http://localhost:8080/api/v1/xdp_stats | jq '.drop'

# Verifique as estatísticas de roteamento
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'

# Monitore as perdas de pacotes
watch -n 1 'curl -s http://localhost:8080/api/v1/packet_stats | jq
".total_rx, .total_tx, .total_drop"'
```

### Causas Comuns:

**Nenhuma correspondência PDR (TEID desconhecido ou IP do UE)**

### 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 consulta 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 ausente
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 perda 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):  
#  $\text{throughput\_kbps} = (\text{volume\_delta\_bytes} \times 8) /$   
#  $\text{time\_delta\_seconds} / 1000$

## 4. Compare MBR vs. throughput real:

- Throughput esperado  $\approx$  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 como esperado

### Resolução:

- **Se o MBR for muito baixo:** Solicite ao SMF que atualize 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 de política do SMF e o perfil QoS

### Entendendo a Aplicação do MBR:

O OmniUPF usa um algoritmo de janela deslizante para aplicar limites de MBR com precisão em nanossegundos 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 do pacote e a taxa determinam decisões de perda
- 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, taxa de tráfego sustentada limitada

---

## 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 ausente
sudo ip route add default via <N6_GATEWAY> dev eth1
```

### **Falha de Downlink (RX N6, sem TX N3):**

**Causa:** Nenhuma PDR de downlink ou encapsulamento GTP ausente

### **Resolução:**

```
# Verifique se a 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 de 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 e Resoluções:

## Kernel sem 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 de verificador
journalctl -u omniupf | grep verifier

# Problemas comuns:
# - A complexidade do eBPF excede os 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
- Aumento das perdas 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 que exclua 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:

```
# Aumente a capacidade do mapa em config.yml
max_sessions: 200000 # Aumente de 100000

# Ou defina tamanhos individuais de mapa
pdr_map_size: 400000
far_map_size: 400000
qer_map_size: 200000
```

**Importante:** Alterar tamanhos de mapa requer reinício do OmniUPF e **limpa todas as sessões existentes.**

---

# Problemas de Desempenho

## Problema: Throughput baixo (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:

```
# Mude para o modo nativo para melhor desempenho
xdp_attach_mode: native # Requer NIC/drivers compatíveis com XDP
```

### Gargalo de núcleo único

#### Resolução:

```
# Ative RSS (Receive Side Scaling) na NIC
ethtool -L eth0 combined 4 # Use 4 filas RX/TX

# Verifique se o RSS está ativado
ethtool -l eth0

# Prenda interrupções a CPUs específicas
# Veja /proc/interrupts e use irqbalance ou afinidade manual
```

### Buffer bloat

#### Resolução:

```
# Reduza os limites de buffer para diminuir a latência
buffer_max_packets: 5000
buffer_packet_ttl: 15
```

---

## Problema: Latência alta

### 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 bufferizados
curl http://localhost:8080/api/v1/upf_buffer_info | jq
'.total_packets_buffered'

# Verifique o desempenho do cache de roteamento
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'
```

### Resoluções:

#### Pacotes sendo bufferizados excessivamente

#### Resolução:

```
# Verifique por que os pacotes estão bufferizados
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 consulta FIB

### Resolução:

```
# Certifique-se de que o cache de roteamento está 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 pequenas  
rotas
```

---

## Problema: Perda de pacotes sob carga

### Sintomas:

- Taxa de perda aumenta com o tráfego
- Erros RX na NIC

### Diagnóstico:

```
# Verifique erros na NIC  
ethtool -S eth0 | grep -E "drop|error|miss"  
  
# Verifique o tamanho do buffer de anel  
ethtool -g eth0  
  
# Monitore perdas em tempo real  
watch -n 1 'ethtool -S eth0 | grep -E "drop|miss"'
```

### Resolução:

```
# Aumente o tamanho do buffer de anel RX  
ethtool -G eth0 rx 4096
```

```
# Aumente o tamanho do buffer de anel TX  
ethtool -G eth0 tx 4096
```

```
# Verifique as novas configurações  
ethtool -g eth0
```

---

# Problemas Específicos de Hypervisor

Para instruções passo a passo de configuração de hypervisor, veja o [Guia de Modos XDP](#).

## Proxmox: XDP não funciona na VM

### Sintomas:

- Não é possível anexar o programa XDP no modo nativo
- Apenas o modo genérico funciona

**Causa:** VM usando rede em ponte sem SR-IOV

### Resolução:

#### Opção 1: Use o modo genérico (mais simples)

```
xdp_attach_mode: generic
```

#### Opção 2: Configure o passthrough SR-IOV

```
# No host Proxmox:
# 1. Habilite IOMMU
nano /etc/default/grub
# Adicione: intel_iommu=on iommu=pt
update-grub
reboot

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

# 3. Atribua VF à VM na interface do Proxmox
# Hardware → Adicionar → Dispositivo PCI → Selecione VF

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

---

## VMware: Modo promíscuo necessário

### Sintomas:

- Pacotes não recebidos pelo OmniUPF

**Causa:** vSwitch bloqueando endereços MAC não correspondentes

### Resolução:

```
# Habilite o modo promíscuo no vSwitch (no vSphere Client):
# 1. Selecione 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:**

```
# Use o modo genérico (única opção)
xdp_attach_mode: generic

# Otimize as configurações do VirtualBox:
# - Use adaptador VirtIO-Net (se disponível)
# - Habilite o modo promíscuo "Permitir Tudo"
# - 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: Driver NIC não suporta XDP

**Sintomas:**

```
ERRO[0000] falha ao anexar o programa XDP: operação não suportada
```

**Diagnóstico:**

```
# Verifique o driver da NIC
ethtool -i eth0 | grep driver

# Verifique se o driver suporta XDP
modinfo <driver_name> | grep -i xdp

# Liste interfaces compatíveis com XDP
ip link show | grep -B 1 "xdpgeneric\|xdpdrv\|xdpoffload"
```

## Resolução:

### Opção 1: Use o modo genérico

```
xdp_attach_mode: generic
```

### Opção 2: Atualize o driver da NIC

```
# Verifique se há atualizações de driver (Ubuntu)
sudo apt update
sudo apt install linux-modules-extra-$(uname -r)

# Ou instale o driver específico do fornecedor
# Exemplo para Intel:
# Baixe de https://downloadcenter.intel.com/
```

### Opção 3: Substitua a NIC

```
# Use 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:

```
# Verifique os logs do kernel
dmesg | tail -100

# Verifique bugs no driver
journalctl -k | grep -E "BUG:|panic:"
```

### **Resolução:**

```
# 1. Atualize kernel e drivers
sudo apt update
sudo apt upgrade
sudo reboot

# 2. Desative o offload do XDP (use apenas nativo)
xdp_attach_mode: native

# 3. Use o modo genérico como solução alternativa
xdp_attach_mode: generic

# 4. Relate o bug ao fornecedor da NIC ou à equipe do kernel Linux
```

---

# **Falhas na Estabelecimento de Sessão**

## **Problema: Estabelecimento de sessão falha**

### **Sintomas:**

- SMF relata falha no estabelecimento de sessão
- UE não consegue estabelecer sessão PDU

Veja [Referência de Códigos de Causa PFCP](#) para cenários comuns de falha e resoluções.

### **Diagnóstico:**

```
# Verifique os logs do OmniUPF em busca de erros de sessão
journalctl -u omniupf | grep -i "estabelecimento de sessão"
```

```
# Verifique a contagem de sessões PFCP
curl http://localhost:8080/api/v1/sessions | jq 'length'
```

```
# Capture o 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:

```
# Verifique o uso do mapa
curl http://localhost:8080/api/v1/map_info | jq '.[0] |
select(.usage_percent > 90)'
```

# Aumente a capacidade (veja a seção sobre mapa eBPF cheio acima)

### Parâmetros PDR/FAR inválidos

#### Resolução:

```
# Verifique os logs do OmniUPF em busca de erros de validação
journalctl -u omniupf | grep -E "inválido|erro" | tail -20
```

# Problemas comuns:

- # - Endereço IP do UE inválido (0.0.0.0 ou duplicado)
- # - TEID inválido (0 ou duplicado)
- # - FAR ausente para PDR
- # - Ação FAR inválida

# Verifique a configuração do SMF e os parâmetros da sessão

### Recurso não suportado (UEIP/FTUP)

#### Resolução:

```
# Habilite recursos necessários, se necessário
feature_ueip: true # Alocação de IP do UE pelo UPF
ueip_pool: 10.60.0.0/16

feature_ftup: true # Alocação de F-TEID pelo UPF
teid_pool: 100000
```

# Problemas de Bufferização

## Problema: Pacotes presos no buffer

### Sintomas:

- Contagem de pacotes bufferizados aumentando
- Pacotes não entregues após transferência

### Diagnóstico:

```
# Verifique as estatísticas do buffer
curl http://localhost:8080/api/v1/upf_buffer_info

# Verifique os buffers individuais do FAR
curl http://localhost:8080/api/v1/upf_buffer_info | jq '.buffers[]
| {far_id, packet_count, oldest_packet_ms}'

# Monitore o tamanho do buffer
watch -n 5 'curl -s http://localhost:8080/api/v1/upf_buffer_info |
jq ".total_packets_buffered"'
```

### Causas e Resoluções:

#### **FAR nunca atualizado para FORWARD**

**Causa:** SMF nunca enviou Modificação de Sessão PCF para aplicar FAR

#### **Resolução:**

```
# Verifique o 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 travado no estado BUFF, solicite ao SMF que:
# - Envie a Solicitação de Modificação de Sessão PFCP
# - Atualize o FAR com a ação FORW
```

## TTL do buffer expirado

**Causa:** Pacotes expiraram antes da atualização do FAR

### Resolução:

```
# Aumente o TTL do buffer
buffer_packet_ttl: 60 # Aumente de 30 para 60 segundos
```

## Overflow do buffer

**Causa:** Muitos pacotes bufferizados por FAR

### Resolução:

```
# Aumente os limites de buffer
buffer_max_packets: 20000 # Por FAR
buffer_max_total: 200000 # Limite global
```

---

# Depuração Avançada

## Habilitar Registro de Depuração

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

```
# Reinicie o OmniUPF com registro de depuração
```

```
sudo systemctl restart omniupf
```

```
# Monitore os logs em tempo real
```

```
journalctl -u omniupf -f --output cat
```

---

## Rastreamento do Programa eBPF

```
# Rastreie a execução do programa eBPF (requer bpftrace)
```

```
sudo bpftrace -e 'tracepoint:xdp:* { @[probe] = count(); }'
```

```
# Rastreie 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 de Captura de Pacotes do XDP:

O XDP processa pacotes **antes** da pilha de rede do kernel, portanto, o `tcpdump` padrão **não pode 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: Use a 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: Capture o 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 do switch para capturar
tráfego
# Exemplos:
# - TAP físico entre gNB e UPF
# - Espelhamento/mirror de porta do switch copiando tráfego N3
para analisador
# - Espelhamento de porta de switch virtual no hypervisor
#
# No host de captura (NÃO no 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 o 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 o espelhamento de porta do switch virtual para enviar
tráfego do UPF para a VM de analisador
# Exemplo: bridge Linux com tcpdump em VM diferente
# No hypervisor, espelhe a interface N3 do UPF para a interface do
analisador

# Na VM de analisador:
tcpdump -i eth1 -n udp port 2152 -w /tmp/n3_virtual.pcap
```

### **Por que a Captura Fora de Banda é Necessária:**

- O XDP contorna completamente a pilha de rede do kernel
- Pacotes são processados no driver da NIC ou no hardware
- O tcpdump baseado em host vê pacotes APÓS o processamento do 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. Colete 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. **Relate o problema** com:

- Versão do OmniUPF
- Versão do kernel Linux
- Diagrama de topologia de rede
- Arquivo de configuração (redigir informações sensíveis)
- Trechos de log relevantes
- Etapas para reproduzir

---

# Documentação Relacionada

- **Guia de Configuração** - Parâmetros de configuração e exemplos
- **Guia de Arquitetura** - Internos do eBPF/XDP e ajuste de desempenho
- **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

# 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:

- **Sessões** - `/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):

<b>Coluna</b>	<b>Descrição</b>
<b>Node ID</b>	Identificador do nó SMF ou PGW-C (FQDN ou IP)
<b>Endereço</b>	Endereço IP do SMF/PGW-C para comunicação PFCP
<b>Próximo ID de Sessão</b>	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
<b>Local SEID</b>	Identificador de ponto final de sessão atribuído pelo UPF
<b>Remote SEID</b>	Identificador de ponto final de sessão atribuído pelo SMF
<b>UE IP</b>	Endereço IPv4 ou IPv6 do equipamento do usuário
<b>TEID</b>	Identificador de Ponto de Túnel GTP-U para tráfego de uplink
<b>PDRs</b>	Número de regras de detecção de pacotes na sessão
<b>FARs</b>	Número de regras de ação de encaminhamento na sessão
<b>QERs</b>	Número de regras de aplicação de QoS na sessão
<b>URRs</b>	Número de regras de relatório de uso na sessão
<b>Ações</b>	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
<b>FAR ID</b>	Identificador único da regra de encaminhamento
<b>Ação</b>	Flags de ação de encaminhamento (FORWARD, DROP, BUFFER, DUPLICATE, NOTIFY)
<b>Buffering</b>	Status atual de buffering (Ativado/Desativado)
<b>Destino</b>	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
<b>QER ID</b>	Identificador único da regra de QoS (clicável quando referenciado a partir dos PDRs)
<b>MBR (Uplink)</b>	Taxa máxima de bits para tráfego de uplink (kbps)
<b>MBR (Downlink)</b>	Taxa máxima de bits para tráfego de downlink (kbps)
<b>GBR (Uplink)</b>	Taxa garantida de bits para tráfego de uplink (kbps)
<b>GBR (Downlink)</b>	Taxa garantida de bits para tráfego de downlink (kbps)
<b>QFI</b>	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
<b>URR ID</b>	Identificador único da regra de relatório de uso (clicável quando referenciado a partir dos PDRs)
<b>Volume de Uplink</b>	Bytes enviados da UE para a rede de dados
<b>Volume de Downlink</b>	Bytes enviados da rede de dados para a UE
<b>Volume Total</b>	Total de bytes em ambas as direções
<b>Ações</b>	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
<b>FAR ID</b>	Identificador da regra de ação de encaminhamento
<b>Contagem de Pacotes</b>	Número de pacotes armazenados para este FAR
<b>Contagem de Bytes</b>	Total de bytes armazenados para este FAR
<b>Pacote Mais Antigo</b>	Timestamp do pacote armazenado mais antigo
<b>Pacote Mais Novo</b>	Timestamp do pacote armazenado mais novo
<b>Ações</b>	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
<b>Nome do Mapa</b>	Nome do mapa eBPF (por exemplo, <code>uplink_pdr_map</code> , <code>far_map</code> )
<b>Usado</b>	Número de entradas atualmente no mapa
<b>Capacidade</b>	Máximo de entradas permitidas no mapa
<b>Uso</b>	Barra de progresso visual com porcentagem
<b>Tamanho da Chave</b>	Tamanho das chaves do mapa em bytes
<b>Tamanho do Valor</b>	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:

<b>Coluna</b>	<b>Descrição</b>
<b>Índice</b>	Número do índice da rota
<b>Endereço IP da UE</b>	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
<b>Neighbor ID</b>	Identificador do roteador OSPF
<b>Endereço</b>	Endereço IP do vizinho OSPF
<b>Interface</b>	Interface usada para a adjacência OSPF
<b>Estado</b>	Estado da adjacência OSPF (Completo, Inicial, etc.)
<b>Prioridade</b>	Valor de prioridade OSPF
<b>Tempo de Atividade</b>	Duração que o vizinho está ativo
<b>Tempo de Morte</b>	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
<b>Neighbor IP</b>	Endereço IP do par BGP
<b>ASN</b>	Número do Sistema Autônomo do par
<b>Estado</b>	Estado da sessão BGP (Estabelecido, Ocioso, etc.)
<b>Up/Down</b>	Duração do estado atual
<b>Prefixos Recebidos</b>	Número de prefixos de rota recebidos do par
<b>Msg Enviadas</b>	Total de mensagens BGP enviadas ao par
<b>Msg Recebidas</b>	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:

<b>Coluna</b>	<b>Descrição</b>
<b>Link State ID</b>	Identificador LSA (tipicamente o endereço da rede)
<b>Máscara</b>	Máscara de rede para a rota
<b>Roteador Anunciante</b>	ID do roteador que anuncia esta rota externa
<b>Tipo de Métrica</b>	Tipo de métrica externa OSPF (E1 ou E2)
<b>Métrica</b>	Métrica de custo OSPF para a rota
<b>Idade</b>	Tempo desde que a LSA foi originada (segundos)
<b>Seq Number</b>	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:

<b>Campo</b>	<b>Descrição</b>
<b>Nome da Interface</b>	Interface de rede usada para XDP (por exemplo, eth0, ens1f0)
<b>Driver</b>	Nome do driver de rede (por exemplo, i40e, ixgbe, virtio_net)
<b>Versão do Driver</b>	String da versão do driver
<b>Modo Atual</b>	Modo XDP ativo (DRV, SKB ou NENHUM)
<b>Contagem de Múltiplas Filas</b>	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 (X) 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:  $\text{Gbps} = \text{Mpps} \times \text{Packet\_Size} \times 8 / 1000$
- 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:  $\text{Gbps} = \text{Mpps} \times (\text{Packet\_Size} - 50) / 1000$

### 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:  $10 \text{ Mpps} \times 1200 \text{ bytes} \times 8 \text{ bits/byte} \div 1000 = 96 \text{ Gbps}$

## 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

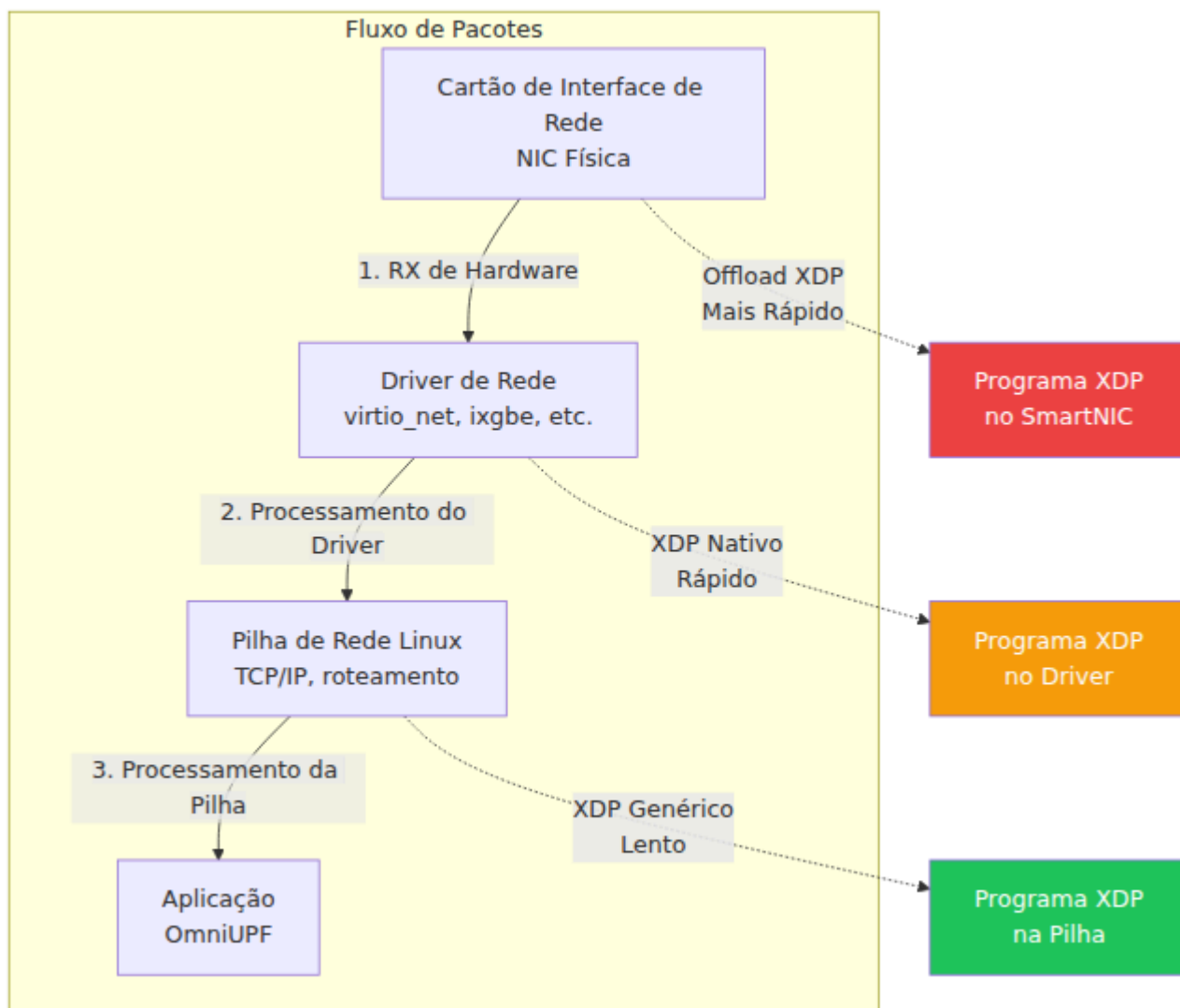
## Índice

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 com XDP](#)
- 

## Visão Geral

OmniUPF utiliza **XDP (eXpress Data Path)** para processamento de pacotes de alto desempenho. O 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 em nível de produção.

---

# Comparação de Modos XDP

Aspecto	Modo Genérico	Modo Nativo	Modo Offload
<b>Ponto de Anexação</b>	Pilha de rede Linux	Driver de rede	Hardware NIC
<b>Desempenho</b>	~1-2 Mpps	~5-10 Mpps	~10-40 Mpps
<b>Latência</b>	~100 µs	~10 µs	~1 µs
<b>Uso de CPU</b>	Alto	Médio	Baixo
<b>Requisitos de NIC</b>	Qualquer NIC	Driver compatível com XDP	SmartNIC com suporte a XDP
<b>Suporte a Hipervisores</b>	Todos os hipervisores	A maioria (requer multi-queue)	Raro (PCI passthrough)
<b>Caso de Uso</b>	Testes, desenvolvimento	<b>Produção (recomendado)</b>	Sites de borda de alto throughput
<b>Configuração</b>	<code>xdp_attach_mode: generic</code>	<code>xdp_attach_mode: native</code>	<code>xdp_attach_mode: offload</code>

**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 microssegundos por pacote
- **Sobrecarga de CPU:** Alta (pacote copiado para a pilha do kernel antes do XDP)

## Quando Usar

- **Apenas desenvolvimento e testes**
- **Ambientes de laboratório** onde o desempenho não importa
- **Implantação inicial** para verificar a funcionalidade antes de otimizar

## Configuração

```
# config.yaml
interface_name: [eth0]
xdp_attach_mode: generic # Modo padrão
```

**Aviso:** O modo genérico **não é adequado para produção**. Ele causará gargalos em altas taxas de pacotes e desperdiçará recursos de CPU.

---

## 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 fornece desempenho próximo ao hardware, mantendo a flexibilidade em nível de kernel.

# Características de Desempenho

- **Throughput:** ~5-10 milhões de pacotes por segundo (Mpps) por núcleo
- **Latência:** ~10 microssegundos por pacote
- **Sobrecarga de CPU:** Baixa (pacote processado no nível do driver)
- **Escalonamento:** Escalonamento linear com núcleos de CPU e filas de NIC

## Quando Usar

- **Implantações em produção** (recomendado)
- **Redes de grau de transportadora** que requerem alto throughput
- **Cenários de computação em borda** com requisitos de desempenho
- **Qualquer implantação** onde o desempenho importa

## 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: `ixgbevfn`, `i40evfn` (PCI passthrough) ✓

**Nota:** O VirtualBox **não** suporta XDP nativo (use apenas o modo genérico).

# Configuração

```
# config.yaml
interface_name: [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 Offload XDP executa o programa eBPF **diretamente no hardware da NIC** (SmartNIC), contornando completamente a CPU para o processamento de pacotes. Isso fornece o desempenho mais alto, mas requer hardware especializado.

### Características de Desempenho

- **Throughput:** ~10-40 milhões de pacotes por segundo (Mpps)
- **Latência:** ~1 microssegundo por pacote
- **Sobrecarga de CPU:** Quase zero (processamento na NIC)

### Quando Usar

- **Implantações de ultra-alto throughput** (10G+ por instância de 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

```
# config.yaml
interface_name: [eth0]
xdp_attach_mode: offload
```

---

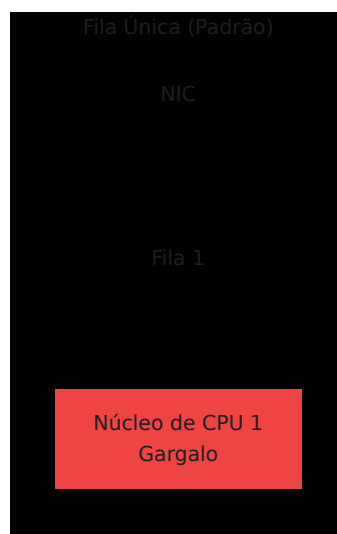
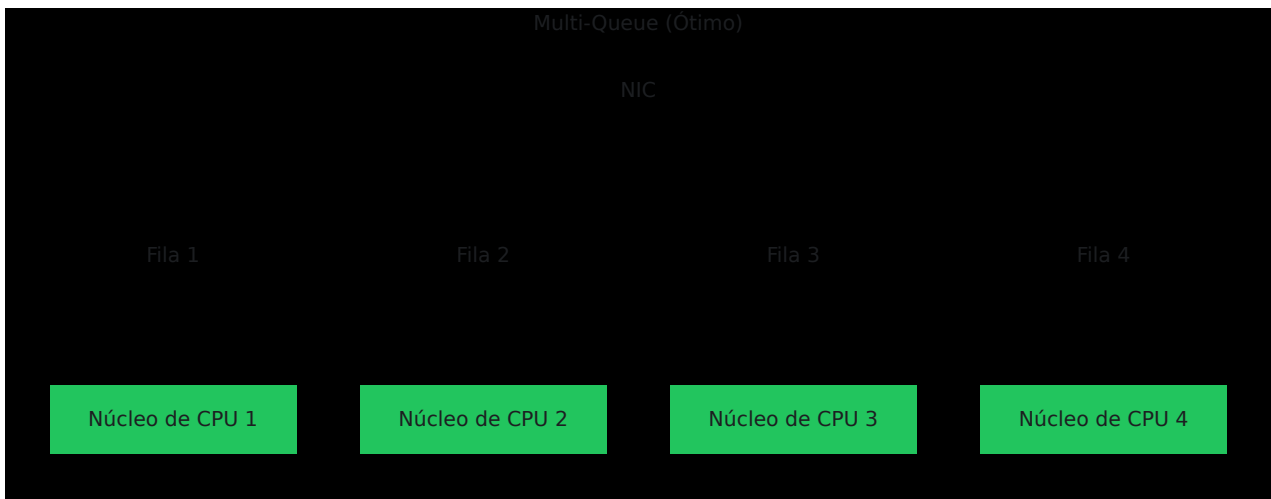
# Habilitando XDP Nativo no Proxmox VE

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: Compreendendo o Requisito

### Por que Multi-Queue Importa:

- **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: Através da Interface Web do Proxmox

- 1. Desligue completamente a VM** (não apenas reinicie)
  - Selecione sua VM na interface web do Proxmox
  - Clique em **Desligar**
- 2. Edite o Dispositivo de Rede**
  - Vá para a aba **Hardware**
  - Clique no seu dispositivo de rede (por exemplo, `net0`)
  - Clique em **Editar**
- 3. Defina Multi-Queue**

- Encontre o campo "**Multiqueue**"
- Defina para **8** (ou corresponda à sua contagem de vCPU, máximo 16)
- Clique em **OK**

#### 4. Inicie a VM

- Clique em **Iniciar**

#### Opção B: Através da Linha de Comando do Proxmox

```
# SSH no 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, depois inicie
qm start XXX
```

#### Recomendações de Contagem de Filas:

- **4 filas:** Mínimo para produção (bom para VMs de 2-4 vCPU)
- **8 filas:** Recomendado para a maioria das implantações (4-8 vCPU)
- **16 filas:** Máximo para alto desempenho (8+ vCPU)

### 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 -ld /sys/class/net/eth0/queues/rx-* | wc -l
ls -ld /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 /config.yaml
```

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 minuto atrás" | 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  $\geq 5.15$ )
- 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 **totalmente 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 superposição)
  - Monitore `top` - o uso da 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. **Edite as configurações da VM:**
  - Clique com o botão direito na VM → Editar Configurações
  - Adaptador de Rede → Avançado
  - Defina **Receive Side Scaling** como **Habilitado**
3. **Edite o 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
```

#### Configure o OmniUPF:

```
interface_name: [ens192] # 0 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-de-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
```

## Configure o OmniUPF:

```
interface_name: [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 ganchos 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 minutos atrás" | 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 o 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,         # Passou 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 com 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 mais antigo
- Certifique-se de que o driver virtio\_net está 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 do 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 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

# 0 tráfego deve ser distribuído entre múltiplas filas
```

### 2. Verifique a utilização da CPU:

```
# Monitore o uso da CPU por núcleo
mpstat -P ALL 1

# Deve-se 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 a superposição 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 com 100% de capacidade
```

## Solução:

- Aumente os tamanhos dos mapas eBPF na configuração
- Verifique pacotes corrompidos que causam erros no eBPF
- Verifique se o suporte eBPF do 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 totalmente desligada:

```
# No host Proxmox
qm status YOUR_VM_ID

# Deve mostrar "status: stopped" antes de iniciar
```

### Solução:

```
# No host Proxmox
# Desligue a força e reinicie
qm shutdown YOUR_VM_ID
sleep 10
qm start YOUR_VM_ID

# Então 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 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	<b>5.5x mais rápido</b>
Latência	95 $\mu$ s	12 $\mu$ s	<b>8x menor</b>
Uso de CPU (1 Gbps)	85% (1 núcleo)	15% (distribuído)	<b>5x mais eficiente</b>
Throughput Máximo	~1.2 Gbps	~10 Gbps	<b>8x maior</b>

**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

Estas NICs foram verificadas para suportar o modo XDP nativo com OmniUPF:

#### NICs Intel (Recomendadas para Bare Metal)

<b>Modelo</b>	<b>Velocidade</b>	<b>Driver</b>	<b>Suporte a XDP</b>	<b>Notas</b>
<b>Intel X520</b>	10GbE	ixgbe	Nativo ✓	Comprovada, amplamente disponível, bom preço/desempenho
<b>Intel X710</b>	10/40GbE	i40e	Nativo ✓	Excelente suporte a multi-queue
<b>Intel E810</b>	100GbE	ice	Nativo ✓	Última geração, melhor desempenho
<b>Intel i350</b>	1GbE	igb	Nativo ✓ (kernel 5.10+)	Bom para necessidades de largura de banda mais baixa

### **NICs Mellanox/NVIDIA (Alto Desempenho)**

<b>Modelo</b>	<b>Velocidade</b>	<b>Driver</b>	<b>Suporte a XDP</b>	<b>Notas</b>
<b>ConnectX-4</b>	25/50/100GbE	mlx5	Nativo ✓	Alto throughput, bom para computação em borda
<b>ConnectX-5</b>	25/50/100GbE	mlx5	Nativo ✓	Excelente desempenho, aceleração de hardware
<b>ConnectX-6</b>	50/100/200GbE	mlx5	Nativo ✓	Última geração, melhor para ultra-alto throughput
<b>BlueField-2</b>	100/200GbE	mlx5	Nativo ✓	SmartNIC com capacidades de DPU

### **NICs Broadcom**

<b>Modelo</b>	<b>Velocidade</b>	<b>Driver</b>	<b>Suporte a XDP</b>	<b>Notas</b>
<b>Série BCM57xxx</b>	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
<b>Proxmox/KVM</b>	VirtIO	virtio_net	Nativo ✓	Sim (configurável)	<b>Melhores VMs</b>
<b>VMware ESXi</b>	vmxnet3	vmxnet3	Nativo ✓	Sim	Requer 6.7+
<b>Hyper-V</b>	NIC Sintética	hv_netvsc	Nativo ✓	Sim	Windows Server 2016
<b>AWS</b>	ENA	ena	Nativo ✓	Sim	Instâncias meta
<b>VirtualBox</b>	Qualquer	vários	Apenas genérico ☐	Não	Não recomendado para produção

## NICs com Suporte a Offload de Hardware

**Offload XDP verdadeiro** (eBPF roda na NIC):

Fornecedor	Modelo	Velocidade	Notas
<b>Netronome</b>	Agilio CX 10G	10GbE	Apenas suporte a offload XDP confirmado
<b>Netronome</b>	Agilio CX 25G	25GbE	Requer firmware especial
<b>Netronome</b>	Agilio CX 40G	40GbE	Muito caro (~\$2,500-5,000)
<b>Netronome</b>	Agilio CX 100G	100GbE	Apenas para empresas

**Nota:** NICs de offload de hardware são raras, caras e requerem implantação bare metal. A maioria das implantações deve usar XDP nativo em vez disso.

## Configurações Testadas

Estas configurações foram verificadas com OmniUPF em produção:

### Opção de Orçamento (1-10 Gbps)

- **NIC:** Intel X520 (10GbE de porta dupla)
- **Modo:** XDP Nativo
- **Throughput:** ~8-10 Gbps por instância de UPF
- **Custo:** ~\$100-200 (usado/refurbished)

### Faixa Intermediária (10-50 Gbps)

- **NIC:** Intel X710 (40GbE) ou Mellanox ConnectX-4 (25GbE)
- **Modo:** XDP Nativo
- **Throughput:** ~25-40 Gbps por instância de UPF
- **Custo:** ~\$300-800

### Alto Desempenho (50-100+ Gbps)

- **NIC:** Mellanox ConnectX-5/6 (100GbE)
- **Modo:** XDP Nativo
- **Throughput:** ~80-100 Gbps por instância de UPF
- **Custo:** ~\$1,000-2,500

### Implantações de VM (Proxmox/KVM)

- **NIC:** VirtIO com 8-16 filas
- **Modo:** XDP Nativo
- **Throughput:** ~5-10 Gbps por instância de UPF
- **Custo:** Sem custo adicional de hardware

## O Que NÃO Comprar

Evite estes para implantações de produção do OmniUPF:

NIC/Plataforma	Razão	Alternativa
<b>NICs Realtek</b>	Sem suporte a XDP, drivers Linux ruins	Intel i350 ou melhor
<b>VirtualBox</b>	Sem suporte a XDP nativo	Migre para Proxmox/KVM
<b>NICs de consumo</b>	Suporte limitado a filas, não confiáveis	Intel/Mellanox de servidor
<b>NICs muito antigas (&lt;2014)</b>	Sem suporte a driver XDP	Intel X520 ou mais recente

## Lista de Verificação Pré-Compra

Antes de comprar hardware, verifique:

1.  **Suporte do Driver:** Verifique se o driver Linux suporta XDP

```
# Em sistema semelhante
modinfo <driver_name> | grep -i xdp
```

2. **Versão do Kernel:** Certifique-se de que o kernel  $\geq 5.15$  para suporte confiável ao XDP

```
uname -r
```

3. **Multi-Queue:** Verifique se a NIC suporta múltiplas filas (RSS/VMDq)
4. **Largura de Banda PCI:** Certifique-se de que o slot PCIe tem faixas 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 física necessária
- VM: Suporte a VirtIO ou SR-IOV necessário
- Contêiner: Configuração da NIC do host herdada

**⚠ Não compre hardware com base apenas neste guia - sempre confirme com o suporte da Omnitouch primeiro!**

---

## Recursos Adicionais

- **Guia de Configuração:** [CONFIGURATION.md](#) - Referência completa de configuração
- **Guia de Resolução 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 no 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/config.yaml # 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
```

# Guia de Operações do OmniUPF

## Índice

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

## Visão Geral

OmniUPF (Função do Plano do Usuário baseada em eBPF) é uma Função do Plano do Usuário 5G/LTE de alto desempenho que fornece encaminhamento de pacotes de nível carrier-grade, 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 oferece a infraestrutura central de processamento de pacotes necessária para redes 5G SA, 5G NSA e LTE.

## O que é uma Função do Plano do Usuário?

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

## Principais Capacidades do OmniUPF

### Processamento de Pacotes:

- Processamento de pacotes do plano do usuário totalmente compatível com 3GPP
- Caminho de dados baseado em eBPF para desempenho em nível de kernel
- Encapsulamento e desencapsulamento 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 aplicativos
- 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 da capacidade do mapa 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 do usuário)
- Escalabilidade multicanal
- Capaz de offload para aceleração de hardware
- Otimizado para implantações nativas em nuvem

Para detalhes sobre o uso do painel de controle, consulte [Operações da Web UI](#).

# Entendendo a Arquitetura do Plano do Usuário

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

- **UPF (Função do 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)

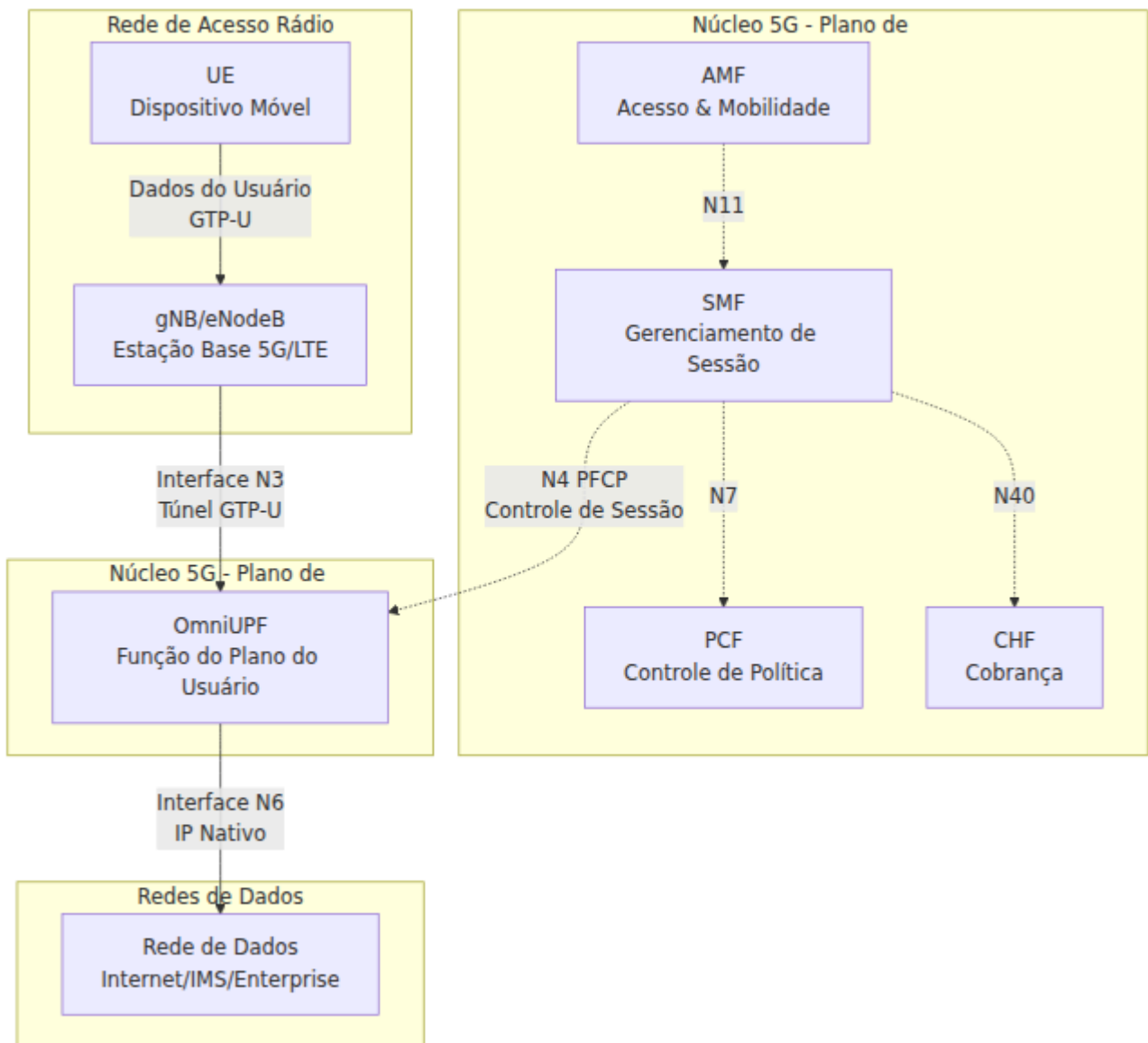
OmniUPF pode operar em **qualquer combinação** desses modos:

- **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 o protocolo PFCP, fornecendo 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 se posiciona no plano de dados das redes 5G, fornecendo a camada de encaminhamento de pacotes de alta velocidade que conecta dispositivos móveis a redes e serviços de dados.



## 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.

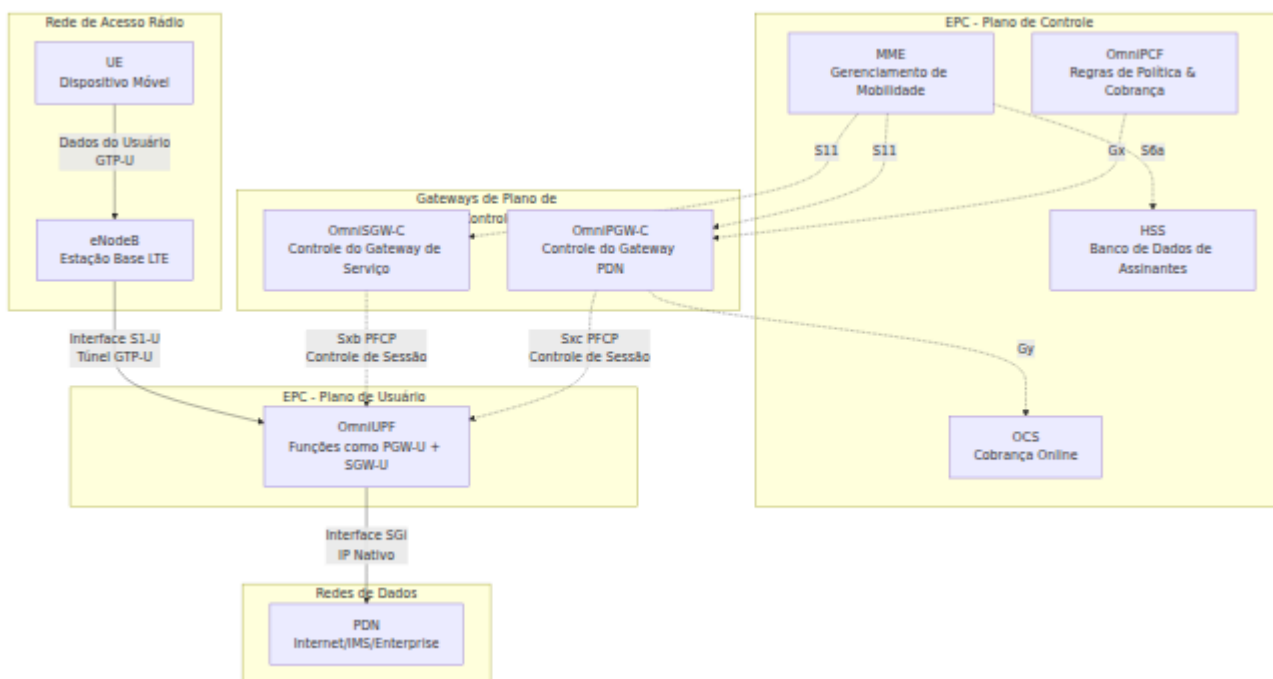
### Modo Combinado PGW-U/SGW-U (Implantação Típica de 4G)

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 Sites)

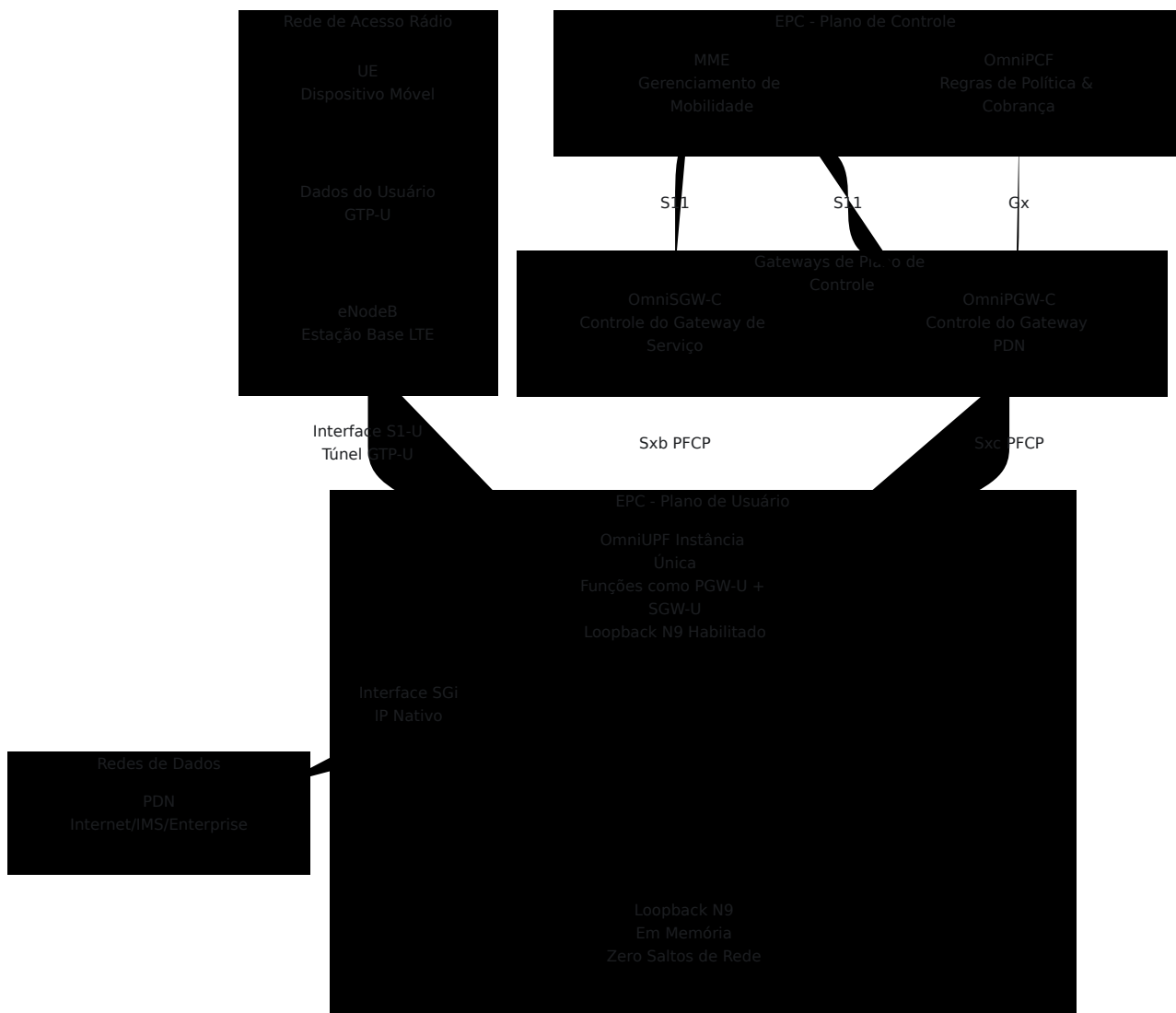
Em implantações de roaming ou múltiplos sites, duas instâncias separadas do OmniUPF podem ser implantadas - uma como SGW-U e uma como PGW-U.



## Modo de Loopback N9 (Instância Única SGWU+PGWU)

Para implantações simplificadas, o OmniUPF pode executar **tanto funções SGWU quanto PGWU em uma única instância** com processamento de

loopback N9 totalmente em eBPF.



### Principais Recursos:

- **Latência N9 sub-microsegundo** - Processado inteiramente em eBPF, nunca toca na rede
- **Redução de CPU de 40-50%** - Uma única passagem XDP vs. duas instâncias separadas
- **Implantação simplificada** - Uma instância, um arquivo de configuração
- **Detecção automática** - Quando `n3_address = n9_address`, o loopback é habilitado
- **Total conformidade com 3GPP** - Protocolos PFPC e GTP-U padrão

### Configuração:

```
# OmniUPF config.yml
interface_name: [eth0]
n3_address: "10.0.1.10"      # IP da interface S1-U
n9_address: "10.0.1.10"     # Mesmo IP habilita o loopback N9
pfcf_address: ":8805"       # Tanto SGWU-C quanto PGWU-C se
                             conectam aqui
```

### Quando usar:

- Implantações de computação de borda (minimizar latência)
- Ambientes com restrições de custo (servidor único)
- Laboratório/testes (configuração simplificada)
- Implantações pequenas a médias (< 100K assinantes)

### Quando NÃO usar:

- Redundância geográfica necessária (SGWU e PGWU em locais diferentes)
- Mandatos regulatórios para gateways separados
- Escala massiva (> 1M assinantes)

Para detalhes completos, exemplos de configuração, solução de problemas e métricas de desempenho, consulte [Guia de Operações de Loopback N9](#).

---

## Como as Funções do Plano do Usuário Funcionam na Rede

A função do 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 encapsulamento GTP-U
- **4G**: Pacotes chegam na interface S1-U (SGW-U) ou S5/S8 (PGW-U) do eNodeB com encapsulamento GTP-U
- O plano do usuário compara pacotes com PDRs de uplink com base no TEID
- O programa eBPF aplica QER (limitação de taxa, marcação)
- FAR determina a ação de encaminhamento (encaminhar, descartar, 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 aplicativo
- FAR determina o túnel GTP-U e os parâmetros de encaminhamento
- Encapsulamento GTP-U adicionado com 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 durante a troca de caminho
- Transição sem interrupções 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
<b>N4</b>	OmniSMF ↔ OmniUPF	Estabelecimento, modificação, exclusão de sessão PFCP	TS 29.244
<b>N3</b>	gNB → OmniUPF	Tráfego do plano do usuário da RAN (GTP-U)	TS 29.281
<b>N6</b>	OmniUPF → Rede de Dados	Tráfego do plano do usuário para DN (IP nativo)	TS 23.501
<b>N9</b>	OmniUPF ↔ OmniUPF	Comunicação inter-UPF para roaming/borda	TS 23.501

### Interfaces 4G/EPC

<b>Interface</b>	<b>De → Para</b>	<b>Propósito</b>	<b>Especificação 3GPP</b>
<b>Sxb</b>	OmniSGW-C ↔ OmniUPF (modo SGW-U)	Controle de sessão PFCP para gateway de serviço	TS 29.244
<b>Sxc</b>	OmniPGW-C ↔ OmniUPF (modo PGW-U)	Controle de sessão PFCP para gateway PDN	TS 29.244
<b>S1-U</b>	eNodeB → OmniUPF (modo SGW-U)	Tráfego do plano do usuário da RAN (GTP-U)	TS 29.281
<b>S5/S8</b>	OmniUPF (SGW-U) ↔ OmniUPF (PGW-U)	Plano do usuário inter-gateway (GTP-U)	TS 29.281
<b>SGi</b>	OmniUPF (modo PGW-U) → PDN	Tráfego do 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 núcleo do mecanismo de processamento de pacotes que roda no kernel do Linux para desempenho máximo.

#### Funções Principais:

- **Processamento de GTP-U:** Encapsulamento e desencapsulamento 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, 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
<code>uplink_pdr_map</code>	PDRs de uplink	TEID (32 bits)	Informações PDR (ID FAR, ID QER, IDs URR)
<code>downlink_pdr_map</code>	PDRs de downlink (IPv4)	Endereço IP do UE	Informações PDR
<code>downlink_pdr_map_ip6</code>	PDRs de downlink (IPv6)	Endereço IPv6 do UE	Informações PDR
<code>far_map</code>	Regras de encaminhamento	ID FAR	Parâmetros de encaminhamento (ação, informações do túnel)
<code>qer_map</code>	Regras de QoS	ID QER	Parâmetros de QoS (MBR, GBR, marcação)
<code>urr_map</code>	Rastreamento de uso	ID URR	Contadores de volume (uplink, downlink, total)
<code>sdf_filter_map</code>	Filtros SDF	ID PDR	Filtros de aplicativo (portas, protocolos)

### Características de Desempenho:

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

- **Multicanal:** Escala através de núcleos de CPU com suporte a mapa por CPU
- **Capacidade:** Milhões de PDRs/FARs em mapas eBPF (limitados 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:

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

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

Para detalhes da API e uso, 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 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 dataplane
- **Visualizador de Logs:** Streaming de logs ao vivo para solução de problemas

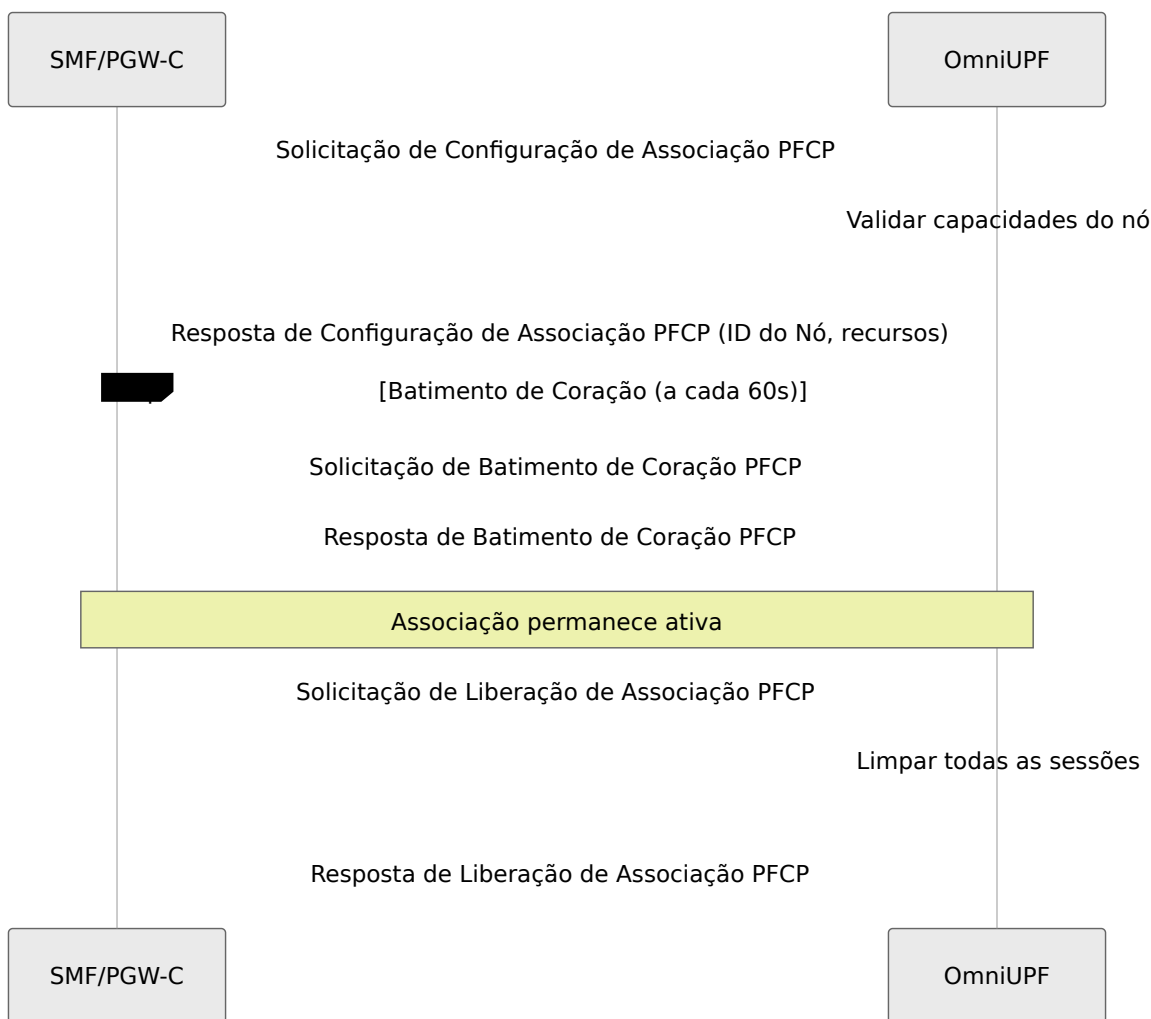
Para operações detalhadas da UI, consulte [Guia de Operações da Web UI](#).

## Protocolo PFCP e Integração com 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

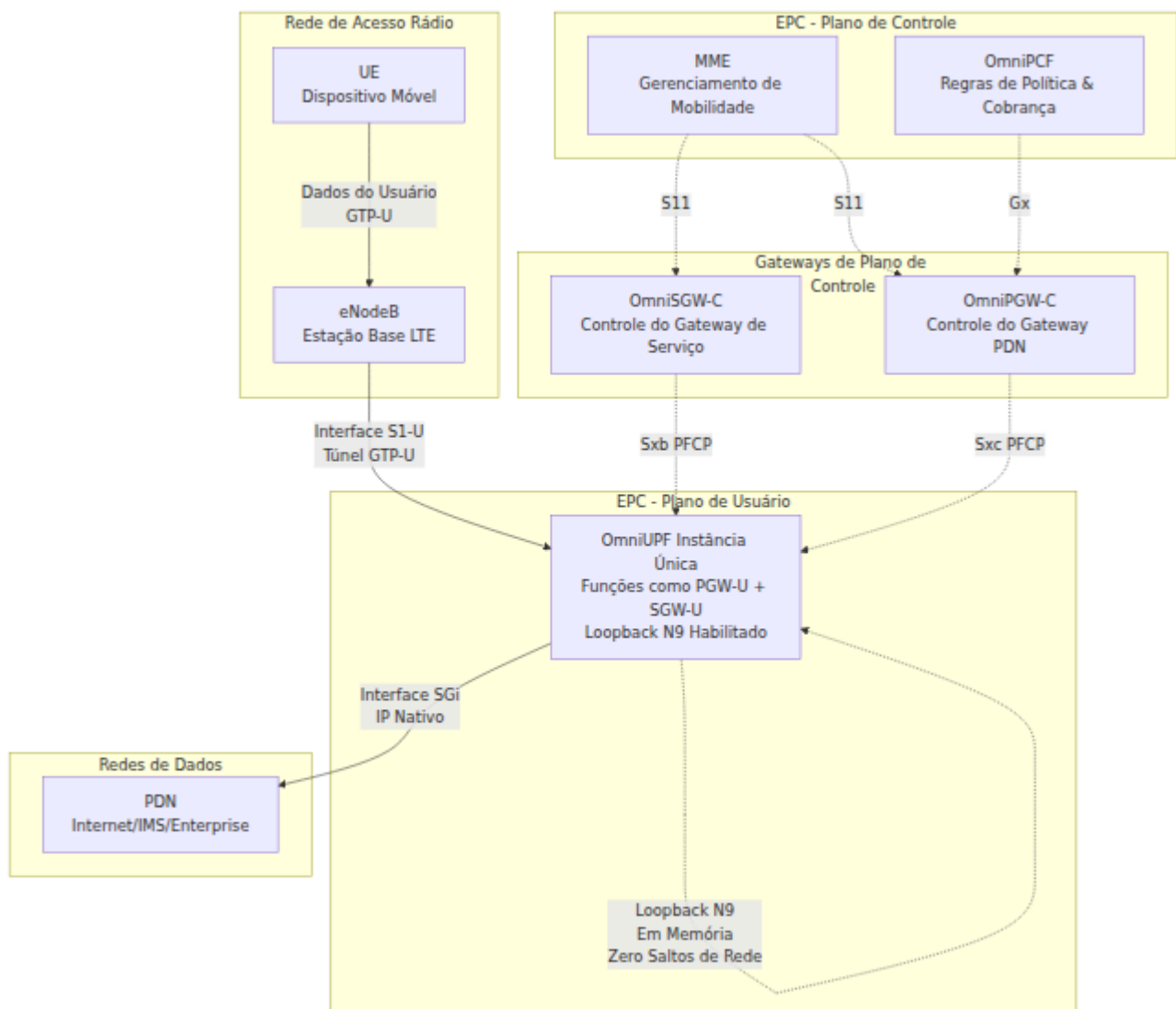
OmniUPF detecta automaticamente quando um SMF reinicia e limpa sessões órfãs de acordo com as especificações da 3GPP TS 29.244.

## Como Funciona:

Quando um SMF estabelece uma associação PFCP, ele fornece um **Timestamp de Recuperação** indicando quando começou. O 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 **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: A 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 Seção 5.22.2 da 3GPP TS 29.244:

"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 reiniciou e deverá excluir todas as sessões PFCP associadas a essa função CP."

Para solução de problemas de sessões órfãs, consulte [Detecção de Sessões Órfãs](#).

---

# Manipulação de Indicações de Erro GTP-U

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

## O que são Indicações de Erro:

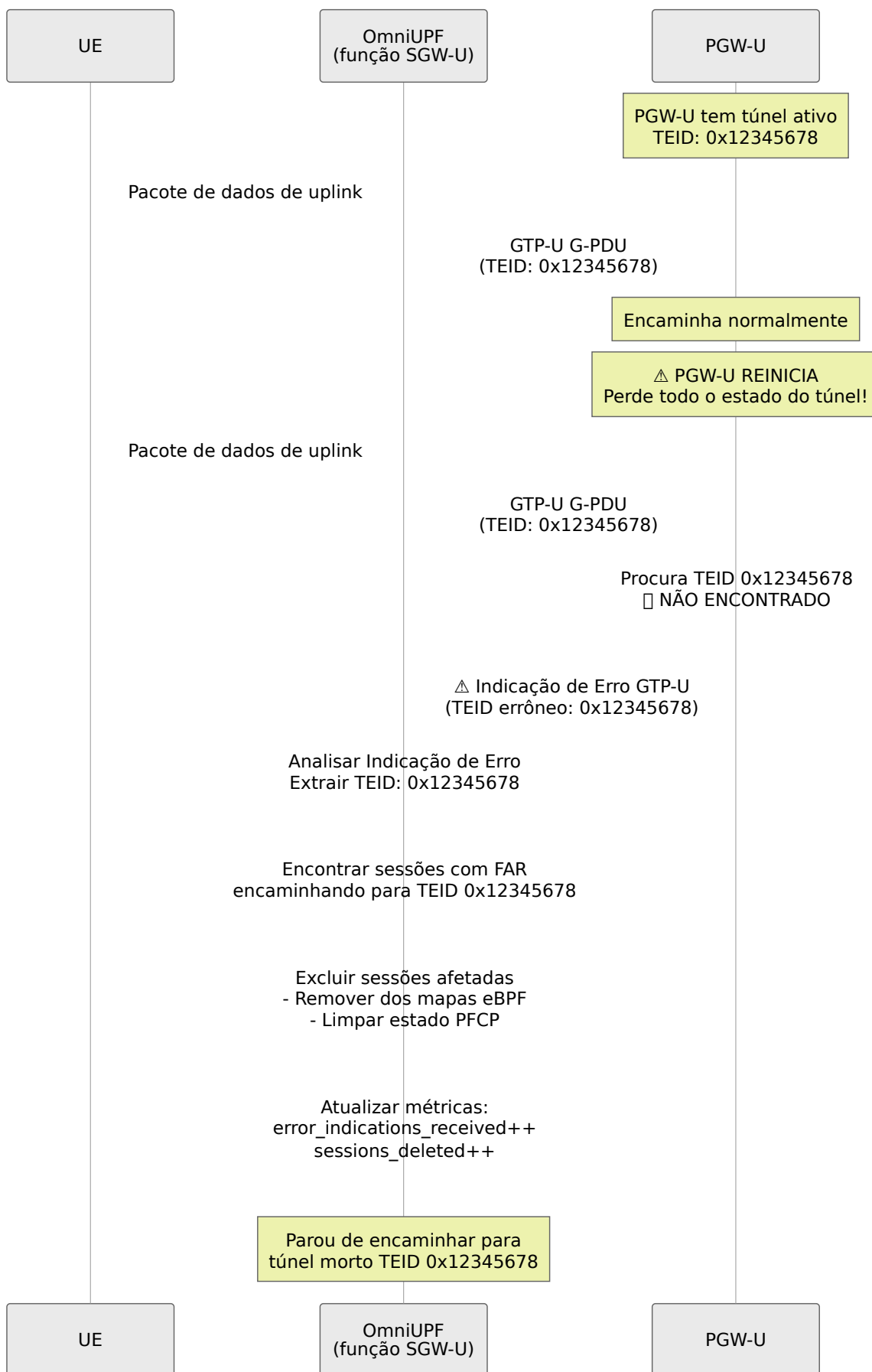
Quando o OmniUPF encaminha um pacote GTP-U para um par remoto (por exemplo, PGW-U na implantação SGW-U), o par pode enviar de volta uma **Indicação de Erro** se não reconhecer o TEID (Tunnel Endpoint Identifier). Isso indica:

- O par remoto reiniciou 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 o par remoto (porta 2152)
2. **Par remoto não reconhece TEID X** → Procura TEID na 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 mensagem para extrair TEID X
5. **UPF encontra sessões afetadas** → Busca todas as sessões por FARs que encaminham 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** (Seção 7.3.1 da 3GPP TS 29.281):

### 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: Encontrada sessão 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 do 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 Métricas:

- `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 do 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 que encaminham para o TEID errôneo exato são excluídas
3. **Isolamento por Par:** Indicações de Erro de um par afetam apenas sessões que encaminham para aquele 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 configuração de associação)
  - Manipulação de Indicação de Erro = reativa (detecta túneis mortos quando o tráfego flui)
6. **Manipulação de Pacote Malformado:** Indicações de Erro inválidas são registradas e ignoradas (nenhuma sessão excluída)

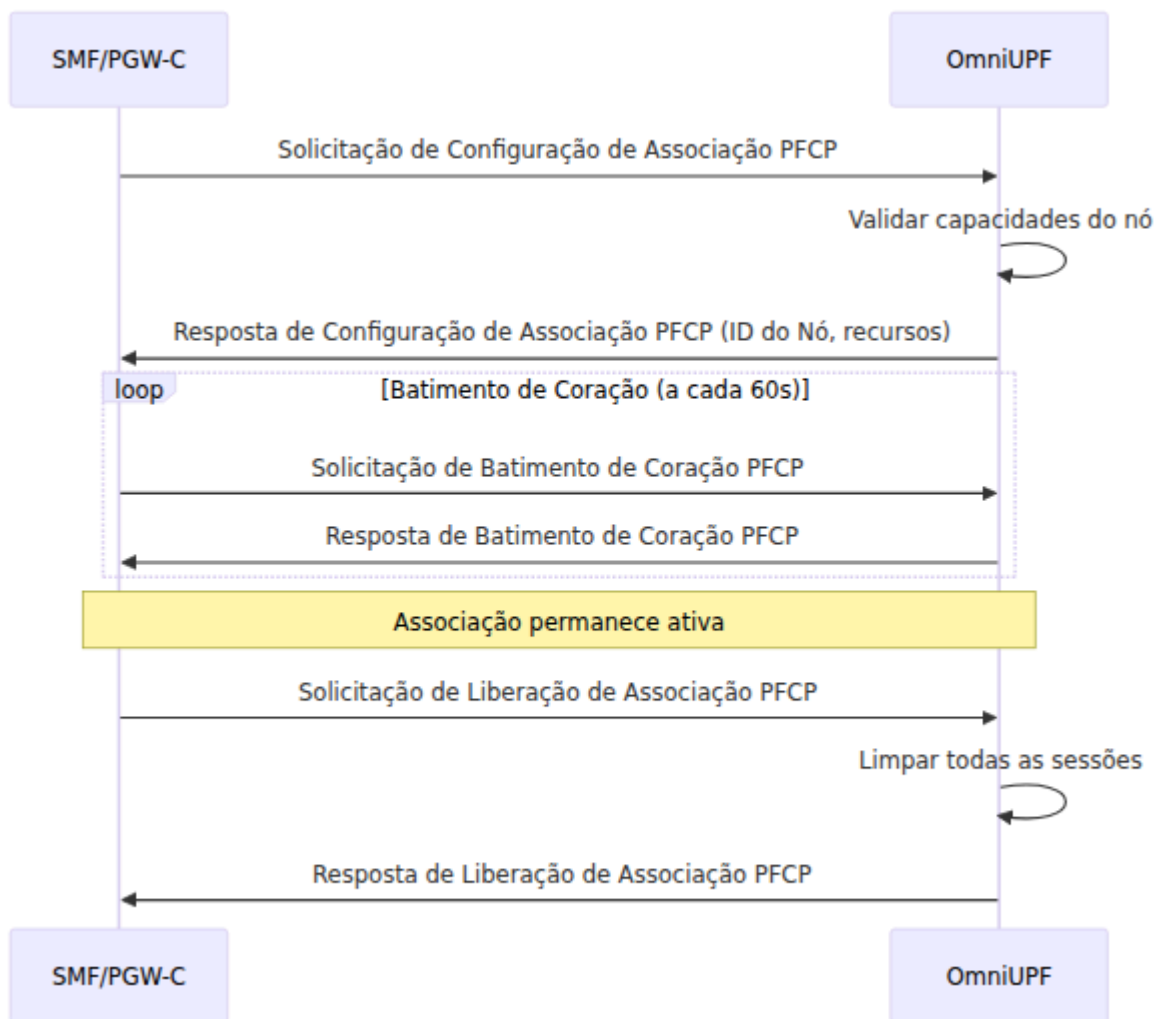
Para solução de problemas de Indicações de Erro, consulte [Depuração de Indicações de Erro GTP-U](#).

---

## 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 de Uplink:** Comparar no TEID N3, encaminhar via FAR para N6
- **PDR de Downlink:** Comparar no endereço IP do UE, encaminhar via FAR para N3 com encapsulamento 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 (transferência), mudanças de QoS ou atualizações de serviço.

### Cenários Comuns de Modificação:

## 1. Transferência (baseada em N2)

- Atualizar FAR de uplink com novo ponto final de túnel gNB (F-TEID)
- Opcionalmente armazenar pacotes durante a troca de caminho
- Limpar buffer para novo caminho quando pronto

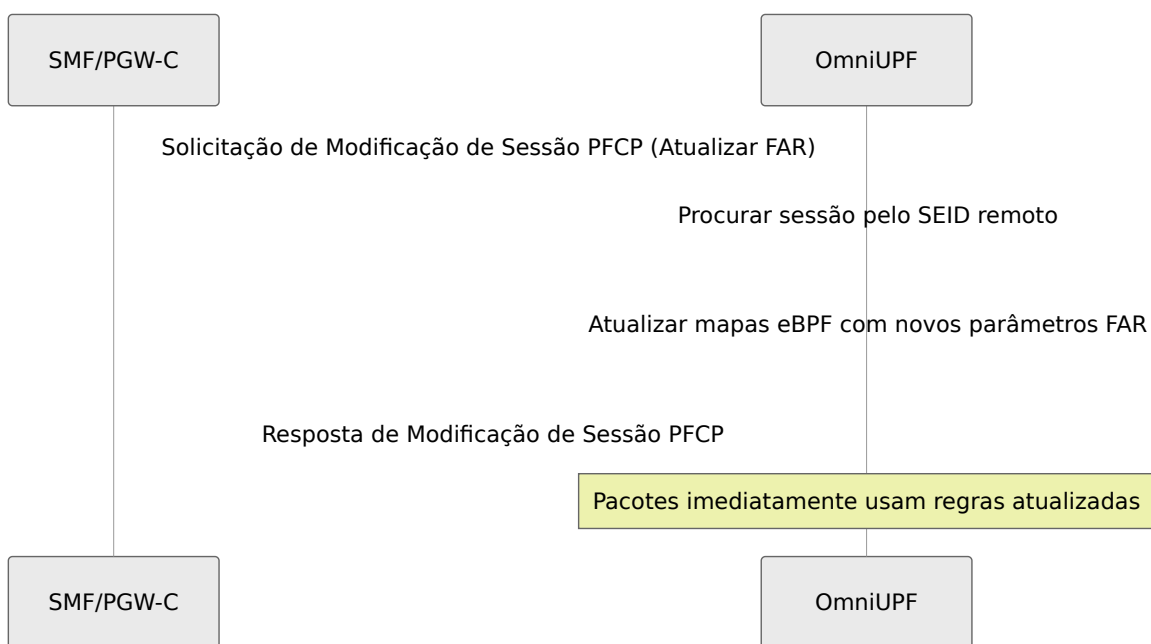
## 2. Mudança de QoS

- Atualizar QER com novos valores de MBR/GBR
- Pode adicionar/remover filtros SDF em 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:



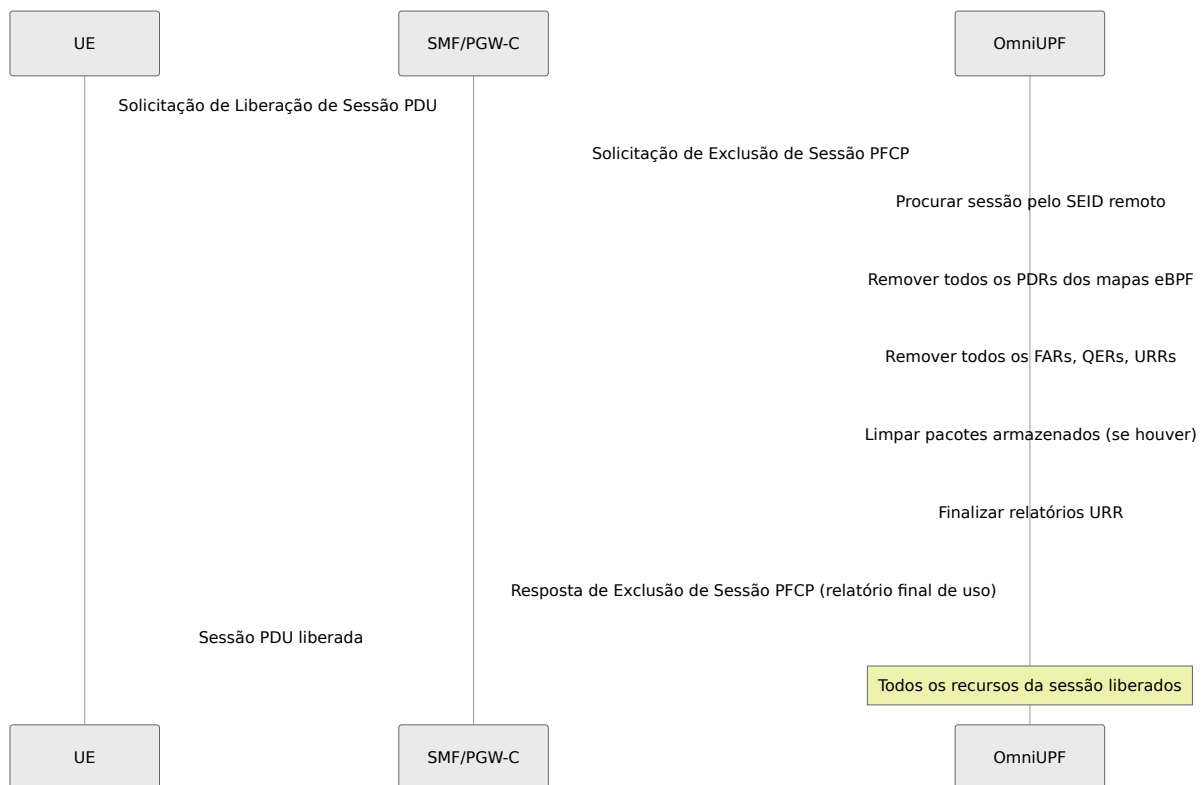
Para gerenciamento de regras, consulte [Guia de Gerenciamento de Regras](#).

---

## 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 final de uso 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 suas significâncias.

## Monitoramento de Sessões

### 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 aplicação de QoS (opcional)
- URRs para rastreamento de uso (opcional)

### **Operações Chave da 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):**

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

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

### **Regras de Ação de Encaminhamento (FAR):**

As 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 armazenamento** por FAR
- **Alternar armazenamento** para FARs específicas durante solução de problemas

### **Regras de Aplicação de QoS (QER):**

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

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

### **Regras de Relatório de Uso (URR):**

As 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](#).

---

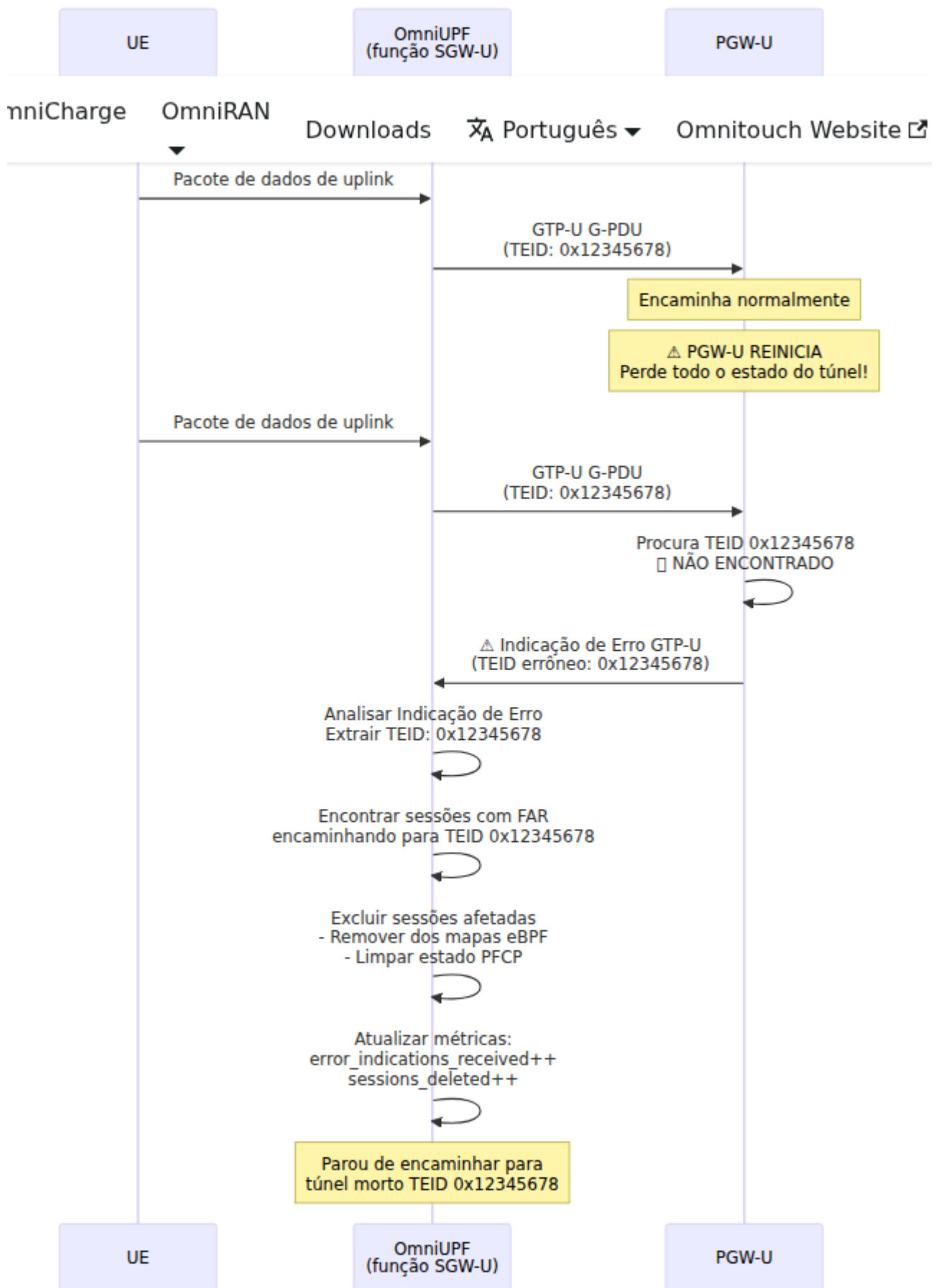
## **Buffering de Pacotes**

### **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, os usuários móveis experimentaríamos desconexões, 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 (transferência), 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:



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

- **Conexões TCP a travar** ou reiniciar (navegação na web, downloads interrompidos)
- **Chamadas de vídeo a congelar** ou cair (Zoom, Teams, chamadas do WhatsApp falham)
- **Sessões de jogos a desconectar** (jogos online, aplicativos em tempo real falham)
- **Chamadas VoIP a ter lacunas** ou cair completamente (chamadas telefônicas interrompidas)
- **Downloads a falhar** e precisar reiniciar

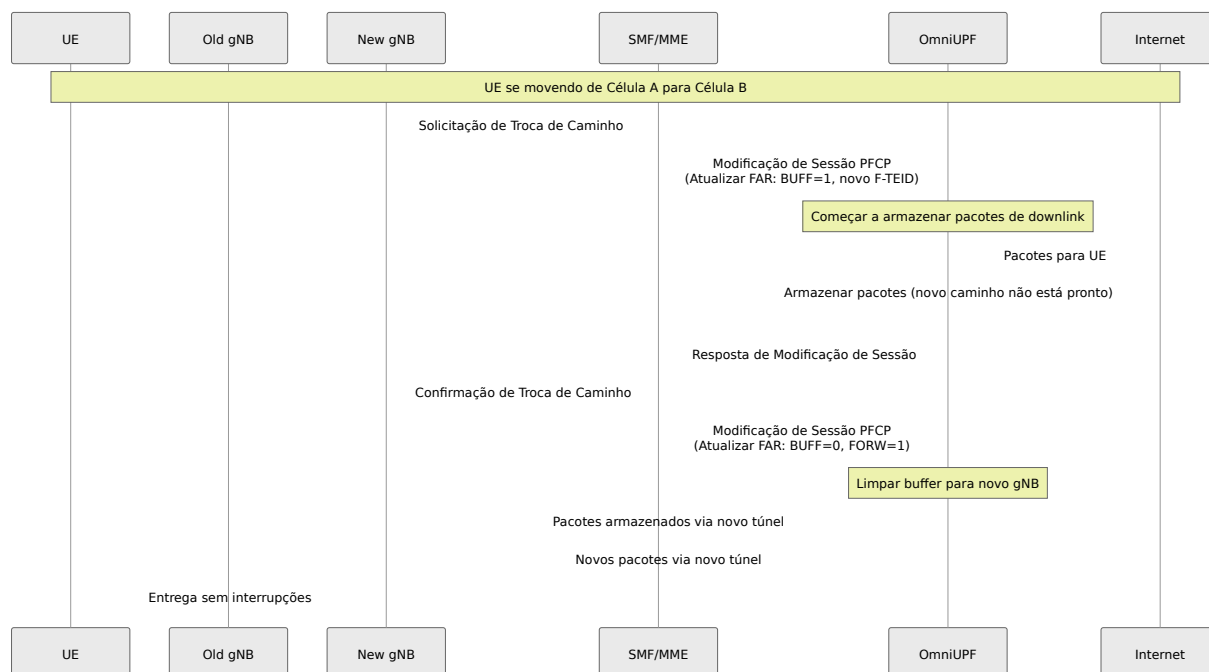
**Com buffering:** OmniUPF retém temporariamente pacotes até que o novo caminho seja estabelecido e, em seguida, os encaminha sem problemas. O usuário experimenta **zero interrupção**.

## Quando o Buffering Acontece

OmniUPF armazena pacotes nessas situações críticas:

### 1. Transferência Baseada em N2 (5G) / Transferência Baseada 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 armazenar (caminho antigo fechando, novo caminho não está pronto)
- **T+10-50ms: Janela crítica de armazenamento** - pacotes chegam, mas não podem ser encaminhados
- **T+50ms:** Novo caminho pronto, SMF diz ao UPF para encaminhar
- **T+50ms+:** UPF limpa pacotes armazenados para o novo caminho, em seguida, encaminha novos pacotes normalmente

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

---

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

Quando a rede precisa mudar parâmetros de sessão:

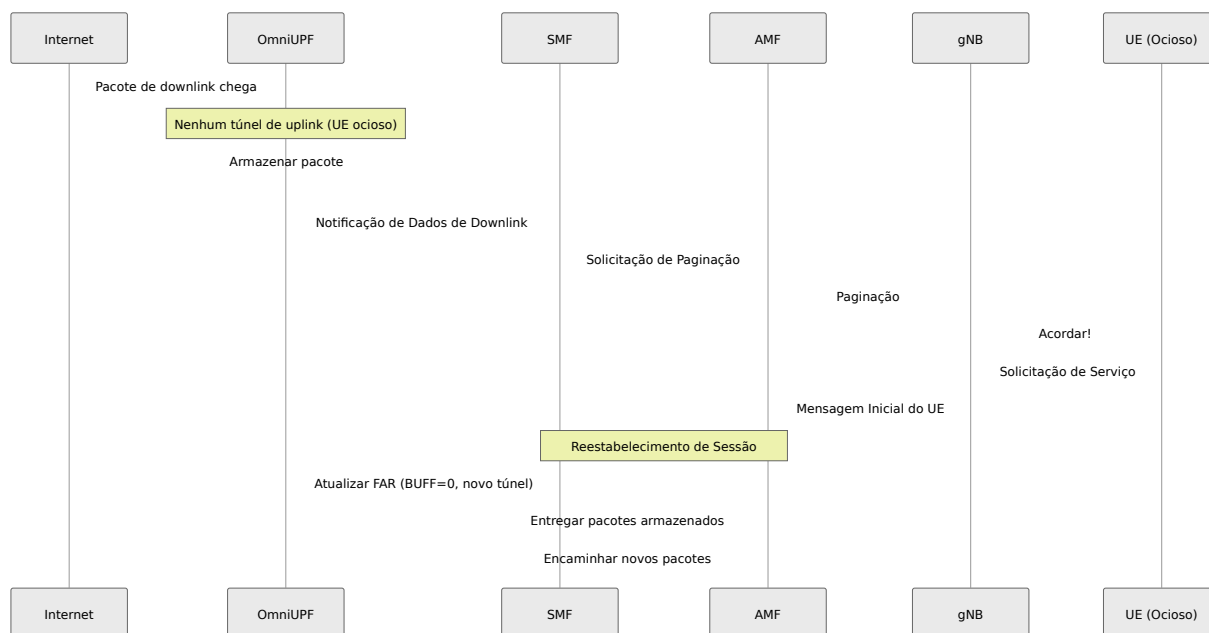
- **Atualização/diminuição de QoS:** Usuário se move de cobertura 4G para 5G (modo NSA)
- **Mudança de política:** Usuário corporativo entra no campus corporativo (mudanças de direcionamento de tráfego)
- **Otimização de rede:** Rede central redireciona tráfego para um UPF mais próximo (atualização ULCL)

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

---

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

Quando um UE está em modo ocioso (tela desligada, economia de bateria) e dados de 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 reconecta.

#### 4. Transferência Inter-RAT (4G ↔ 5G)

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

- Mudanças de arquitetura (eNodeB ↔ gNB)
- Mudanças nos pontos finais do túnel (nova alocação de TEID)
- O buffering garante uma transição suave entre tipos de RAT

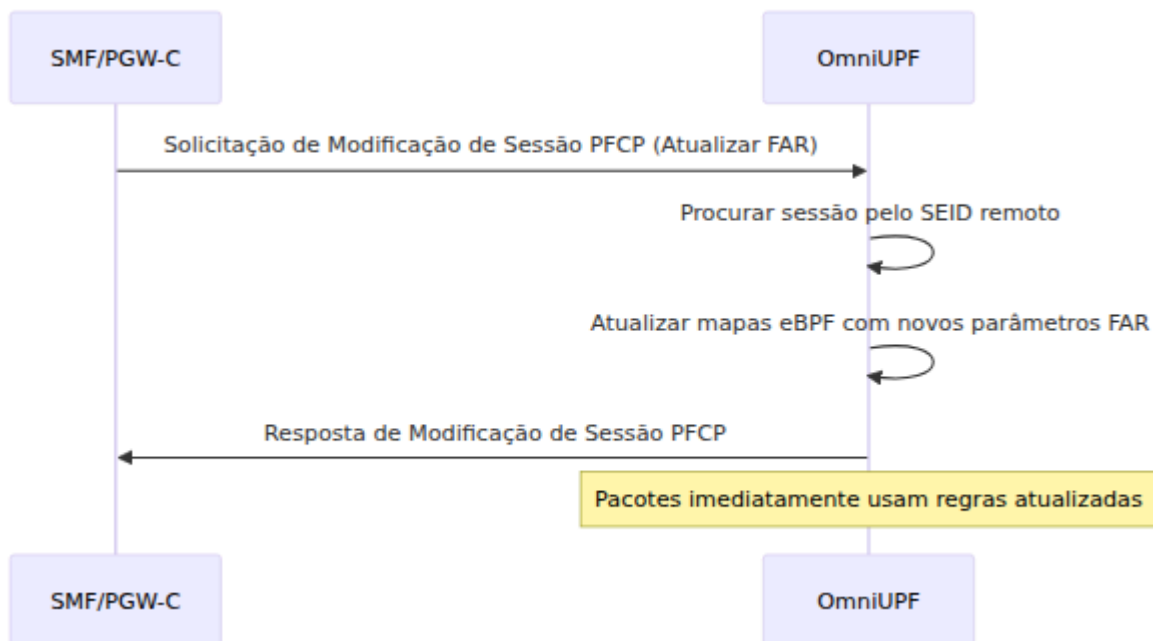
#### Como o Buffering Funciona no OmniUPF

##### 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 armazenados na memória

##### Processo de Buffering:



### Detalhes Chave:

- **Porta de Buffer:** Porta UDP 22152 (pacotes enviados do eBPF para o espaço do usuário)
- **Encapsulamento:** Pacotes envoltos em GTP-U com ID FAR como TEID
- **Armazenamento:** Buffers em 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 velhos que 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 Armazenados:** eBPF detecta flag BUFF, encapsula pacotes, envia para a porta 22152
3. **Armazenamento do 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:** Gerenciador de espaço do usuário repete pacotes armazenados usando novas regras FAR (novo ponto final do 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
<b>Chamada de Vídeo Durante a Transferência</b>	Chamada congela por 1-2 segundos, pode cair	Sem interrupções, nenhuma interrupção
<b>Download de Arquivo na Borda da Célula</b>	Download falha, deve reiniciar	Download continua sem interrupções
<b>Jogo Online Enquanto se Move</b>	Conexão cai, expulso do jogo	Jogo suave, sem desconexões
<b>Chamada VoIP no Carro</b>	Chamada cai a cada transferência	Cristalina, sem quedas
<b>Vídeo em Streaming em um Trem</b>	Vídeo carrega, qualidade cai	Reprodução suave
<b>Hotspot Móvel para Laptop</b>	Sessão SSH cai, chamada de vídeo falha	Todas as conexões mantidas

### Benefícios para o Operador de Rede:

- **Redução da Taxa de Queda de Chamadas (CDR):** KPI crítico para qualidade da rede

- **Maior Satisfação do Cliente:** Usuários não notam transferências
  - **Menores Custos de Suporte:** Menos reclamações sobre conexões perdidas
  - **Vantagem Competitiva:** Marketing "melhor rede para cobertura"
- 

## Operações de Gerenciamento de Buffer

Os operadores podem monitorar e controlar o buffering via Web UI e API:

### Monitoramento:

- **Visualizar pacotes armazenados** por ID FAR (contagem, bytes, idade)
- **Rastrear uso do buffer** contra limites (por FAR, global)
- **Alertar sobre overflow de buffer** ou duração excessiva de armazenamento
- **Identificar buffers presos** (pacotes armazenados > limite TTL)

### Operações de Controle:

- **Limpar buffers:** Acionar manualmente a repetição do buffer (solução de problemas)
- **Limpar buffers:** Descartar pacotes armazenados (limpar buffers presos)
- **Ajustar TTL:** Mudar tempo de expiração de pacotes
- **Modificar limites:** Aumentar capacidade de buffer por FAR ou global

### Solução de Problemas:

- **Buffer não limpando:** Verifique se o SMF enviou atualização de FAR para desabilitar o buffering
- **Overflow de buffer:** Aumente limites ou investigue por que a duração do buffering é excessiva
- **Pacotes antigos no buffer:** TTL pode estar muito alto, ou atualização de FAR atrasada
- **Buffering excessivo:** Pode indicar problemas de mobilidade ou problemas no SMF

Para operações detalhadas de buffer, consulte [Guia de Gerenciamento de Buffer](#).

---

## Configuração de Buffer

Configure o comportamento de buffering em `config.yml`:

```
# Configurações de Buffer
buffer_port: 22152                # Porta UDP para pacotes
armazenados (padrão)
buffer_max_packets: 10000        # Máx. pacotes por FAR (prevenir
exaustão de memória)
buffer_max_total: 100000        # Máx. total de pacotes em todos
os FARs
buffer_packet_ttl: 30            # TTL em segundos (descartar
pacotes antigos)
buffer_cleanup_interval: 60     # Intervalo de limpeza em
segundos
```

### 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 completas de configuração, consulte [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:** Contagens 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 Erros:** TEIDs inválidos, IPs de UE desconhecidos

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

Para monitoramento detalhado, consulte [Guia de Monitoramento](#).

---

## **Gerenciamento de Capacidade**

### **Monitoramento de Capacidade de Mapas eBPF:**

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

- **Monitorar uso de mapas** 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 de downlink IPv4
- `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, consulte [Gerenciamento de Capacidade](#).

---

## Gerenciamento de Configuração

### Configuração do UPF:

Visualizar e verificar parâmetros operacionais do UPF:

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

### Configuração do Dataplane:

Parâmetros ativos do caminho de dados eBPF:

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

Para visualização de configuração, consulte [Visualização de Configuração](#).

## Solução de Problemas

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

### Falhas no Estabelecimento de Sessões

**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 Exaurida

- Verifique a visualização de Capacidade para uso de mapa vermelho (>90%)
- Aumente os tamanhos dos mapas eBPF na configuração do UPF
- Exclua sessões obsoletas se o mapa estiver cheio

### 3. Configuração de PDR/FAR Inválida

- 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 FAR referencia instâncias de rede válidas

### 4. Problemas de Configuração de Interface

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

Para solução de problemas detalhada, consulte [Guia de Solução de Problemas](#).

---

## Problemas de Perda de Pacotes ou Encaminhamento

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

### Causas Raiz Comuns:

#### 1. Configuração de PDR Incorreta

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

#### 2. Problemas de Ação FAR

- Verifique se a ação FAR é ENCAMINHAR (não DESCARTAR ou BUFFER)

- Verifique os parâmetros de criação de cabeçalho externo para GTP-U
- Certifique-se de que o ponto final de destino está correto

### **3. Limites de QoS Excedidos**

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

### **4. Problemas de MTU da Interface**

- Verifique a sobrecarga do GTP-U (40-50 bytes) não causa fragmentação
- Verifique a configuração de MTU da interface N3/N6
- Monitore mensagens ICMP