Guia de Operações do OmniUPF

Índice

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 qualidade de operadora, aplicação de QoS e gerenciamento de tráfego para redes móveis. Construído sobre a tecnologia eBPF do Linux (Filtro de Pacote de Berkeley estendido) e aprimorado com capacidades de gerenciamento abrangentes, o OmniUPF fornece a infraestrutura central de processamento de pacotes necessária para redes 5G SA, 5G NSA e LTE.

O que é uma Função do Plano do Usuário?

A Função do Plano do Usuário (UPF) é o elemento de rede padronizado pelo 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 nas especificações 3GPP TS 23.501 (5G) e TS 23.401 (LTE), fornecendo uma solução completa e pronta para produção para o 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
- Encapsulação e desencapsulação de GTP-U (Protocolo de Tunelamento GPRS)
- Suporte a IPv4 e IPv6 para redes de acesso e de dados
- XDP (Caminho de Dados eXpress) 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 (Protocolo de Controle de Encaminhamento de Pacotes) para SMF/PGW-C
- API RESTful para monitoramento e diagnósticos
- Estatísticas e métricas em tempo real
- Monitoramento de capacidade de mapas eBPF
- Painel de controle baseado na web

Características de Desempenho:

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

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

Entendendo a Arquitetura do Plano do Usuário

OmniUPF é uma solução unificada do plano do usuário que fornece encaminhamento de pacotes de qualidade de operadora para redes 5G Autônomas (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 (Gateway de PDN do Plano do Usuário)** - gateway EPC 4G para redes externas (controlado pelo OmniPGW-C via Sxc/PFCP)
- **SGW-U (Gateway de Serviço do Plano do Usuário)** - 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, proporcionando desempenho alto e consistente, seja operando como UPF, PGW-U, SGW-U, ou 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 4G)

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

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

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

Modo de Loopback N9 (Instância Única SGWU+PGWU)

Para implantações simplificadas, o OmniUPF pode executar **ambos os papéis SGWU e 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 a 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 PFCP e GTP-U padrão

Quando usar:

- Implantações de computação de borda (minimizar latência)
- Ambientes com restrições de custo (servidor único)
- Lab/testes (configuração simplificada)
- Implanta��ões 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

- Mapas eBPF são preenchidos com regras de encaminhamento

2. **Processamento de Pacotes de Uplink** (UE → Rede de Dados)
    - **5G**: Pacotes chegam na interface N3 do gNB com encapsulação GTP-U
    - **4G**: Pacotes chegam na interface S1-U (SGW-U) ou S5/S8 (PGW-U) do eNodeB com encapsulação 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
    - Encapsulação GTP-U adicionada 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 bufferizar pacotes durante a troca de caminho
    - Transi��ão sem costura 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 3GPP:

Interfaces 5G

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

Interfaces 4G/EPC

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

Para gerenciamento de sessões PFCP, consulte Operações PFCP.

Componentes do UPF

Caminho de Dados eBPF

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

Funções Principais:

- **Processamento GTP-U**: Encapsulação e desencapsulação de túneis GTP-U

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

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

Nome do Mapa	Propósito	Chave	Valor
`uplink_pdr_map`	PDRs de uplink	TEID (32 bits)	Informações PDR (ID FAR, ID QER, IDs URR)
`downlink_pdr_map`	PDRs de downlink (IPv4)	Endereço IP do UE	Informações PDR
`downlink_pdr_map_ip6`	PDRs de downlink (IPv6)	Endereço IPv6 do UE	Informações PDR
`far_map`	Regras de encaminhamento	ID FAR	Parâmetros de encaminhamento (ação, informações do túnel)
`qer_map`	Regras de QoS	ID QER	Parâmetros de QoS (MBR, GBR, marcação)
`urr_map`	Rastreamento de uso	ID URR	Contadores de volume (uplink, downlink, total)
`sdf_filter_map`	Filtros SDF	ID PDR	Filtros de aplicativos (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
- **Multicore**: Escala entre núcleos de CPU com suporte a mapas por CPU
- **Capacidade**: Milhões de PDRs/FARs em mapas eBPF (limitado pela memória do kernel)

Para monitoramento de capacidade, consulte Gerenciamento de Capacidade.

Manipulador de Interface PFCP

A **interface PFCP** implementa a TS 29.244 do 3GPP 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
- **Relatórios de Eventos**: Notificar SMF sobre limites de uso, erros ou eventos de sessão

Mensagens PFCP Suportadas:

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

Elementos de Informação (IE) Suportados:

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

Para operações PFCP detalhadas, consulte **Guia de Operações PFCP**.

Servidor API REST

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

Funções Principais:

- **Monitoramento de Sessões**: Consultar sessões PFCP ativas e associações
- **Inspeção de Regras**: Visualizar configurações de PDR, FAR, QER, URR

- **Estatísticas**: Recuperar contadores de pacotes, estatísticas de rotas, estatísticas de XDP
- **Gerenciamento de Buffers**: Visualizar e controlar buffers de pacotes
- **Informações de Mapas**: Monitorar uso e capacidade de mapas eBPF

Endpoints da API: (34 endpoints no total)

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

Para detalhes e uso da API, consulte **Guia de Operações PFCP** e **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:

- **Visão 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 bufferizados e controlar o buffering por FAR
- **Painel de Estatísticas**: Estatísticas em tempo real de pacotes, rotas, XDP e estatísticas das interfaces N3/N6
- **Monitoramento de Capacidade**: Uso de mapas eBPF com indicadores de capacidade codificados por cores
- **Visão 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 interface, consulte **Guia de Operações da Interface Web**.

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 Visão de Sessões.

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 encapsulação GTP-U
- **FAR**: Parâmetros de encaminhamento (criação de cabeçalho externo,

instância de rede)
- **QER**: Limites de QoS (MBR, GBR) e marcação de pacotes (QFI)
- **URR**: Relatório de volume para cobrança (opcional)

Para monitoramento de sessões, consulte Operações PFCP.

Modificação de Sessão PFCP

O SMF pode modificar sessões para eventos de mobilidade (transferência), alterações 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 bufferizar pacotes durante a troca de caminho
    - Limpar buffer para novo caminho quando estiver pronto
2. **Mudança de QoS**
    - Atualizar QER com novos valores MBR/GBR
    - Pode adicionar/remover filtros SDF no PDR para QoS específica de aplicativo
3. **Atualização de Serviço**
    - Adicionar novos PDRs para fluxos de tráfego adicionais
    - Modificar FARs para alterações 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 de UE (5G) ou contextos PDP (LTE). Cada sessão contém:

- SEIDs locais e remotos (Identificadores de Ponto de 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 de Sessão:

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

Para procedimentos detalhados de sessão, consulte Visão de Sessões.

Gerenciamento de Regras

Regras de Detecção de Pacotes (PDR):

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

- **Visualizar PDRs de uplink** com chave TEID da interface N3
- **Visualizar PDRs de downlink** com chave de 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):

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

- **Visualizar ações FAR** (ENCAMINHAR, DESCARTAR, BUFFER, DUPLICAR, NOTIFICAR)
- **Inspecionar parâmetros de encaminhamento** (criação de cabeçalho externo, destino)
- **Monitorar status de buffering** por FAR
- **Alternar buffering** para FARs específicas durante solução de problemas

Regras de Aplicação de QoS (QER):

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

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

Regras de Relatório de Uso (URR):

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

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

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

Bufferização de Pacotes

Por que a Bufferização é Crítica para o UPF

A bufferização 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 bufferização, os usuários móveis experimentariam 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 Mobilidade

Em redes móveis, os usuários estão constantemente se movendo. Quando um dispositivo se move de uma torre 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 bufferização: Pacotes que chegam durante esta janela crítica seriam **descartados**, causando:

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

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

Quando a Bufferização Acontece

OmniUPF bufferiza pacotes nestes cenários críticos:

1. Transferência Baseada em N2 (5G) / Transferência Baseada em X2 (4G)

Quando um UE se move entre torres celulares:

Linha do Tempo:

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

Sem bufferização: ~40ms de pacotes (potencialmente milhares) seriam **perdidos**. **Com bufferização**: 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**: A rede central redireciona tráfego para um UPF mais

próximo (atualização ULCL)

Durante a modificação, o plano de controle pode precisar atualizar várias regras de forma atômica. A bufferização 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 bufferização: O pacote inicial que acionou a notificação seria **perdido**, exigindo que o remetente retransmitisse (adiciona latência). **Com bufferização**: O pacote que acordou o UE é entregue imediatamente quando o UE se 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)
- A bufferização garante uma transição suave entre tipos de RAT

Como a Bufferização Funciona no OmniUPF

Mecanismo Técnico:

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

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

Processo de Bufferização:

Detalhes Chave:

- **Porta de Buffer**: Porta UDP 22152 (pacotes enviados de eBPF para espaço do usuário)
- **Encapsulação**: Pacotes embrulhados 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 antigos que o TTL são descartados
- **Limpeza**: Processo em segundo plano remove pacotes expirados a cada 60 segundos

Ciclo de Vida do Buffer:

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

---

Por que Isso Importa para a Experiência do Usuário

Impacto no Mundo Real:

Cenário	Sem Bufferização	Com Bufferização
Chamada de Vídeo Durante Transferência	Chamada congela por 1-2 segundos, pode cair	Sem interrupção, sem problemas
Download de Arquivo na Borda da Célula	Download falha, precisa reiniciar	Download continua sem interrupções
Jogo Online Enquanto se Move	Conexão cai, expulso do jogo	Jogo suave, sem desconexões
Chamada VoIP no Carro	Chamada cai a cada transferência	Cristalina, sem quedas
Streaming de Vídeo em um Trem	Vídeo bufferiza, qualidade cai	Reprodução suave
Hotspot Móvel para Laptop	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 percebem transferências
- **Menores Custos de Suporte**: Menos reclamações sobre conexões caídas
- **Vantagem Competitiva**: Marketing "melhor rede para cobertura"

Operações de Gerenciamento de Buffer

Os operadores podem monitorar e controlar a bufferização via a Interface Web e API:

Monitoramento:

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

Operações de Controle:

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

Solução de Problemas:

- **Buffer não limpando**: Verifique se o SMF enviou atualização de FAR para desabilitar a bufferização
- **Overflow de buffer**: Aumente os limites ou investigue por que a duração da bufferização é excessiva
- **Pacotes antigos no buffer**: TTL pode estar muito alto, ou atualização de FAR atrasada
- **Bufferização excessiva**: 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 bufferização em `config.yml`:

```
# Configurações de buffer
buffer_port: 22152                # Porta UDP para pacotes bufferizados (padrão)
buffer_max_packets: 10000         # Máx. pacotes por FAR (prevenir exaustão de memória)
buffer_max_total: 100000          # Máx. pacotes totais 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** (autoestradas, trens): Aumentar `buffer_max_packets` para 20.000+
- **Áreas urbanas densas** (transferências frequentes): Diminuir `buffer_packet_ttl` para 15s
- **Aplicativos de baixa latência**: Definir `buffer_packet_ttl` para 10s para evitar dados obsoletos
- **Redes IoT**: Diminuir limites (dispositivos IoT geram menos tráfego durante transferências)

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 para/de RAN (gNB/eNodeB)

- **N6 RX/TX**: Tráfego para/de rede de dados
- **Contagens totais de pacotes**: Estatísticas agregadas da interface

Para detalhes de monitoramento, 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 para 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:

Visualize e verifique parâmetros operacionais do UPF:

- **Interface N3**: Endereço IP para conectividade com RAN (GTP-U)
- **Interface N6**: Endereço IP para conectividade com a rede de dados

- **Interface N9**: Endereço IP para comunicação inter-UPF (opcional)
- **Interface PFCP**: Endereço IP para conectividade com SMF
- **Porta 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 Visão 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 serem criadas, 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 visão de Sessões
    - Verifique se a configuração do ID do Nó corresponde entre SMF e UPF
2. **Capacidade do Mapa eBPF Exaurida**
    - Verifique a visão 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 o FAR faz referência a 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 pelo firewall

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

Problemas de Perda ou Encaminhamento de Pacotes

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 PDR de uplink TEID 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 as configurações de QER MBR (Taxa Máxima de Bits)
   - Verifique a alocação de GBR (Taxa Garantida de Bits)
   - Monitore perdas de pacotes devido a limitação de taxa
4. **Problemas de MTU da Interface**
   - Verifique se a sobrecarga do GTP-U (40-50 bytes) não causa fragmentação
   - Verifique a configuração de MTU das interfaces N3/N6
   - Monitore mensagens ICMP de fragmentação necessárias

Problemas Relacionados ao Buffer

Sintomas: Pacotes bufferizados indefinidamente, overflow de buffer

Causas Raiz Comuns:

1. **Bufferização Não Desabilitada Após Transferência**
    - Verifique a flag de bufferização FAR (bit 2)
    - Verifique se o SMF enviou Modificação de Sessão para desabilitar a bufferização
    - Desabilite manualmente a bufferização via painel de controle se travada
2. **Expiração do TTL do Buffer**
    - Verifique a idade do pacote na visão de buffer
    - Verifique a configuração de TTL do buffer (padrão pode estar muito longo)
    - Limpe buffers expirados manualmente
3. **Capacidade do Buffer Exaurida**
    - Monitore o uso total do buffer e os limites por FAR
    - Verifique regras mal configuradas que causam bufferização excessiva
    - Ajuste os limites de max_per_far e max_total do buffer

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

Anomalias nas Estatísticas

Sintomas: Contadores de pacotes inesperados, estatísticas ausentes

Causas Raiz Comuns:

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

Degradação de Desempenho

Sintomas: Alta latência, baixa largura de banda, saturação da CPU

Diagnóstico:

1. **Monitore Estatísticas de XDP**: Verifique se há descartes ou abortos de XDP
2. **Verifique o Tempo de Acesso ao Mapa eBPF**: Consultas hash devem ser sub-microsegundo
3. **Revise a Utilização da CPU**: eBPF deve distribuir entre núcleos
4. **Analise a Interface de Rede**: Verifique se o NIC suporta descarregamento XDP

Considerações de Escalabilidade:

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

Para ajuste de desempenho, consulte Guia de Arquitetura.

Documentação Adicional

Guias de Operações Específicos de Componentes

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

Guia de Configuração

Referência completa de configuração incluindo:

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

Guia de Modos XDP

Configuração e otimização detalhadas do XDP incluindo:

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

Guia de Arquitetura

Aprofundamento técnico incluindo:

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

Guia de Gerenciamento de Regras

Referência de regras PFCP incluindo:

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

Guia de Monitoramento

Estatísticas e gerenciamento de capacidade incluindo:

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

Guia de Operações da Interface Web

Uso do painel de controle incluindo:

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

Documentação da API

Referência completa da API REST incluindo:

- Documentação interativa OpenAPI/Swagger
- Endpoints da API de sessões e associações PFCP
- Regras de Detecção de Pacotes (PDR) - IPv4 e IPv6
- Regras de Ação de Encaminhamento (FAR)
- Regras de Aplicação de QoS (QER)
- Regras de Relatório de Uso (URR)
- Gerenciamento de buffer de pacotes
- Estatísticas e endpoints de monitoramento
- Gerenciamento de rotas e integração FRR
- Informações sobre mapas eBPF
- Gerenciamento de configuração
- Diretrizes de autenticação e segurança
- Fluxos de trabalho e exemplos comuns da API

Guia de Gerenciamento de Rotas do UE

Integração de roteamento FRR incluindo:

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

Guia de Solução de Problemas

Diagnóstico abrangente de problemas incluindo:

- Lista de verificação e ferramentas de diagnóstico rápidas
- Problemas de instalação e configuração
- Falhas de associação PFCP
- Problemas de processamento de pacotes
- Erros de XDP e eBPF
- Degradação de desempenho
- Problemas específicos de hipervisores (Proxmox, VMware, VirtualBox)
- Problemas de NIC e driver
- Procedimentos de resolução passo a passo

Documentação por Caso de Uso

Instalando e Configurando o OmniUPF

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

Implantando SGWU+PGWU em Uma Única Instância (Loopback N9)

1. Guia de Operações de Loopback N9 - Guia completo para implantação combinada SGWU+PGWU
2. Loopback N9 - Configuração - Configuração de rede e PFCP
3. Loopback N9 - Monitoramento - Verifique se o loopback está ativo
4. Loopback N9 - Solução de Problemas - Problemas comuns e soluções

Implantando no Proxmox

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

Otimizando o Desempenho

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

Entendendo o Processamento de Pacotes

1. Guia de Arquitetura - Pipeline de Processamento de Pacotes

2. Guia de Gerenciamento de Regras
3. Guia de Monitoramento - Estatísticas

Planejando Capacidade

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

Gerenciando Rotas do UE e Integração FRR

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

Usando a API REST

1. [Documentação da API](./

Guia de Arquitetura do OmniUPF

Índice

Visão Geral

OmniUPF aproveita o eBPF (Extended Berkeley Packet Filter) e o XDP (eXpress Data Path) para alcançar desempenho de nível carrier 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 em microsegundos.

Camadas da Arquitetura

Princípios de Design Chave

Processamento Zero-Copy:

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

Estruturas de Dados Sem Bloqueio:

- Mapas eBPF usam tabelas hash por CPU
- Operações atômicas para acesso concorrente
- Sem sobrecarga de mutex/spinlock

Pronto para Offload de Hardware:

- O modo de offload XDP suporta execução em SmartNIC
- Compatível com placas de rede que suportam XDP

- Retorno para modos nativos do driver ou genéricos

Fundação da Tecnologia eBPF

O que é eBPF?

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

Características Principais:

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

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 Mapas Usados no OmniUPF:

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

Operações de Mapa:

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

Caminho de Dados XDP

Visão Geral do XDP

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

Modos de Anexação XDP

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

1. Modo de Offload XDP

Execução em Hardware (Melhor Desempenho):

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

Configuração:

```
xdp_attach_mode: offload
```

Limitações:

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

---

2. Modo Nativo XDP (Padrão para Produção)

Execução em Nível de Driver (Alto Desempenho):

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

Configuração:

```
xdp_attach_mode: native
```

Vantagens:

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

Drivers Suportados:

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

---

3. Modo Genérico XDP

Emulação de Software (Compatibilidade):

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

Configuração:

```
xdp_attach_mode: generic
```

Casos de Uso:

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

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

Códigos de Retorno XDP

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

Código de Retorno	Significado	Uso no OmniUPF
`XDP_PASS`	Enviar pacote para a pilha de rede do kernel	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	Não utilizado atualmente

Código de Retorno	Significado	Uso no OmniUPF
	mesma interface	
`XDP_REDIRECT`	Enviar pacote para interface diferente	Caminho principal de encaminhamento (N3 ↔ N6)
`XDP_ABORTED`	Erro de processamento, descartar pacote e registrar	Erros de programa eBPF

Pipeline de Processamento de Pacotes

Estrutura do Programa

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

Chamadas de Cauda:

- Permitem que programas eBPF chamem outros programas eBPF
- Reutiliza o mesmo quadro de pilha (profundidade de pilha limitada)
- Habilita design modular de pipeline
- Profundidade máxima de 33 chamadas de cauda

Processamento de Pacotes de Uplink

Processamento de Pacotes de Downlink

Arquitetura do Mapa eBPF

Layout de 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 de espaço de usuário via socket UDP.

Arquitetura de Buffering

Detalhes da Encapsulação do Buffer

Quando o buffering é ativado (bit de ação FAR 2 definido), o programa eBPF:

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

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

2. **Expande o Cabeçalho do Pacote**:

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

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

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

4. **Constrói o Cabeçalho UDP**:

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

5. **Constrói o Cabeçalho GTP-U**:

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

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

Operação de Limpeza do Buffer

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

Parâmetros de Gerenciamento de Buffer

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

Aplicação de QoS

Algoritmo de Limitação de Taxa

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

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**:

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

2. **Verificação de Taxa**:

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

3. **Taxa Máxima de Pacotes**:

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

Características de Desempenho

Taxa de Transferência

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

Quebra de Latência

Latência Total de Processamento de Pacotes (Nativo XDP):

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

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

Utilização da CPU

Capacidade de Processamento por Núcleo:

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

Uso da CPU pela Taxa de Pacotes:

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

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

Largura de Banda de Memória

Acesso ao Mapa eBPF:

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

Largura de Banda de Memória Necessária:

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

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

Escalabilidade e Ajustes

Escalonamento Horizontal

Múltiplas Instâncias UPF:

Distribuição de Sessões:

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

Escalonamento Vertical

Ajustes de CPU:

1. Habilitar afinidade de CPU para processamento XDP
2. Usar RSS (Receive Side Scaling) para distribuir filas RX
3. Fixar programas eBPF em núcleos específicos

Ajustes de NIC:

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

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

- CPU: (20 Gbps ÷ 10 Gbps por núcleo) × 1.5 = 3-4 núcleos
- Memória: (1M × 212 B × 3) + 100 MB ≈ 750 MB
- Rede: (20 Gbps × 2) + 10 Gbps = 50 Gbps de interfaces

Documentação Relacionada

- **Guia de Operações 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

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

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

Modo Genérico (Padrão)

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

Vantagens:

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

Desvantagens:

- Desempenho mais baixo (~1-2 Mpps por núcleo)
- Pacotes já passaram pelo driver antes do processamento XDP

Configuração:

```
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 o modo genérico
- Funciona em bare metal e VMs com SR-IOV

Desvantagens:

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

Configuração:

```
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 (Máximo Desempenho)

Descrição: O programa XDP é executado diretamente no hardware SmartNIC

Vantagens:

- Máximo desempenho (~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 throughput
- Computação de borda com requisitos de latência rigorosos
- Ambientes onde os recursos de CPU são limitados

Requisitos:

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

Parâmetros de Configuração

Interfaces de Rede

Parâmetro	Descrição	Tipo	Padrão
`interface_name`	Interfaces de rede para tráfego N3/N6/N9 (pontos de anexação XDP)	Lista	`[lo]`
`n3_address`	Endereço IPv4 para a interface N3 (GTP-U da RAN)	IP	`127.0.0.1`
`n9_address`	Endereço IPv4 para a interface N9 (UPF-to-UPF para ULCL)	IP	Mesmo que `n3_address`

Exemplo:

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

Configuração PFCP

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

Parâmetro	Descrição	Tipo	Padrão
	como inativo		
`heartbeat_interval`	Intervalo de heartbeat PFCP (segundos)	Inteiro	`5`
`heartbeat_timeout`	Tempo limite de heartbeat PFCP (segundos)	Inteiro	`5`

Exemplo:

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

API e Monitoramento

Parâmetro	Descrição	Tipo	Padrão
`api_address`	Endereço local para o servidor REST API	Host:Port	`:8080`
`metrics_address`	Endereço local para o endpoint de métricas do Prometheus	Host:Port	`:9090`
`logging_level`	Nível de log (`trace`, `debug`, `info`, `warn`, `error`)	String	`info`

Exemplo:

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

Gerenciamento de Caminho GTP

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

Exemplo:

```yaml
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
`max_sessions`	Número máximo de sessões concorrentes	Inteiro	`65535`	Usado para calcular tamanhos de mapa
`pdr_map_size`	Tamanho do mapa eBPF PDR	Inteiro	`0`	`max_sessions × 2`
`far_map_size`	Tamanho do mapa eBPF FAR	Inteiro	`0`	`max_sessions × 2`
`qer_map_size`	Tamanho do mapa eBPF QER	Inteiro	`0`	`max_sessions`
`urr_map_size`	Tamanho do mapa eBPF URR	Inteiro	`0`	`max_sessions × 2`

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

Exemplo:

```yaml
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:

```yaml
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
`buffer_port`	Porta UDP para pacotes armazenados do eBPF	Inteiro	`22152`
`buffer_max_packets`	Número máximo de pacotes a serem armazenados por FAR	Inteiro	`10000`
`buffer_max_total`	Número máximo total de pacotes armazenados (0=ilimitado)	Inteiro	`100000`
`buffer_packet_ttl`	TTL para pacotes armazenados em segundos (0=sem expiração)	Inteiro	`30`
`buffer_cleanup_interval`	Intervalo de limpeza do buffer em	Inteiro	`60`

Parâmetro	Descrição	Tipo	Padrão
	segundos (0=sem limpeza)		

Exemplo:

```yaml
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
`feature_ueip`	Habilitar alocação de IP UE pelo OmniUPF	Booleano	`false`
`ueip_pool`	Pool de IP para alocação de IP UE (requer `feature_ueip`)	CIDR	`10.60.0.0/24`
`feature_ftup`	Habilitar alocação de F-TEID pelo OmniUPF	Booleano	`false`
`teid_pool`	Tamanho do pool de TEID para alocação de F-TEID (requer `feature_ftup`)	Inteiro	`65535`

Exemplo (alocação de IP UE):

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

Exemplo (alocação de F-TEID):

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

Métodos de Configuração

Arquivo de Configuração YAML (Recomendado)

Arquivo: `config.yml`

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

# Configuração PFCP
pfcp_address: :8805
pfcp_node_id: 10.100.50.241
```

```yaml
pfcp_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
```

Iniciando o OmniUPF:

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

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 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: vmbr0
```

Configuração do OmniUPF:

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

---

2. SR-IOV Passthrough (XDP Nativo)

Caso de uso: Produção de alto desempenho

Configuração:

- Dispositivo de Rede: Função Virtual SR-IOV
- Modo XDP: `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. PCI Passthrough (XDP Nativo)

Caso de uso: NIC dedicada para uma única VM

Configuração:

- NIC física inteira 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 \
  ...
```

SR-IOV Passthrough:

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

VMware ESXi

vSwitch Padrão (XDP Genérico):

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

Em 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 bytes × 1.000.000 pps × 8 bits/byte = 520 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 bytes × 1.000.000 pps × 8 bits/byte = 12.288 Mbps ≈ 12.3 Gbps
```

Pacotes IP Nativos (Interface N6):

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

Pacote N6 Mínimo:

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

Throughput a 1 Mpps com pacotes N6 mínimos:

```
29 bytes × 1.000.000 pps × 8 bits/byte = 232 Mbps
```

Pacote N6 Máximo (MTU de 1500):

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

Throughput a 1 Mpps com pacotes N6 máximos:

```
1500 bytes × 1.000.000 pps × 8 bits/byte = 12.000 Mbps = 12 Gbps
```

Exemplos de Desempenho do Mundo Real

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

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

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

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

Padrão de Tráfego	Tamanho do Pacote	Throughput a 10 Mpps	Caso de Uso Típico
			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 Prática 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
1 Mpps	~1.0 Gbps	~1.0 Gbps	Pequeno site celular, gateway IoT
5 Mpps	~4.9 Gbps	~4.8 Gbps	Site celular médio, empresa
10 Mpps	~9.9 Gbps	~9.6 Gbps	Grande site celular, pequena cidade
20 Mpps	~19.7 Gbps	~19.2 Gbps	Área metropolitana, cidade média
40 Mpps	~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 aplicativos 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

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

NICs Mellanox/NVIDIA

Modelo	Driver	Suporte a XDP	Modo	Desempenho
Mellanox ConnectX-5	mlx5	Sim	Nativo	~12 Mpps
Mellanox ConnectX-6	mlx5	Sim	Nativo	~20 Mpps

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

NICs Broadcom

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

Outros Fornecedores

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

Verificando Suporte a XDP da NIC

Verifique se o driver suporta XDP:

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

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

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

Verifique a anexação do programa XDP:

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

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

NICs Recomendadas por Caso de Uso

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

Caso de Uso	NIC Recomendada	Modo	Capacidade Mpps	Throughput (N3)	Cenário de Implantação
Teste/	Qualquer NIC	Genérico	1-2 Mpps	1-2 Gbps	Testes em

Caso de Uso	NIC Recomendada	Modo	Capacidade Mpps	Throughput (N3)	Cenário de Implantação
Desenvolvimento	(VirtIO, E1000)				laboratório, PoC
Site de Pequena Célula	Intel X710, Mellanox CX-5	Nativo	5-10 Mpps	5-10 Gbps	Célula rural, empresa
Célula Média/ Metrô	Intel E810, Mellanox CX-6	Nativo	10-20 Mpps	10-20 Gbps	Célula urbana, cidade pequena
Grande Metrô	Mellanox CX-6, Intel E810 (duplo)	Nativo	20-40 Mpps	20-40 Gbps	Área metropolitana, cidade média
Hub Regional	Mellanox BlueField, Netronome Agilio	Offload	40+ Mpps	40+ Gbps	Agregação regional
VM Proxmox (Bridge)	VirtIO	Genérico	1-2 Mpps	1-2 Gbps	Apenas teste
VM Proxmox (SR-IOV)	Intel X710/ E810 VF, Mellanox CX-5 VF	Nativo	5-10 Mpps	5-10 Gbps	VM de produção

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

```yaml
# Configuração de desenvolvimento
interface_name: [eth0]
xdp_attach_mode: generic
api_address: :8080
pfcp_address: :8805
pfcp_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

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

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

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

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

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

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

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

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

Cenário: OmniUPF atendendo simultaneamente redes 5G e 4G

```yaml
# Configuração de múltiplos modos
interface_name: [eth0, eth1]
xdp_attach_mode: native
api_address: :8080
pfcp_address: :8805
pfcp_node_id: 10.50.1.100
n3_address: 10.50.1.100
```

```yaml
n9_address: 10.50.1.101
metrics_address: :9090
logging_level: info
max_sessions: 300000
gtp_peer:
  - 10.50.2.10:2152  # gNB 5G
  - 10.50.2.20:2152  # eNodeB 4G (via SGW-U)
gtp_echo_interval: 15
pfcp_remote_node:
  - 10.50.3.10  # OmniSMF (5G)
  - 10.50.3.20  # OmniPGW-C (4G)
heartbeat_interval: 10
feature_ueip: true
ueip_pool: 10.60.0.0/16
```

Exemplo 6: Modo Offload SmartNIC

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

```yaml
# Configuração de offload SmartNIC
interface_name: [enp1s0np0]  # Interface SmartNIC
xdp_attach_mode: offload
api_address: :8080
pfcp_address: 10.10.1.50:8805
pfcp_node_id: 10.10.1.50
n3_address: 10.10.1.50
metrics_address: :9090
logging_level: warn  # 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
pfcp_remote_node:
  - 10.10.3.10
heartbeat_interval: 15
buffer_max_packets: 100000
buffer_max_total: 1000000
```

Dimensionamento de Mapas e Planejamento de Capacidade

Dimensionamento Automático (Recomendado)

Defina `max_sessions` e deixe o OmniUPF calcular tamanhos de mapa 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 cálculo automático por requisitos personalizados:

```
max_sessions: 100000
pdr_map_size: 300000  # Suporte a 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) = **100.000**

Requisitos de Memória

Uso de memória por sessão:

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

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

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

```
# Verifique o limite atual
ulimit -l

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

Documentação Relacionada

- **Guia de Arquitetura** - Detalhes técnicos de eBPF/XDP e otimização de desempenho
- **Guia de Gerenciamento de Regras** - Configuração de PDR, FAR, QER, URR
- **Guia de Monitoramento** - Estatísticas, monitoramento de capacidade e alertas
- **Guia de UI Web** - Operações do painel de controle
- **Guia de Operações** - Visão geral da arquitetura e implantação do UPF

Guia de Monitoramento

Índice

Visão Geral

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

Categorias de Monitoramento

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

Monitoramento de Estatísticas

Estatísticas da Interface N3/N6

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

Métricas:

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

Comportamento Esperado:

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

Estatísticas de XDP

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

Métricas:

- **Abortado**: O programa XDP encontrou um erro (deve ser sempre 0)
- **Descartar**: Pacotes 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 Endereço
- **RX GTP ECHO**: Solicitação/Resposta GTP-U Echo (keepalive)
- **RX GTP OUTRO**: 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 Mensagem de Controle da Internet (ping, erros)
- **RX ICMP6**: Pacotes ICMPv6
- **RX IP4**: Pacotes IPv4
- **RX IP6**: Pacotes IPv6
- **RX OUTRO**: Outros protocolos
- **RX TCP**: Pacotes do Protocolo de Controle de Transmissão
- **RX UDP**: Pacotes do Protocolo de Datagramas do Usuário

Casos de Uso:

- **Monitorar contagem de GTP-U PDU**: Indicador primário de tráfego do usuário
- **Verificar ICMP para conectividade**: Teste de acessibilidade da rede
- **Acompanhar a proporção TCP vs UDP**: Padrões de tráfego de aplicativos
- **Detectar protocolos inesperados**: Questões 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 rota em cache (caminho rápido)
- **OK**: Consultas de rota bem-sucedidas

Consulta FIB IPv6:

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

Indicadores de Desempenho:

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

---

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
- **Crítico**: Alto - 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
- **Crítico**: Crítico - Estabelecimento de sessão falha se cheio

pdr_map_downlin_ip6 (PDRs de Downlink - IPv6):

- **Capacidade**: 131.070 entradas
- **Tamanho da Chave**: 16 B (endereço IPv6 do UE)
- **Tamanho do Valor**: 208 B (informações do PDR)
- **Uso de Memória**: ~29 MB
- **Crítico**: Crítico - Estabelecimento de sessão IPv6 falha se cheio

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
- **Crítico**: Crítico - Tráfego de uplink falha se cheio

qer_map (Regras de Aplicação de QoS):

- **Capacidade**: 65.535 entradas
- **Tamanho da Chave**: 4 B (ID QER)
- **Tamanho do Valor**: 32 B (parâmetros de QoS)
- **Uso de Memória**: ~2,3 MB
- **Crítico**: Médio - 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
- **Crítico**: Baixo - Afeta apenas a cobrança

Limites de Capacidade

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

Procedimento de Aumento de Capacidade

Antes de aumentar a capacidade:

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

Passos para aumentar a capacidade do mapa:

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

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

Métricas de Desempenho

Taxa de Processamento de Pacotes

Cálculo:

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

Exemplo:

- Pacotes RX iniciais: 7.000
- Após 10 segundos: 17.000
- Taxa de Pacotes = (17.000 - 7.000) / 10 = 1.000 pps

Metas de Desempenho:

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

Indicadores de Gargalo:

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

Cálculo de Throughput

Cálculo:

```
Throughput (Mbps) = (Delta de Contagem de Bytes × 8) / (Delta de Tempo em segundos × 1.000.000)
```

Exemplo:

- Bytes RX iniciais: 500 MB
- Após 60 segundos: 800 MB
- Throughput = (300 MB × 8) / (60 × 1.000.000) = 40 Mbps

Planejamento de Capacidade:

- Monitorar horários de pico de throughput (ex: 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:

```
Taxa de Descarte (%) = (Pacotes Descartados / Total de Pacotes RX) × 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 (problema sério 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 XDP abortados > 0
- Taxa de descarte > 5%
- Verificação de saúde do UPF falhou

Alertas de Aviso (Resposta dentro de 1 hora):

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

Alertas Informativos (Monitorar tendências):

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

Configuração de Alertas

Os alertas podem ser configurados via:

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

---

Planejamento de Capacidade

Estimativa de Capacidade de Sessão

Calcular sessões máximas:

```
Sessões Máximas = 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

Sessões Máximas = min(131.070 / 2, 131.070 / 2, 65.535) = **65.535 sessões**

Capacidade de Memória

Calcular memória total do mapa eBPF:

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

Exemplo de Configuração:

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

Considerações 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 contagem de sessões de pico diárias
2. Calcular taxa de crescimento semanal
3. Extrapolar para o limite de capacidade

Fórmula da Taxa de Crescimento:

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

Exemplo:

- Sessões atuais: 30.000
- Capacidade: 65.535 sessões
- Crescimento semanal: 2.000 sessões

- Semanas até a capacidade: (65.535 - 30.000) / 2.000 = **17,8 semanas**

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

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

Erros de Processamento de XDP

Sintomas: XDP abortado > 0

Diagnóstico:

1. Navegar até Estatísticas → Estatísticas de XDP
2. Verificar contagem 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 no 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 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 utilização da CPU no host UPF
4. Verificar 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 CPU ou habilitar RSS (Receber Escalonamento Lateral)
- Atualizar interfaces de rede para maior velocidade
- Ajustar a função de hash do mapa eBPF

Documentação Relacionada

- **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 Resolução 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 da infraestrutura e operacional
- **Computação de borda** - Minimizar a latência para cenários de quebra 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 um 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 (Instância Única)

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. **Detecção 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

Passos Chave:

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)

Passos Chave:

1. **Receber:** Pacote IP simples da Internet destinado ao UE (10.60.0.1)
2. **Combinação de PDR:** Lookup PDR de downlink por UE IP (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 se conectar à interface PFCP do OmniUPF (por exemplo, `192.168.1.10:8805`)
- **PGWU-C**: Deve se conectar à **mesma** interface PFCP do OmniUPF

Rede:

- **Único endereço IP** para as interfaces N3 e N9
- **Endereços IP diferentes** para SGWU-C e PGWU-C (se executando na mesma máquina, use portas diferentes)

Configuração do OmniUPF

config.yml:

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

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

# Interfaces do Plano de 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 Prometheus

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

# Capacidade
max_sessions: 100000                # Máximo de sessões de UE simultâneas
```

Configuração Chave:

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

Configuração do Plano de Controle

Configuração do SGWU-C

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

# Interface S1-U (igual ao n3_address do OmniUPF)
sgwu_s1u_address: "10.0.1.10"

# Interface N9 para encaminhamento para PGWU (igual ao OmniUPF)
sgwu_n9_address: "10.0.1.10"
```

Configuração do PGWU-C

```
# Apontar para a MESMA interface PFCP do OmniUPF
upf_pfcp_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 se conectam 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 de UE e Estabelecimento de Sessão PDU

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

Sessões PFCP Criadas:

Sessão SGWU (do OmniSGW-C):

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

Sessão PGWU (do OmniPGW-C):

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

Monitoramento e Verificação

Verificar se o Loopback N9 está Ativo

Verificar Logs do XDP:

```
# Ver 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

Verificar 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)
- Contagens de `xdp_tx` ou `xdp_redirect` para encaminhamento de loopback

Resolução 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` ≠ `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 desvio de TEID

Diagnóstico:

1. **Verificar Sessões PFCP:**

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

2. **Verificar Configuração FAR:**

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

Solução: Garantir que o PGWU-C crie uma sessão com TEID correspondente que o SGWU-C usa para 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:

```
# Verificar 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 busca)
- Verificar se o limite de taxa QER não está causando perdas e retransmissões
- Verificar por buffering excessivo de pacotes

Perda de Pacotes Durante Handover

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 handover
```

Verificação:

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

Benefícios do Loopback N9

Desempenho

Métrica	Duas Instâncias	Instância Única (Loopback N9)	Melhoria
Latência	1-5 ms	< 1 µs	**1000x mais rápido**
Taxa de Transferência	Limitada pela rede	Limitada pela CPU/ memória	**2-3x maior**
Uso de CPU	2× passes XDP + pilha de rede	1× passe XDP	**Redução de 40-50%**
Perda de Pacotes	Risco durante congestionamento de rede	Zero (em memória)	**Eliminado**

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
- **Resolução de Problemas Mais Fácil:** Rastreamento de pacotes único, saída de depuração eBPF única

Casos de Uso

Ideal Para:

- ✅ **Computação de Borda:** Minimizar latência para quebra local
- ✅ **Implanta��ões Pequenas/Médias:** < 100K assinantes
- ✅ **Laboratório/Teste:** Plano de usuário EPC completo em uma única VM

- 🔰 **Restrições de Custo:** Orçamento de hardware limitado

Não Recomendado Para:

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

Comparação com Outros Modos de Implantação

Instância Única (Loopback N9) vs. Instâncias Separadas

Recurso	Loopback N9	Separado	Containers
Latência	⚡ < 1 µs	🔰 1-5 ms	🔰 5-20 ms
Taxa de Transferência	⚡ 40+ Gbps	🔰 20+ Gbps	🔰 10+ Gbps
Infraestrutura	🔰 1 servidor	🔰 2 servidores	⚠ 1 servidor, 2 VMs
Complexidade	🔰 Simples	🔰 Complexo	⚠ Moderado
Custo	🔰 Mais Baixo	🔰 Mais Alto	⚠ Médio
Escalonamento	⚠ Apenas vertical	🔰 Horizontal	🔰 Horizontal
Redundância	🔰 Ponto único de falha	🔰 Redundância geográfica	⚠ Redundância local

Resumo

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

- 🔰 **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` - sem bandeiras ou configurações especiais necessárias. O datapath eBPF do OmniUPF detecta condições de loopback e processa pacotes inline.

Para mais informações:

- **Configuração:** CONFIGURATION.md
- **Arquitetura:** ARCHITECTURE.md

- **Operações:** OPERATIONS.md
- **Resolução de Problemas:** TROUBLESHOOTING.md

Guia de Gerenciamento de Regras

Índice

Visão Geral

OmniUPF usa um conjunto de regras interconectadas para classificar, encaminhar, moldar e rastrear o tráfego do plano do usuário. Essas regras são instaladas pelo SMF via PFCP e armazenadas em mapas eBPF para processamento de pacotes de alto desempenho. Compreender essas regras e seus relacionamentos é crítico para operar e solucionar problemas do UPF.

Tipos de Regras

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

Fluxo de Processamento de Regras

Regras de Detecção de Pacotes (PDR)

Propósito

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

Estrutura da PDR

PDRs de Uplink

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

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

- Inteiro 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 pacote

Exemplo:

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

PDRs de Downlink

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

Campo Chave: Endereço IP do UE

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

Campos de Valor:

- **ID do FAR**: Referência à regra de ação de encaminhamento
- **ID do QER**: Referência à regra de aplicação de QoS (opcional)
- **IDs do URR**: Lista de regras de relatório de uso (opcional)
- **Modo SDF**: Modo de filtro de Fluxo de Dados de Serviço
  - `Sem SDF`: Sem filtragem, todo o tráfego combina
  - `Somente SDF`: Somente tráfego que combina com SDF é encaminhado
  - `SDF + Padrão`: Tráfego que combina com SDF usa regras específicas, outro tráfego usa o FAR padrão
- **Filtros SDF**: Filtros específicos de aplicação (portas, protocolos, intervalos de IP)

Processo de Busca:

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

Exemplo:

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

Filtros SDF (Fluxo de Dados de Serviço)

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

Casos de Uso:

- Diferenciar tráfego do YouTube de 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 extremidade de destino.

Estrutura da FAR

Flags de Ação

As ações da FAR são flags bitwise que podem ser combinadas:

Flag	Bit	Valor	Descrição
ENCAMINHAR	1	2	Encaminhar pacote para o destino
BUFFER	2	4	Armazenar pacote no buffer
DESCARTAR	0	1	Descartar pacote
NOTIFICAR	3	8	Enviar notificação para o plano de controle
DUPLICAR	4	16	Duplicar pacote para múltiplos destinos

Combinações Comuns de Ação:

- `Ação: 2 (ENCAMINHAR)` - Encaminhamento normal (mais comum)
- `Ação: 6 (ENCAMINHAR + BUFFER)` - Encaminhar e armazenar durante a

transferência
- `Ação: 4 (BUFFER)` - Apenas armazenar (durante a troca de caminho)
- `Ação: 1 (DESCARTAR)` - Descartar pacote (raro, geralmente para aplicação de políticas)

Controle de Buffering

A flag BUFFER (bit 2) controla o armazenamento de pacotes durante eventos de mobilidade.

Operações de Buffering:

- **Habilitar Buffer**: Definir bit 2 da ação FAR (Ação |= 4)
- **Desabilitar Buffer**: Limpar bit 2 da ação FAR (Ação &= ~4)
- **Limpar Buffer**: Repetir todos os pacotes armazenados usando as regras FAR atuais
- **Limpar Buffer**: Descartar todos os pacotes armazenados sem encaminhar

Criação do Cabeçalho Externo

Determina se a encapsulação GTP-U deve ser adicionada.

FAR de Uplink (N3 → N6):

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

FAR de Downlink (N6 → N3):

- Criação do 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. Navegar para Regras → Aba FARs
2. Inserir ID do FAR no campo de busca
3. Clicar em "Buscar" para visualizar detalhes da FAR

Informações Exibidas:

- ID do FAR
- Ação (numérica + flags decodificadas)

- Status de buffering (LIGADO/DESLIGADO)
- Criação do 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 de QoS

QFI (Identificador de Fluxo de QoS):

- Identificador de 6 bits para fluxos de QoS 5G
- Valores de 1 a 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 de Bits Máxima (MBR):

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

Taxa de Bits Garantida (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

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 carga útil

Volume de Downlink:

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

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 Evento:

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

Formatação de Exibição de Volume

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

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

Exemplo:

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

Fluxo de Relatório da URR

Relacionamentos de 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 do Cabeçalho Externo: Verdadeira
IP Remoto: 200.198.5.10
TEID: 5678
```

FAR ID 2 (Uplink):

```
Ação: 2 (ENCAMINHAR)
Criação do 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. Navegar para Sessões
2. Encontrar UE por IP ou TEID
3. Clicar em "Expandir" para visualizar todas as regras (PDR, FAR, QER, URR)

Via Página de Regras:

1. Navegar para Regras
2. Usar busca por TEID (uplink) ou IP do UE (downlink) na aba PDR
3. Anotar o ID do FAR, ID do QER, IDs do URR
4. Mudar para as abas FAR/QER/URR para visualizar regras referenciadas

Habilitar/Desabilitar Buffering

Cenário: Durante a transferência, armazenar pacotes para evitar perda

Passos:

1. Navegar para Regras → FARs
2. Inserir ID do FAR no campo de busca
3. Clicar em "Buscar"
4. Se o buffering estiver DESLIGADO, clicar em "Habilitar Buffering"
5. Verificar se o bit 2 da ação FAR está definido (o valor da ação aumenta em 4)

Alternativa via Página de Buffers:

1. Navegar para Buffers
2. Visualizar FARs com buffering habilitado
3. Clicar em "Desabilitar Buffer" quando a transferência for concluída

Monitorar Conformidade com QoS

Verificar se o tráfego está sendo limitado por taxa:

1. Navegar para Regras → QERs
2. Encontrar ID do QER associado à sessão do UE
3. Anotar os valores de MBR Uplink e MBR Downlink
4. Comparar com a taxa de crescimento do volume da URR

Calcular Throughput Médio:

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

```
em segundos × 1000)
```

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

Rastrear Uso de Dados

Monitorar volumes da URR:

1. Navegar para Regras → URRs
2. Visualizar volumes de uplink, downlink e total
3. Classificar por Volume Total para encontrar os maiores usuários
4. Atualizar 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 Tráfego Fluindo

Verificar PDR:

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

Verificar FAR:

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

Verificar QER:

1. Verificar se o Estado do Portão está Aberto (0)
2. Verificar se a MBR não é muito restritiva

Pacotes Sendo Descartados

Verificar Limitação de Taxa da QER:

1. Navegar para Regras → QERs
2. Verificar se a MBR é adequada para a carga de tráfego
3. Verificar se o crescimento do volume da URR corresponde ao throughput

esperado

Verificar Ação da FAR:

1. Navegar para Regras → FARs
2. Verificar se a ação é ENCAMINHAR, não DESCARTAR
3. Verificar se o buffering não está preso em modo apenas BUFFER

Problemas de Buffering

Pacotes presos no buffer:

1. Navegar para a página de Buffers
2. Verificar o timestamp do pacote mais antigo
3. Se > 30 segundos, a transferência pode ter falhado
4. Limpar manualmente ou limpar o buffer
5. Desabilitar o buffering na FAR

Overflow de Buffer:

1. Verificar total de pacotes vs. Máximo Total (padrão 100.000)
2. Verificar pacotes por FAR vs. Máximo por FAR (padrão 10.000)
3. Limpar buffers se estiverem cheios
4. Investigar por que o buffering não foi desabilitado

URR Não Rastreando

Contadores de volume em zero:

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

Volume não relatando ao SMF:

1. Verificar configuração do Relatório de Sessão PFCP
2. Verificar gatilhos de relatório da URR (limites de volume/tempo)
3. Revisar 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 do PFCP** - Gerenciamento de sessão PFCP e instalação de regras
- **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

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 um diagnóstico mais profundo, verifique:

```
# 1. Verifique se o OmniUPF está em execução
systemctl status omniupf  # ou ps aux | grep eupf

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

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

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

# 5. Verifique os logs do kernel em busca de erros
```

```
dmesg | tail -50
journalctl -u omniupf -n 50
```

Ferramentas de Diagnóstico

API REST do OmniUPF

Verifique o status do UPF:

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

Verifique as associações PFCP:

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

Verifique a contagem de sessões:

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

Verifique a capacidade do mapa eBPF:

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

Verifique as estatísticas de pacotes:

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

Verifique as estatísticas do XDP:

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

Inspeção do Mapa eBPF

Liste todos os mapas eBPF:

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

Mostre 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 GTP-U na N3:

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

Capture o tráfego PFCP na N4:

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

Monitore contadores de pacotes:

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

Verifique a tabela de roteamento:

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

Verifique a tabela ARP:

```
ip neigh show
```

Problemas de Instalação

Problema: "sistema de arquivos eBPF não montado"

Sintomas:

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

Causa: sistema de arquivos eBPF não montado

Resolução:

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

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

# Verifique o montado
mount | grep bpf
```

Problema: "Operação não permitida" ao carregar eBPF

Sintomas:

```
ERRO[0000] falha ao carregar o programa eBPF: operação não permitida
```

Causa: Capacidades insuficientes ou limites de memória bloqueados

Resolução:

```
# Verifique o limite atual de memória bloqueada
ulimit -l

# Defina memória bloqueada ilimitada (necessária para eBPF)
ulimit -l unlimited

# Torne persistente (adicione ao /etc/security/limits.conf)
echo "* soft memlock unlimited" | sudo tee -a /etc/security/
limits.conf
echo "* hard memlock unlimited" | sudo tee -a /etc/security/
limits.conf

# Execute o OmniUPF com as capacidades necessárias
sudo setcap cap_sys_admin,cap_net_admin,cap_bpf+eip /usr/bin/eupf
```

```
# Ou execute com sudo
sudo ./eupf
```

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)
# - Faltando dois pontos após as chaves
# - Strings não citadas com caracteres especiais
# - Itens de lista sem hífens

# Exemplo de YAML correto:
cat > config.yml <<EOF
interface_name: [eth0]
xdp_attach_mode: generic
api_address: :8080
pfcp_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 mude a porta do OmniUPF na configuração
api_address: :8081
pfcp_address: :8806
metrics_address: :9091
```

Problema: ID do Nó PFCP inválido

Sintomas:

```
ERRO[0000] pfcp_node_id 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)
pfcp_node_id: 10.100.50.241

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

Problemas de Associação PFCP

Problema: Nenhuma associação PFCP estabelecida

Sintomas:

- A interface da Web mostra "Nenhuma associação"
- Os logs do SMF mostram "Falha na configuração da associação PFCP"

Diagnóstico:

```
# 1. Verifique se o servidor PFCP está ouvindo
sudo netstat -ulpn | grep 8805

# 2. Verifique as regras do firewall
sudo iptables -L -n | grep 8805
sudo ufw status

# 3. Capture o tráfego PFCP
tcpdump -i any -n udp port 8805 -vv

# 4. Verifique as associações PFCP via API
curl http://localhost:8080/api/v1/upf_pipeline
```

Causas Comuns & Resoluções:

Firewall bloqueando PFCP

Resolução:

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

ID do Nó PFCP incorreto

Resolução:

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

Rede inacessível ao SMF

Resolução:

```
# Teste a conectividade com o SMF
ping <SMF_IP>
```

```
# Verifique o roteamento para o SMF
ip route get <SMF_IP>

# Adicione a rota se estiver faltando
sudo ip route add <SMF_NETWORK>/24 via <GATEWAY>
```

SMF configurado com IP do UPF incorreto

Resolução:

- Verifique a configuração do SMF para o endereço do UPF
- Certifique-se de que o SMF tenha o IP `pfcp_node_id` do UPF configurado
- Verifique se o SMF pode rotear para a rede N4 do UPF

Problema: Falhas de heartbeat PFCP

Sintomas:

```
WARN[0030] tempo limite de heartbeat PFCP para associação
10.100.50.10
```

Diagnóstico:

```
# Verifique as estatísticas PFCP
curl http://localhost:8080/api/v1/upf_pipeline | jq '.associations[]
| {remote_id, uplink_teid_count}'

# Monitore os logs de heartbeat
journalctl -u omniupf -f | grep heartbeat
```

Causas & Resoluções:

Perda de pacotes na rede

Resolução:

```
# Verifique a perda de pacotes para o SMF
ping -c 100 <SMF_IP> | grep loss

# Se a perda for alta, investigue a rede:
# - Verifique o status do link
# - Verifique a saúde do switch/roteador
# - Verifique se há congestionamento
```

Intervalo de heartbeat muito agressivo

Resolução:

```
# Aumente o intervalo de heartbeat
heartbeat_interval: 30  # Aumentar de 5 para 30 segundos
heartbeat_retries: 5    # Aumentar tentativas
heartbeat_timeout: 10   # Aumentar 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 UP
ip link show eth0

# 3. Capture o tráfego na interface
tcpdump -i eth0 -n -c 10

# 4. Verifique as estatísticas de pacotes
curl http://localhost:8080/api/v1/packet_stats
```

Resoluções:

Programa XDP não anexado

Resolução:

```
# Reinicie o OmniUPF para re-anexar XDP
sudo systemctl restart omniupf

# Verifique a anexação
ip link show eth0 | grep xdp
bpftool net list
```

Interface desligada 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 desligado, 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 pesquisa FIB
curl http://localhost:8080/api/v1/packet_stats | jq '.route_stats'

# Teste o roteamento para o IP do UE
ip route get 10.45.0.100

# Adicione a rota que estiver faltando
sudo ip route add 10.45.0.0/16 dev eth1  # Roteie o pool do UE para
N6
```

Limitação de taxa QER

Resolução:

```
# Verifique as estatísticas QER
curl http://localhost:8080/api/v1/sessions | jq '.[].qers'

# Se o MBR (Taxa Máxima de Bits) estiver muito baixo, solicite ao SMF
que atualize o QER
# Ou verifique se o tráfego excede as taxas configuradas
```

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

# Capture tráfego em ambas as interfaces
tcpdump -i eth0 -n udp port 2152 &  # N3
tcpdump -i eth1 -n not udp port 2152 &  # N6
```

Falha de Uplink (RX N3, sem TX N6):

Causa: Nenhuma ação FAR ou problema de roteamento para N6

Resolução:

```
# Verifique se o FAR tem ação FORWARD
curl http://localhost:8080/api/v1/sessions | jq '.[].fars[] |
select(.applied_action == 2)'

# Verifique se a rota N6 existe
ip route get 8.8.8.8  # Teste a rota para a internet

# Adicione a rota padrão se estiver faltando
sudo ip route add default via <N6_GATEWAY> dev eth1
```

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

Causa: 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, consulte o Guia de Modos XDP.

Problema: Programa XDP falhou ao carregar

Sintomas:

```
ERRO[0000] falha ao carregar o programa XDP: argumento inválido
```

Diagnóstico:

```
# Verifique o suporte XDP do kernel
grep XDP /boot/config-$(uname -r)

# Deve mostrar:
# CONFIG_XDP_SOCKETS=y
# CONFIG_BPF=y
# CONFIG_BPF_SYSCALL=y

# Verifique dmesg para erro detalhado
dmesg | grep -i bpf
```

Causas & Resoluções:

Kernel não possui suporte a XDP

Resolução:

```
# Recompile o kernel com suporte a XDP ou atualize para um kernel
mais novo
# Ubuntu 22.04+ tem XDP habilitado por padrão
sudo apt install linux-generic-hwe-22.04
sudo reboot
```

Falha de verificação do programa XDP

Resolução:

```
# Verifique os logs do OmniUPF para erros de verificador
journalctl -u omniupf | grep verifier

# Problemas comuns:
# - Complexidade do eBPF excede limites (aumente os limites do
kernel)
# - Acesso à memória inválido (bug no código eBPF)

# Aumente o nível de log do verificador eBPF para depuração
sudo sysctl kernel.bpf_stats_enabled=1
```

Problema: Contagem de abortos do XDP aumentando

Sintomas:

- Estatísticas do XDP mostram abortos > 0
- Aumento de quedas de pacotes

Diagnóstico:

```
# Verifique a contagem de abortos do XDP
curl http://localhost:8080/api/v1/xdp_stats | jq '.aborted'

# Monitore as estatísticas do XDP
watch -n 1 'curl -s http://localhost:8080/api/v1/xdp_stats'
```

Causa: programa eBPF encontrou erro em tempo de execução

Resolução:

```
# Verifique os logs do kernel para erros de eBPF
dmesg | grep -i bpf

# Reinicie o OmniUPF para recarregar o programa eBPF
sudo systemctl restart omniupf

# Se o problema persistir, habilite o log de 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  # Aumentar 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: Baixa taxa de transferência (abaixo do esperado)

Sintomas:

- Taxa de transferência < 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'

# Monitore o uso da CPU
top -bn1 | grep eupf

# 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:

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

# Verifique se o RSS está habilitado
ethtool -l eth0

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

Buffer bloat

Resolução:

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

Problema: Alta latência

Sintomas:

- Latência de ping > 50ms
- Degradação da experiência do usuário

Diagnóstico:

```
# Teste a latência para o UE
ping -c 100 <UE_IP> | grep avg

# Verifique pacotes em buffer
curl http://localhost:8080/api/v1/upf_buffer_info | jq
'.total_packets_buffered'
```

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

Resoluções:

Pacotes sendo bufferizados excessivamente

Resolução:

```
# Verifique por que os pacotes estão em buffer
curl http://localhost:8080/api/v1/upf_buffer_info | jq '.buffers[] |
{far_id, packet_count, direction}'

# Limpe os buffers se estiverem travados
# (reinicie o OmniUPF ou acione a modificação da sessão PFCP para
aplicar FAR)
```

Latência de pesquisa FIB

Resolução:

```
# Certifique-se de que o cache de rota esteja habilitado (opção de
tempo de compilação)
# Compile com BPF_ENABLE_ROUTE_CACHE=1

# Otimize a tabela de roteamento
# Use menos rotas, mais específicas em vez de muitas rotas pequenas
```

Problema: Quedas de pacotes sob carga

Sintomas:

- Taxa de queda 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 quedas 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 Hipervisor

Para instruções passo a passo sobre configuração de hipervisor, consulte 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 SR-IOV passthrough

```
# 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 Proxmox
# Hardware → Adicionar → Dispositivo PCI → Selecionar 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 → Alterações de Endereço MAC: Aceitar
# 4. Segurança → Transmissões Forjadas: Aceitar
```

VirtualBox: Desempenho muito baixo

Sintomas:

- Taxa de transferência < 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"
# - Alocar 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 da 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 trava 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 por bugs no driver
journalctl -k | grep -E "BUG:|panic:"
```

Resolução:

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

# 2. Desative o offload 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: Falha no estabelecimento da sessão

Sintomas:

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

Diagnóstico:

```
# Verifique os logs do OmniUPF para 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 '.[] |
select(.usage_percent > 90)'

# Aumente a capacidade (veja a seção mapa eBPF cheia acima)
```

Parâmetros PDR/FAR inválidos

Resolução:

```
# Verifique os logs do OmniUPF para 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 em buffer aumentando
- Pacotes não entregues após a transferência

Diagnóstico:

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

# Verifique 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 & Resoluções:

FAR nunca atualizado para FORWARD

Causa: SMF nunca enviou Modificação de Sessão PFCP 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  # Aumentar de 30 para 60 segundos
```

Overflow do buffer

Causa: Muitos pacotes em buffer 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 no Nível XDP

```
# Capture pacotes antes do XDP (tcpdump)
tcpdump -i eth0 -w /tmp/before_xdp.pcap

# Capture pacotes após o XDP (requer XDP_PASS)
tcpdump -i any -w /tmp/after_xdp.pcap

# Compare contagens de pacotes para identificar quedas
```

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 da topologia de rede
    - Arquivo de configuração (redigir informações sensíveis)
    - Trechos de log relevantes
    - Etapas para reproduzir

3. **Suporte da comunidade**:
    - GitHub Issues: https://github.com/edgecomllc/eupf/issues
    - Documentação: Consulte guias relacionados abaixo

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
- **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

Visão Geral

A Interface Web do 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 oferece:

- **Visibilidade em tempo real** sobre 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 via 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 em tempo real e métricas de desempenho
- 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 de mapas eBPF e monitoramento de capacidade
- **Config** - `/upf_config` - Configuração do UPF e endereços do dataplane
- **Rotas** - `/routes` - Rotas de UE e sessões de protocolo de roteamento (OSPF, BGP)
- **Capacidades XDP** - `/xdp_capabilities` - Suporte ao modo XDP e capacidades de desempenho
- **Logs** - `/logs` - Streaming de logs ao vivo

Visualização de Sessões

URL: `/sessions`

Recursos

A visualização de Sessões exibe todas as sessões PFCP ativas e associações das

instâncias do OmniUPF selecionadas.

Resumo de Associações PFCP

Exibe todas as associações PFCP ativas (conexões de controle do SMF/PGW-C):

Coluna	Descrição
Node ID	Identificador do nó SMF ou PGW-C (FQDN ou IP)
Endereço	Endereço IP do SMF/PGW-C para comunicação PFCP
Próximo ID de Sessão	Próximo ID de sessão PFCP disponível para esta associação

Propósito:

- Verificar a conectividade do SMF com o UPF
- Monitorar o número de conexões do plano de controle
- Rastrear a alocação de ID de sessão por associação

Tabela de Sessões Ativas

Exibe todas as sessões PFCP representando sessões PDU ativas de UE:

Coluna	Descrição
Local SEID	Identificador do ponto de extremidade da sessão atribuído pelo UPF
Remote SEID	Identificador do ponto de extremidade da sessão atribuído pelo SMF
UE IP	Endereço IPv4 ou IPv6 do equipamento do usuário
TEID	Identificador do Ponto de Extremidade do Tunnel GTP-U para tráfego uplink
PDRs	Número de regras de detecção de pacotes na sessão
FARs	Número de regras de ação de encaminhamento na sessão
QERs	Número de regras de aplicação de QoS na sessão
URRs	Número de regras de relatório de uso na sessão
Ações	Botão de expandir para visualizar informações detalhadas da regra

Recursos:

- **Filtrar por IP**: Encontrar sessões para um endereço IP específico de UE
- **Filtrar por TEID**: Encontrar sessões por ID de ponto de extremidade do 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
- **Regras de Ação de Encaminhamento (FARs)**: Flags de ação, criação de cabeçalho externo, pontos de extremidade 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, bytes totais)

Casos de Uso

Verificar Conectividade de UE:

1. Navegar para a visualização de Sessões
2. Inserir o endereço IP da UE no filtro
3. Confirmar que a sessão existe com o TEID correto
4. Expandir para verificar a configuração de PDR/FAR

Monitorar Contagem de Sessões:

- Verificar a contagem total de sessões no cabeçalho
- Comparar entre várias instâncias do UPF
- Rastrear o crescimento da sessão ao longo do tempo

Solução de Problemas de Sessões:

- Pesquisar por IP específico de UE ou TEID
- Expandir a sessão para inspecionar a configuração da regra
- Verificar os parâmetros de encaminhamento do FAR
- Verificar 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 está 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 relatórios de uso em todas as sessões.

Aba PDR - Regras de Detecção de Pacotes

Visualizar e inspecionar todos os PDRs no UPF:

PDRs de Uplink (N3 → N6):

- **TEID**: ID do ponto de extremidade do túnel GTP-U do gNB
- **FAR ID**: Regra de ação de encaminhamento associada
- **QER ID**: Regra de aplicação de QoS associada (se houver)
- **Remoção de Cabeçalho Externo**: Flag de desencapsulação GTP-U

PDRs de Downlink (N6 → N3):

- **UE IP**: Endereço IPv4 ou IPv6 do equipamento do usuário
- **FAR ID**: Regra de ação de encaminhamento associada
- **QER ID**: Regra de aplicação de QoS associada (se houver)
- **Modo SDF**: Modo de filtro de fluxo de dados de serviço (nenhum, apenas sdf, sdf + padrão)

PDRs IPv6:

- Tabelas separadas para PDRs de uplink e downlink IPv6
- Mesma estrutura que IPv4, mas indexada por endereços IPv6

Aba FAR - Regras de Ação de Encaminhamento

Visualizar todos os FARs com suas ações de encaminhamento e parâmetros:

Coluna	Descrição
FAR ID	Identificador único da regra de encaminhamento
Ação	Flags de ação de encaminhamento (FORWARD, DROP, BUFFER, DUPLICATE, NOTIFY)
Buffering	Status atual de buffering (Ativado/Desativado)
Destino	Parâmetros de criação de cabeçalho externo (TEID, endereço IP)

Flags de Ação FAR:

- **FORWARD (1)**: Encaminhar pacote para o destino
- **DROP (2)**: Descartar pacote
- **BUFFER (4)**: Armazenar pacote em 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 a ação do FAR imediatamente no mapa eBPF

Aba QER - Regras de Aplicação de QoS

Visualizar regras de QoS aplicadas aos fluxos de tráfego:

Coluna	Descrição
QER ID	Identificador único da regra de QoS
MBR (Uplink)	Taxa máxima de bits para tráfego uplink (kbps)
MBR (Downlink)	Taxa máxima de bits para tráfego downlink (kbps)
GBR (Uplink)	Taxa garantida de bits para tráfego uplink (kbps)
GBR (Downlink)	Taxa garantida de bits para tráfego downlink (kbps)
QFI	Identificador de Fluxo de QoS (marcação 5G)

Interpretação de QoS:

- **MBR = 0**: Sem limite de taxa
- **GBR = 0**: Melhor esforço (sem largura de banda garantida)
- **GBR > 0**: Fluxo de taxa garantida (priorizado)

Aba URR - Regras de Relatório de Uso

Visualizar regras de rastreamento de uso e contadores de volume:

Coluna	Descrição
URR ID	Identificador único da regra de relatório de uso
Volume de Uplink	Bytes enviados da UE para a rede de dados
Volume de Downlink	Bytes enviados da rede de dados para a UE
Volume Total	Total de bytes em ambas as direções
Método	Método de relatório (volume, tempo, evento)

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
- Limitado a 1000 URRs mais ativos para desempenho

Casos de Uso

Inspecionar Classificação de Tráfego:

1. Navegar para Regras → aba PDR
2. Pesquisar por TEID ou UE IP específico
3. Verificar se o PDR associa com o FAR e QER corretos

Solução de Problemas de Encaminhamento:

1. Navegar para Regras → aba FAR
2. Localizar FAR ID da sessão PDR
3. Verificar se a ação é FORWARD (não DROP ou BUFFER)
4. Verificar parâmetros de criação de cabeçalho externo

Monitorar Aplicação de QoS:

1. Navegar para Regras → aba QER
2. Verificar se os valores de MBR e GBR correspondem à política
3. Verificar a marcação QFI para fluxos 5G

Rastrear Uso de Dados:

1. Navegar para Regras → aba URR
2. Classificar por volume total para encontrar os maiores usuários
3. Monitorar o crescimento do volume ao longo do tempo
4. Verificar a integração de cobrança

Gerenciamento de Buffers

URL: `/buffers`

Recursos

A visualização de Buffers exibe os 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 dos dados armazenados
- **Total de FARs**: Número de FARs com pacotes armazenados
- **Máx. Por FAR**: Máximo de pacotes permitidos por FAR
- **Máx. Total**: Máximo total de pacotes armazenados
- **TTL do Pacote**: Tempo de vida para pacotes armazenados (segundos)

Buffers por FAR

Tabela de todos os FARs com pacotes armazenados:

Coluna	Descrição
FAR ID	Identificador da regra de ação de encaminhamento
Contagem de Pacotes	Número de pacotes armazenados para este FAR
Contagem de Bytes	Total de bytes armazenados para este FAR
Pacote Mais Antigo	Timestamp do pacote armazenado mais antigo
Pacote Mais Novo	Timestamp do pacote armazenado mais novo
Ações	Botões de controle de buffer (estilo pílula)

Ações de Controle de Buffer

Para cada FAR com pacotes armazenados, os seguintes botões de estilo pílula estão disponíveis:

Controle de Buffering:

- **Desativar Buffer** (vermelho): Desligar o buffering para este FAR (atualiza a flag de ação do FAR)
- **Ativar Buffer** (roxo): Ligar o buffering para este FAR

Operações de Buffer:

- **Limpar** (azul): Repetir todos os pacotes armazenados usando as regras atuais do FAR
- **Limpar Tudo** (cinza): Excluir todos os pacotes armazenados sem encaminhar

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, verificar se os pacotes estão sendo armazenados
2. Verificar status de buffering do FAR (deve estar ativado)
3. Monitorar contagem e idade dos pacotes

Completar Transferência:

1. Após a mudança de caminho, clicar em "Limpar" para repetir os pacotes armazenados
2. Verificar se os pacotes são encaminhados para o novo caminho
3. Clicar em "Desativar Buffer" para parar o buffering

Limpar Buffers Presos:

1. Identificar FARs com pacotes armazenados antigos (verificar timestamp mais antigo)
2. Clicar em "Limpar" para descartar pacotes obsoletos
3. Ou clicar em "Desativar Buffer" para evitar mais armazenamento

Solução de Problemas de Overflow de Buffer:

1. Verificar contagem total de pacotes vs. máximo total
2. Identificar FARs com buffering excessivo
3. Verificar se o SMF enviou modificação de sessão para desativar o buffering
4. Desativar 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.

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 descarte 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 busca 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 entre interfaces

Uso: Monitorar o equilíbrio de tráfego e problemas específicos da interface

Casos de Uso

Monitorar Carga de Tráfego:

1. Verificar taxas de pacotes RX/TX
2. Confirmar que o tráfego está fluindo em ambas as direções
3. Comparar tráfego N3 vs N6 (deve ser aproximadamente igual)

Detectar Descartes de Pacotes:

1. Verificar contador de pacotes descartados
2. Revisar contador de pacotes descartados do XDP
3. Investigar a causa nos logs se os descartes forem altos

Análise de Desempenho:

1. Monitorar a razão de pacotes processados vs. passados do XDP
2. Verificar abortos do XDP (indica erros)
3. Confirmar a distribuição de tráfego das interfaces N3/N6

Planejamento de Capacidade:

1. Rastrear a taxa de pacotes ao longo do tempo
2. Comparar com os limites de capacidade do UPF
3. Planejar escalonamento se estiver se aproximando dos limites

Atualizações em Tempo Real

As estatísticas atualizam automaticamente a cada 10 segundos.

Monitoramento de Capacidade

URL: `/capacity`

Recursos

A visualização de Capacidade exibe o uso de mapas eBPF e limites de capacidade para todos os mapas no datapath do UPF.

Tabela de Uso de Mapas eBPF

Tabela de todos os mapas eBPF com informações de uso:

Coluna	Descrição
Nome do Mapa	Nome do mapa eBPF (por exemplo, `uplink_pdr_map`, `far_map`)
Usado	Número de entradas atualmente no mapa
Capacidade	Máximo de entradas permitidas no mapa
Uso	Barra de progresso visual com porcentagem
Tamanho da Chave	Tamanho das chaves do mapa em bytes
Tamanho do Valor	Tamanho dos valores do mapa em bytes

Indicadores de Uso Coloridos

A barra de progresso de uso é colorida com base na utilização:

- **Verde (<50%)**: Operação normal, capacidade ampla
- **Amarelo (50-70%)**: Cuidado, monitorar crescimento
- **Âmbar (70-90%)**: Aviso, planejar aumento de capacidade
- **Vermelho (>90%)**: Crítico, ação imediata necessária

Mapas Críticos para Monitorar

uplink_pdr_map:

- Armazena PDRs de uplink indexados por TEID
- Uma entrada por fluxo de tráfego uplink
- **Crítico**: 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**: 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. Monitorar tendências de uso de mapas ao longo do tempo
2. Identificar quais mapas estão crescendo mais rápido
3. Planejar aumentos de capacidade antes de atingir limites

Prevenir Falhas no Estabelecimento de Sess��es:

1. Verificar uso do mapa PDR antes de um aumento de tráfego esperado
2. Aumentar a capacidade do mapa se estiver se aproximando dos limites
3. Monitorar após o aumento de capacidade para verificar

Solução de Problemas de Falhas de Sessão:

1. Quando o estabelecimento de sessão falhar, verificar a visualização de Capacidade
2. Se os mapas PDR estiverem vermelhos (>90%), a capacidade está esgotada
3. Aumentar a capacidade do mapa ou limpar sessões obsoletas

Otimizar Configuração do Mapa:

1. Revisar tamanhos de chave e valor
2. Calcular uso de memória por mapa
3. Otimizar 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 de runtime do dataplane ativos:

- **Endereço N3 Ativo**: Vínculo da interface N3 em runtime
- **Endereço N9 Ativo**: Vínculo da interface N9 em runtime (se habilitado)

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. Verificar se o IP da interface N3 corresponde à configuração do gNB
2. Confirmar se a interface N6 pode rotear para a rede de dados
3. Confirmar se a interface PFCP é acessível a partir do SMF

Solução de Problemas de Problemas de Interface:

1. Comparar configuração estática com endereços ativos do dataplane
2. Verificar se as interfaces estão vinculadas corretamente
3. Verificar se houve alterações na configuração da interface

Documentação e Auditoria:

1. Registrar a configuração do UPF para documentação
2. Verificar se a implantação corresponde às especificações de design
3. Auditar atribuições de interface

Visualização de Rotas

URL: `/routes`

Recursos

A visualização de Rotas fornece monitoramento abrangente das rotas IP do Equipamento do Usuário (UE) e sessões de protocolo de roteamento (OSPF e BGP).

Visão Geral do Status da Rota

O painel exibe estatísticas agregadas de rotas:

- **Status**: Roteamento habilitado ou desabilitado
- **Total de Rotas**: Número total de rotas IP de UE
- **Sincronizado**: Número de rotas sincronizadas com sucesso
- **Falhou**: Número de rotas que falharam ao sincronizar

Rotas IP Ativas de UE

Tabela exibindo todas as rotas IP ativas do Equipamento do Usuário:

Coluna	Descrição
Índice	Número do índice da rota
Endereço IP da UE	Endereço IPv4 ou IPv6 atribuído à UE

Propósito:

- Ver todas as 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
ID do Vizinhos	Identificador do roteador OSPF
Endereço	Endereço IP do vizinho OSPF
Interface	Interface usada para adjacência OSPF
Estado	Estado de adjacência OSPF (Completo, Inicial, etc.)
Prioridade	Valor de prioridade OSPF
Tempo Ativo	Duração que o vizinho está ativo
Tempo Morto	Tempo até que o vizinho seja considerado morto

Estados OSPF:

- **Completo** (verde): Totalmente adjacente e trocando informações de roteamento
- **Outros estados** (amarelo): Adjacência formando ou incompleta

Pares BGP

Tabela de pares BGP (Border Gateway Protocol):

Coluna	Descrição
Endereço IP do Vizinhos	Endereço IP do par BGP
ASN	Número do Sistema Autônomo do par
Estado	Estado da sessão BGP (Estabelecido, Ocioso, etc.)
Ativo/Inativo	Duração do estado atual
Prefixos Recebidos	Número de prefixos de rota recebidos do par
Msg Enviadas	Total de mensagens BGP enviadas para o par
Msg Recebidas	Total de mensagens BGP recebidas do par

Estados BGP:

- **Estabelecido** (verde): Sessão BGP ativa, trocando rotas
- **Outros estados** (vermelho): Sessão inativa ou em estabelecimento

O cabeçalho também exibe o ID do Roteador BGP local e ASN quando o BGP está configurado.

Rotas Redistribuídas OSPF

Tabela mostrando LSAs Externas OSPF (Link State Advertisements) para rotas de UE redistribuídas:

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

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 rota
- Atualiza vizinhos OSPF, pares BGP e tabelas de rotas

Casos de Uso

Monitorar Saúde do Protocolo de Roteamento:

1. Navegar para a visualização de Rotas
2. Verificar estados dos vizinhos OSPF (deve ser "Completo")
3. Confirmar se os pares BGP estão "Estabelecidos"
4. Confirmar número esperado de vizinhos/pares

Verificar Distribuição de Rotas de UE:

1. Verificar tabela de Rotas IP Ativas para UE específica
2. Rolar para a seção de Rotas Redistribuídas OSPF
3. Verificar se a rota de UE aparece nas LSAs externas
4. Confirmar que o roteador anunciante corresponde ao UPF esperado

Solução de Problemas de Problemas de Sincronização de Rotas:

1. Verificar contadores Sincronizados vs. Falhados na visão geral de status
2. Se as rotas estão falhando, clicar no botão "Sincronizar Rotas"
3. Monitorar mensagens de erro na faixa vermelha se a sincronização falhar
4. Verificar mensagens de erro OSPF/BGP nas seções respectivas

Verificar Implantação Multi-UPF:

1. Selecionar diferentes instâncias do UPF no menu suspenso
2. Comparar contagens de rotas entre instâncias
3. Verificar se os vizinhos OSPF se veem
4. Verificar relacionamentos de peering BGP

Monitorar Escalonamento de Rotas:

1. Rastrear contagem total de rotas à medida que as sessões de UE aumentam
2. Verificar se as rotas estão sendo distribuídas para os protocolos de roteamento
3. Monitorar crescimento da contagem de LSA OSPF
4. Verificar 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 rotas de UE.

Integração de Roteamento

A visualização de Rotas se integra com o FRR (Free Range Routing) em execução no UPF:

- **OSPF**: Rotas são redistribuídas como LSAs Externas Tipo-2
- **BGP**: Rotas são anunciadas para pares BGP configurados
- **Mecanismo de Sincronização**: Chamadas da API REST acionam comandos vtysh para atualizar o FRR

Visualização de Capacidades XDP

URL: `/xdp_capabilities`

Recursos

A visualização de Capacidades XDP exibe suporte ao modo eXpress Data Path (XDP), capacidades de desempenho e cálculos de throughput para o dataplane do UPF.

Configuração da Interface

Exibe informações sobre a interface de rede e o driver:

Campo	Descrição
Nome da Interface	Interface de rede usada para XDP (por exemplo, eth0, ens1f0)
Driver	Nome do driver de rede (por exemplo, i40e, ixgbe, virtio_net)
Versão do Driver	String da versão do driver
Modo Atual	Modo XDP ativo (DRV, SKB ou NENHUM)
Contagem de Múltiplas Filas	Número de pares de filas NIC para processamento paralelo

Modos XDP

A visualização exibe todos os modos XDP com seu status de suporte e características de desempenho:

XDP_DRV (Modo do Driver):

- **Desempenho**: ~5-10 Mpps (milhões de pacotes por segundo)
- **Descrição**: Suporte nativo ao XDP no driver, maior desempenho
- **Requer**: Driver NIC com suporte nativo ao XDP (i40e, ixgbe, mlx5, etc.)
- **Status**: Suportado se o driver tiver ganchos XDP
- **Indicador**: Marca de verificação verde (✓) se suportado, X vermelho (✗) se

não

XDP_SKB (Modo Genérico):

- **Desempenho**: ~1-2 Mpps
- **Descrição**: Modo de fallback usando a pilha de rede do kernel
- **Requer**: Qualquer interface de rede
- **Status**: Sempre suportado
- **Indicador**: Marca de verificação verde (✓)

Indicador de Modo Atual:

- Ponto azul ao lado do modo XDP atualmente ativo
- Mostra qual modo está realmente em uso

Razões para Modos Não Suportados:

- Se um modo não for suportado, o campo "Razão" explica o porquê
- Razões comuns: driver não possui suporte ao XDP, incompatibilidade do tipo de interface

Visualização de Capacidades XDP mostrando configuração da interface, modos suportados e o calculador interativo de throughput em 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 em 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**: Controle de plano 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 em taxa de fio incluindo todos os cabeçalhos
- Fórmula: `Gbps = Mpps × Packet_Size × 8 / 1000`
- Inclui cabeçalhos GTP, UDP, IP e Ethernet

Taxa de Dados do Usuário (Gbps):

- Throughput real da carga útil do usuário
- Exclui ~50 bytes de sobrecarga de encapsulação GTP
- Fórmula: `Gbps = Mpps × (Packet_Size - 50) / 1000`

Taxa de Pacotes:

- Exibe Mpps e pacotes/segundo com separador de milhares
- Exemplo: 10 Mpps = 10.000.000 pacotes/segundo

Exibição da Fórmula:

- Mostra a decomposição do cálculo passo a passo

- Exemplo: `10 Mpps × 1200 bytes × 8 bits/byte ÷ 1000 = 96 Gbps`

Compreendendo Mpps

A visualização inclui uma seção de explicação cobrindo:

O que é Mpps:

- Milhões de Pacotes por Segundo
- Métrica chave para desempenho de processamento de pacotes
- Independente do tamanho do pacote

Relação com Throughput:

- Mesma Mpps com pacotes maiores = maior Gbps
- Mesma Mpps com pacotes menores = menor Gbps
- Throughput depende da taxa e 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. Navegar para a visualização de Capacidades XDP
2. Verificar modo XDP atual (deve ser DRV para melhor desempenho)
3. Anotar a faixa de desempenho em Mpps
4. Revisar banner de recomendação

Calcular Throughput Esperado:

1. Inserir taxa de pacotes esperada em Mpps
2. Inserir tamanho médio do pacote para seu perfil de tráfego
3. Revisar throughput calculado em Gbps
4. Comparar com capacidade de link ou requisitos de desempenho

Otimizar Configuração do XDP:

1. Verificar se o modo XDP_DRV é suportado, mas não ativo
2. Revisar versão do driver e compatibilidade
3. Seguir recomendação para atualizar para o modo do driver, se disponível
4. Verificar se a contagem de múltiplas filas corresponde aos núcleos da CPU

Planejamento de Capacidade:

1. Usar a calculadora para determinar Mpps necessário para throughput alvo
2. Comparar com as capacidades do modo XDP atual
3. Determinar se é necessária atualização de hardware
4. Planejar seleção de interface e driver para novas implantações

Solução de Problemas de Problemas de Desempenho:

1. Verificar se o modo XDP é DRV, não SKB
2. Verificar versão do driver para problemas de desempenho conhecidos
3. Verificar se a contagem de múltiplas filas é suficiente
4. Calcular se o modo atual suporta o throughput necessário

Dicas de Otimização de Desempenho

Modo do Driver (XDP_DRV):

- Usar NICs com suporte nativo ao XDP (Intel i40e/ixgbe, Mellanox mlx5)
- Atualizar drivers de NIC para a versão mais recente
- Ativar múltiplas filas (RSS) para processamento paralelo
- Ajustar tamanhos de buffer de fila da NIC

Modo Genérico (XDP_SKB):

- Aceitável para desenvolvimento e testes
- Não recomendado para produção de alto throughput
- Considerar atualização de hardware para implantações de produção

Configuração de Múltiplas Filas:

- 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 do modo.

Visualizador de Logs

URL: `/logs`

Recursos

Visualizar 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. Abrir a visualização de Logs
2. Iniciar o estabelecimento de sessão a partir do SMF
3. Assistir aos logs de mensagens PFCP e quaisquer erros

Monitorar Comunicação PFCP:

1. Visualizar mensagens de configuração de associação PFCP
2. Rastrear criação/modificação/exclusão de sessão
3. Verificar mensagens de heartbeat

Depurar Problemas de Encaminhamento:

1. Procurar erros de processamento de pacotes
2. Verificar logs de operação do mapa eBPF
3. Identificar problemas de configuração de FAR/PDR

Melhores Práticas

Diretrizes Operacionais

Monitoramento:

- Verificar regularmente a visualização de Capacidade para prevenir exaustão de mapas
- Monitorar Estatísticas para padrões de tráfego incomuns ou descartes
- Rastrear crescimento da contagem de sessões ao longo do tempo

- Observar erros de processamento do XDP

Gerenciamento de Buffers:

- Monitorar buffers durante cenários de transferência
- Limpar buffers presos se os pacotes ultrapassarem o TTL
- Verificar se o buffering está desativado após a conclusão da transferência
- Usar "Limpar" em vez de "Limpar Tudo" para evitar perda de pacotes

Gerenciamento de Sessões:

- Usar filtros para localizar rapidamente sessões específicas de UE
- Expandir sessões para verificar configuração da regra
- Comparar sessões entre várias instâncias do UPF
- Verificar indicador de saúde antes de solucionar problemas

Solução de Problemas:

- Usar Logs para depuração em tempo real
- Verificar visualização de Sessões para confirmar conectividade de UE
- Verificar configuração de Regras para fluxos de tráfego
- Monitorar Estatísticas para descartes de pacotes ou erros de encaminhamento

Desempenho

- A atualização automática do painel é de 5-10 segundos dependendo da visualização
- Listas de sessões grandes podem levar tempo para carregar
- A visualização de Regras filtra por entradas ativas (volumes diferentes de zero para URRs)
- Operações de buffer são executadas imediatamente no UPF selecionado

Documentação Relacionada

- **Guia de Operações PFCP** - Gerenciamento de sessões PFCP e detalhes do protocolo
- **Guia de Gerenciamento de Regras** - Configuração de PDR, FAR, QER, URR
- **Guia de Monitoramento** - Estatísticas, métricas e planejamento de capacidade
- **Guia de Rotas** - Detalhes sobre roteamento de UE e integração FRR
- **Guia de Modos XDP** - Documentação detalhada sobre modos XDP e informações sobre eBPF
- **Guia de Solução de Problemas** - Problemas comuns e diagnósticos
- **Guia de Operações UPF** - Operações gerais do UPF e arquitetura

Modos de Anexação XDP para OmniUPF

Índice

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 de nível de produção.

Comparação de Modos XDP

Aspecto	Modo Genérico	Modo Nativo	Modo Offload
Ponto de Anexação	Pilha de rede Linux	Driver de rede	Hardware NIC
Desempenho	~1-2 Mpps	~5-10 Mpps	~10-40 Mpps
Latência	~100 µs	~10 µs	~1 µs
Uso de CPU	Alto	Médio	Baixo
Requisitos de	Qualquer NIC	Driver compatível	SmartNIC com suporte

Aspecto	Modo Genérico	Modo Nativo	Modo Offload
NIC		com XDP	a XDP
Suporte a Hipervisores	Todos os hipervisores	A maioria (requer multi-queue)	Raro (PCI passthrough)
Caso de Uso	Testes, desenvolvimento	**Produção (recomendado)**	Sites de borda de alto throughput
Configuração	`xdp_attach_mode: generic`	`xdp_attach_mode: native`	`xdp_attach_mode: offload`

Recomendação: Use **modo nativo** para implantações de 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

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

Aviso: O modo genérico **não é adequado para produção**. Ele irá criar gargalos em altas taxas de pacotes e desperdiçar 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 enquanto mantém 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 de produção** (recomendado)
- **Redes de grau de transportadora** que requerem alto throughput
- **Cenários de computação de 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: `ixgbevf`, `i40evf` (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 XDP offload executa o programa eBPF **diretamente no hardware da NIC** (SmartNIC), contornando completamente a CPU para o processamento de pacotes. Isso fornece o mais alto desempenho, 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: Entendendo 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 vários 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. **Editar Dispositivo de Rede**
    - Vá para a aba **Hardware**
    - Clique no seu dispositivo de rede (por exemplo, `net0`)
    - Clique em **Editar**
3. **Definir Multiqueue**
    - Encontre o campo **"Multiqueue"**
    - Defina para **8** (ou corresponda à sua contagem de vCPU, máximo 16)
    - Clique em **OK**
4. **Iniciar a VM**
    - Clique em **Iniciar**

Opção B: Através da Linha de Comando do Proxmox

```
# SSH no seu host Proxmox

# Encontre seu ID da 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 VMs)
- **16 filas**: Máximo para alto desempenho (8+ vCPU VMs)

Passo 3: Verificar Multi-Queue Dentro da VM

Após a reinicialização da VM, faça SSH na VM e verifique:

```
# Verifique a configuração da fila
ethtool -l eth0

# Saída esperada:
# Parâmetros de canal para eth0:
# Combinado:        8         <-- Deve corresponder ao seu valor
configurado

# Conte as filas reais
ls -1d /sys/class/net/eth0/queues/rx-* | wc -l
ls -1d /sys/class/net/eth0/queues/tx-* | wc -l

# Ambos devem mostrar 8 (ou seu valor configurado)
```

Passo 4: Habilitar XDP Nativo no OmniUPF

Edite a configuração do OmniUPF:

```
# Edite o arquivo de configuração
sudo nano /etc/eupf/config.yaml
```

Altere o modo XDP:

```
# Antes
xdp_attach_mode: generic
```

```
# Depois
xdp_attach_mode: native
```

Reinicie o OmniUPF:

```
sudo systemctl restart eupf
```

Passo 5: Verificar se o XDP Nativo Está Ativo

Verifique os logs:

```
# Veja os logs de inicialização
journalctl -u eupf --since "1 minute ago" | grep -i "xdp\|attach"

# Saída esperada:
# xdp_attach_mode:native
# XDPAttachMode:native
# Programa XDP anexado à iface "eth0" (índice 2)
```

Verifique via API:

```
# Consulte a configuração
curl -s http://localhost:8080/api/v1/config | grep xdp_attach_mode

# Saída esperada:
# "xdp_attach_mode": "native",
```

Problemas Comuns no Proxmox

Problema: "Falha ao anexar o programa XDP"

Solução:

- Verifique se o multi-queue está habilitado (`ethtool -l eth0`)
- Verifique a versão do kernel: `uname -r` (deve ser ≥ 5.15)
- Certifique-se de que o driver VirtIO está carregado: `lsmod | grep virtio_net`

Problema: Apenas 1 fila apesar da configuração

Solução:

- A VM deve ser **totalmente desligada** (não reiniciada) para mudanças de 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 de CPU deve se espalhar entre os núcleos
- Verifique as estatísticas do XDP: `curl http://localhost:8080/api/v1/xdp_stats`

Habilitando XDP Nativo em Outros Hipervisores

VMware ESXi / vSphere

O VMware utiliza o driver `vmxnet3`, que suporta XDP nativo.

Requisitos:

- ESXi 6.7 ou posterior
- Versão do driver vmxnet3 1.4.16+ na VM
- Versão de hardware da VM 14 ou posterior

Habilitar Multi-Queue:

1. **Desligue a VM**
2. **Editar configurações da VM**:
   - Clique com o botão direito na VM → Editar Configurações
   - Adaptador de Rede → Avançado
   - Defina **Receive Side Scaling** como **Habilitado**
3. **Editar arquivo .vmx** (opcional, para mais filas):

   ```
   ethernet0.pnicFeatures = "4"
   ethernet0.multiqueue = "8"
   ```

4. **Iniciar VM e verificar**:

   ```
   ethtool -l ens192  # Verifique a contagem de filas
   ```

Configurar OmniUPF:

```yaml
interface_name: [ens192]  # O VMware geralmente usa ens192
xdp_attach_mode: native
```

KVM / libvirt (Raw)

Habilitar Multi-Queue via virsh:

```
# Editar configuração da VM
virsh edit seu-nome-vm
```

Adicione à seção de 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 utiliza o driver `hv_netvsc`, que suporta XDP nativo.

Requisitos:

- Windows Server 2016 ou posterior
- Linux Integration Services 4.3+ na VM
- VM de Geração 2

Habilitar Multi-Queue:

PowerShell no host Hyper-V:

```
# Definir VMQ (Virtual Machine Queue) - multi-queue do Hyper-V
Set-VMNetworkAdapter -VMName "SuaVM" -VrssEnabled $true -VmmqEnabled
$true
```

Configurar OmniUPF:

```
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 hooks de XDP.

Solução alternativa: Use apenas o modo genérico:

```
xdp_attach_mode: generic  # Única opção para VirtualBox
```

Verificando o Modo XDP

Após configurar o XDP nativo, verifique se está funcionando corretamente:

1. Verifique os Logs do OmniUPF

```
# Veja os logs recentes
journalctl -u eupf --since "5 minutes ago" | grep -i xdp

# Procure por:
# ✓ "xdp_attach_mode:native"
# ✓ "Programa XDP anexado à iface"
# ✗ "Falha ao anexar" ou "retornando ao genérico"
```

2. Verifique via API

```
# Consulte 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

# Exemplo de saída:
{
  "xdp_aborted": 0,      # Deve ser 0 (erros)
  "xdp_drop": 1234,      # Pacotes descartados
  "xdp_pass": 5678,      # Passados para a pilha
  "xdp_redirect": 9012,  # Pacotes redirecionados
  "xdp_tx": 3456         # Pacotes transmitidos
}
```

4. Verifique o Suporte do Driver

```
# Verifique se o driver suporta XDP
ethtool -i eth0 | grep driver

# Para Proxmox/KVM: Deve mostrar "virtio_net"
```

```
# Para VMware: Deve mostrar "vmxnet3"
# Para Hyper-V: Deve mostrar "hv_netvsc"
```

5. Teste de Desempenho

Compare o processamento de pacotes antes e depois:

```
# Monitore a taxa de pacotes
watch -n 1 'curl -s http://localhost:8080/api/v1/packet_stats | jq
.rx_packets'

# Modo genérico: ~1-2 Mpps
# Modo nativo: ~5-10 Mpps (melhoria de 5-10x)
```

Resolvendo Problemas 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 existem programas XDP existentes**:

   ```
   ip link show eth0 | grep xdp

   # Se outro programa XDP estiver anexado, descarte-o primeiro
   ip link set dev eth0 xdp off
   ```

Solução:

- Atualize o kernel para 5.15+ se for mais antigo

- Certifique-se de que o driver virtio_net 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 de driver:

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

Causas Comuns:

1. **Driver não suporta XDP nativo**:
    - Drivers do VirtualBox (sem suporte a XDP nativo)
    - Drivers de NIC mais antigos
2. **Multi-queue não habilitado**:
    - Verifique: `ethtool -l eth0`
    - Deve mostrar > 1 fila combinada
3. **Suporte ao XDP no kernel desabilitado**:

```
# Verifique se o XDP está habilitado no kernel
grep XDP /boot/config-$(uname -r)

# Deve mostrar:
# CONFIG_XDP_SOCKETS=y
# CONFIG_BPF=y
```

Solução:

- Habilite multi-queue (veja a seção do Proxmox)
- Atualize para um driver suportado
- Recompile o kernel com suporte a XDP, se necessário

Problema: Desempenho Não Melhora com o Modo Nativo

Sintomas: Modo nativo habilitado, mas a taxa de pacotes é a mesma que o modo

genérico

Diagnóstico:

1. **Verifique a distribuição do multi-queue**:

   ```
   # Verifique as estatísticas por fila
   ethtool -S eth0 | grep rx_queue

   # O tráfego deve estar distribuído entre várias filas
   ```

2. **Verifique a utilização da CPU**:

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

   # Deve ver carga distribuída entre várias 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 ao eBPF no kernel Linux está completo

---

Problema: Multi-Queue Não Funciona no Proxmox

Sintomas: `ethtool -l eth0` mostra apenas 1 fila apesar da configuração

Diagnóstico:

1. **Verifique a configuração da VM do Proxmox**:

```
# No host Proxmox
grep net0 /etc/pve/qemu-server/YOUR_VM_ID.conf

# Deve mostrar: queues=8
```

2. **Verifique se a VM foi totalmente desligada**:

```
# No host Proxmox
qm status YOUR_VM_ID

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

Solução:

```
# No host Proxmox
# Forçar desligamento e reiniciar
qm shutdown YOUR_VM_ID
sleep 10
qm start YOUR_VM_ID

# Depois verifique dentro da VM
ethtool -l eth0
```

Importante: Mudanças na contagem de filas requerem um **desligamento total 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:

Operações XDP requerem capacidades `CAP_NET_ADMIN` e `CAP_SYS_ADMIN`.

Solução:

1. **Execute o OmniUPF como root** (ou com capacidades):

   ```
   sudo systemctl restart eupf
   ```

2. **Se estiver usando systemd**, verifique se o arquivo de serviço tem capacidades:

   ```
   # /lib/systemd/system/eupf.service
   [Service]
   CapabilityBoundingSet=CAP_NET_ADMIN CAP_SYS_ADMIN CAP_NET_RAW
   AmbientCapabilities=CAP_NET_ADMIN CAP_SYS_ADMIN CAP_NET_RAW
   ```

3. **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 processamento de pacotes do OmniUPF:

Cenário	Modo Genérico	Modo Nativo	Melhoria
Taxa de Pacotes	1.5 Mpps	8.2 Mpps	**5.5x mais rápido**
Latência	95 µs	12 µs	**8x menor**
Uso de CPU (1 Gbps)	85% (1 núcleo)	15% (distribuído)	**5x mais eficiente**
Throughput Máximo	~1.2 Gbps	~10 Gbps	**8x maior**

Recomendação: Sempre use **modo nativo** com **multi-queue habilitado** para implantações de produção.

Recomendações de Hardware para XDP

⚠ IMPORTANTE: Antes de comprar qualquer hardware, consulte o suporte da Omnitouch para confirmar que é 100% compatível com sua configuração e requisitos de implantação específicos.

NICs Conhecidas que Suportam XDP Nativo

Essas NICs estão verificadas para suportar o modo XDP nativo com OmniUPF:

NICs Intel (Recomendadas para Bare Metal)

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

NICs Mellanox/NVIDIA (Alto Desempenho)

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

NICs Broadcom

Modelo	Velocidade	Driver	Suporte a XDP	Notas
Série BCM57xxx	10/25/50GbE	bnxt_en	Nativo ✓	Comum em servidores Dell/HP

NICs Virtuais (Implantações de VM)

Plataforma	Tipo de NIC	Driver	Suporte a XDP	Multi-Queue	Notas
Proxmox/ KVM	VirtIO	virtio_net	Nativo ✓	Sim (configurável)	**Melhor para VMs**
VMware ESXi	vmxnet3	vmxnet3	Nativo ✓	Sim	Requer ESXi 6.7+
Hyper-V	NIC Sintética	hv_netvsc	Nativo ✓	Sim	Windows Server 2016+
AWS	ENA	ena	Nativo ✓	Sim	Instâncias de metal EC2
VirtualBox	Qualquer	vários	Apenas genérico ⚠	Não	Não recomendado para produção

NICs com Suporte a Offload de Hardware

True XDP hardware offload (eBPF roda na NIC):

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

Essas configurações foram verificadas com OmniUPF em produção:

Opção de Orçamento (1-10 Gbps)

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

Faixa Média (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 isso para implantações de OmniUPF em produção:

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

Lista de Verificação Pré-Compra

Antes de comprar hardware, verifique:

1. 🔰 **Suporte 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 ≥ 5.15 para suporte confiável ao XDP

   ```
   uname -r
   ```

3. 🔰 **Multi-Queue**: Verifique se a NIC suporta várias 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 do XDP Nativo 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/eupf/config.yaml  # Defina: xdp_attach_mode: native
sudo systemctl restart eupf
journalctl -u eupf --since "1 min ago" | grep xdp  # Verifique o modo nativo
```

Verifique se o Modo XDP Está Ativo

```
# Verifique a configuração
curl -s http://localhost:8080/api/v1/config | grep xdp_attach_mode

# Verifique as estatísticas
```

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

# Verifique as filas
ethtool -l eth0
```

Documentação da API OmniUPF

Visão Geral

A API OmniUPF fornece uma interface RESTful para gerenciar e monitorar a Função de Plano do Usuário baseada em eBPF. A API permite controle em tempo real e observabilidade de:

- **Sessões PFCP**: Gerenciamento do ciclo de vida e associação de sessões
- **Regras de Detecção de Pacotes (PDR)**: Classificação de tráfego para uplink e downlink (IPv4 e IPv6)
- **Regras de Ação de Encaminhamento (FAR)**: Ações de encaminhamento, armazenamento e descarte de pacotes
- **Regras de Aplicação de QoS (QER)**: Políticas de Qualidade de Serviço e limitação de taxa
- **Regras de Relatório de Uso (URR)**: Rastreamento e relatório de volume de dados
- **Buffers de Pacotes**: Funcionalidade de armazenamento e reprodução de pacotes
- **Estatísticas**: Métricas em tempo real para pacotes, rotas, XDP e interfaces N3/N6
- **Gerenciamento de Rotas**: Sincronização de rotas de UE com o daemon de roteamento FRR
- **Configuração**: Gerenciamento de configuração do UPF e do plano de dados

Documentação da API Swagger

A API está totalmente documentada usando a especificação **OpenAPI 3.0 (Swagger)**. A interface interativa do Swagger UI fornece:

- Documentação completa dos endpoints com esquemas de requisição/resposta
- Funcionalidade de teste para chamadas de API diretamente do navegador
- Definições de esquema para todos os modelos de dados
- Códigos de status HTTP e respostas de erro

Interface interativa do Swagger UI mostrando os endpoints da API OmniUPF com documentação detalhada.

Acessando o Swagger UI

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
```

```
## Veja Também

- [Documentação de Gerenciamento de Rotas de UE](./routes.md) - Guia detalhado sobre integração FRR e sincronização de rotas
- [Guia de Operações](../OPERATIONS.md) - Operações e monitoramento da interface web
- [Swagger UI](http://10.98.0.20:8080/swagger/index.html) - Documentação interativa da API
```

Gerenciamento de Rota UE

Documentação Relacionada:

- Documentação da API - Referência completa da API incluindo endpoints de gerenciamento de rota
- Guia de Operações - Operações e monitoramento da interface web

Visão Geral

O UPF (Função de Plano do Usuário) integra-se ao **FRR (Free Range Routing)** para gerenciar dinamicamente as rotas IP do Equipamento do Usuário (UE). Essa integração garante que, à medida que as sessões de UE são estabelecidas ou encerradas, a infraestrutura de roteamento se 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 ativado, o sistema de sincronização de rotas:

1. **Monitora Sessões de UE**: Rastreia todas as sessões PFCP ativas e seus endereços IP de UE associados
2. **Mantém Lista de Rotas**: Mantém uma lista atualizada de rotas que precisam estar na tabela de roteamento
3. **Sincroniza com o FRR**: Envia automaticamente atualizações de rotas para o daemon FRR via sua API
4. **Gerencia Falhas**: Rastreia o status de sincronização (sincronizado/falhou)

para cada rota e tenta novamente conforme necessário

Configuração do FRR

Configuração

O FRR é implantado e configurado usando **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 kernel
 network {{
hostvars[inventory_hostname]['ansible_default_ipv4']['network'] }}/{{
mask_cidr }} area 0
 area 0 authentication message-digest
!
line vty
!
end
```

Modelo de Implantação:

1. **Definir Uma Vez**: Crie o modelo Jinja2 do FRR em seu papel Ansible (por exemplo, `roles/frr/templates/frr.conf.j2`)
2. **Configurar Parâmetros**: Defina variáveis em seu inventário Ansible para cada host UPF
3. **Implantar 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 de área, métodos de autenticação, anúncios de rede
- **Configuração BGP**: ASN, relacionamentos de vizinhança, políticas de rota, comunidades
- **Redistribuição de Rotas**: Quais rotas do kernel redistribuir (por exemplo, `redistribute kernel`)
- **Filtragem de Rotas**: Mapas de rotas, listas de prefixos, listas de acesso
- **Configurações de Interface**: Parâmetros de interface OSPF/BGP

Integração do 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** à tabela de roteamento do kernel com base nas sessões PFCP ativas. Essas rotas são então:

1. **Instaladas na tabela de roteamento do kernel** pelo mecanismo de sincronização de rotas do UPF
2. **Capturadas pelo FRR** via a diretiva `redistribute kernel`
3. **Anunciadas para protocolos de roteamento** (OSPF, BGP) de acordo com sua configuração do FRR
4. **Propagadas para a rede** para que o tráfego do UE possa ser roteado para esta instância de UPF

Pontos Chave:

- **Definir Uma Vez, Implantar 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 a 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 de IP /32 do UE com base em suas sessões PFCP ativas
- **Anúncio automático de rotas**: O FRR em cada UPF redistribui automaticamente as rotas locais de UE de acordo com suas políticas configuradas
- **Gerenciamento centralizado**: Atualize o 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 com a Interface Web

O Painel de Controle do UPF fornece uma página de **Rotas** que exibe:

- **Status da Rota**: Se a sincronização de rotas está habilitada ou desabilitada
- **Total de Rotas**: Número de endereços IP de UE sendo rastreados
- **Estatísticas de Sincronização**: Contagem de rotas sincronizadas com sucesso e quaisquer falhas
- **Rotas Ativas**: Lista em tempo real de todos os endereços IP de UE atualmente na tabela de roteamento
- **Vizinhos OSPF**: Status ao vivo das adjacências OSPF com detalhes dos vizinhos
- **Pares BGP**: Status da sessão BGP e estatísticas de prefixo (quando configurado)
- **Rotas Redistribuídas OSPF**: Visão completa dos LSAs externos mostrando como as rotas de UE são anunciadas

A página de Rotas fornece visibilidade abrangente sobre a sincronização de rotas de UE, vizinhos de protocolos de roteamento e anúncios de rotas redistribuídas.

Operação de Sincronização Manual

Os administradores podem acionar uma sincronização manual de rotas através da interface web usando o botão **Sincronizar Rotas**. Esta operação:

1. Lê novamente a lista atual de sessões de UE ativas do UPF
2. Compara com a tabela de roteamento do FRR
3. Adiciona quaisquer rotas ausentes
4. Remove quaisquer rotas obsoletas
5. Retorna estatísticas de sincronização atualizadas

Fluxo de Rotas

Benefícios

- **Provisionamento Sem Toque**: As rotas são gerenciadas automaticamente sem intervenção manual
- **Adaptação Dinâmica**: O roteamento da rede se adapta em tempo real à mobilidade do UE e mudanças nas sessões
- **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 rastreadas sem sincronização
- `POST /api/v1/routes/sync` - Sincroniza rotas com o FRR e retorna lista atualizada
- `GET /api/v1/route_stats` - Obtém estatísticas detalhadas de roteamento
- `GET /api/v1/routing/sessions` - Obtém sessões de protocolo de roteamento (vizinhos OSPF, pares BGP)
- `GET /api/v1/ospf/database/external` - Obtém 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 lida com todos os detalhes da entrada da rota, incluindo:

- Prefixo/máscara de destino
- Gateway/próximo salto
- Vinculação de interface
- Métrica e distância administrativa

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 métrica tipo E2 (Tipo Externo 2)

Essa verificação confirma que:

1. O UPF está rastreando com sucesso o endereço IP do UE
2. O mecanismo de sincronização de rotas enviou a rota para o FRR
3. O FRR redistribuiu a rota no OSPF
4. Os vizinhos OSPF estão recebendo os anúncios de rota