Arquitetura do OmniTWAG

Visão Geral

OmniTWAG é um Gateway de Acesso WLAN Confiável (TWAG) conforme a Seção 16 do 3GPP TS 23.402. Ele conecta pontos de acesso WLAN confiáveis ao Núcleo de Pacote Evoluído (EPC), fornecendo autenticação, gerenciamento de sessão e tunelamento de plano de dados.

Índice

- Arquitetura de Componentes e Processos
- Árvore de Supervisão
- Referência de Componentes
- Modos de Operação
- Integração de AP (Túnel GRE)
- Comportamento do DHCP
- Autenticação
- Ciclo de Vida da Sessão (Roteado para Casa)
- Interfaces Diameter
- Resumo da Configuração
- Derivação de Chaves
- Páginas Relacionadas

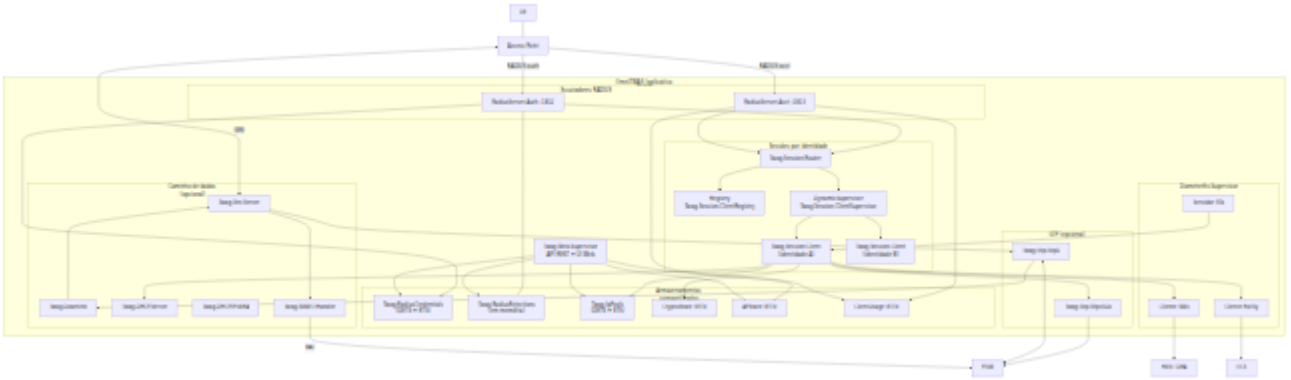
Arquitetura de Componentes e Processos

OmniTWAG é executado como um único aplicativo BEAM (Erlang VM). O supervisor de nível superior inicia um conjunto fixo de serviços compartilhados

e, com base na configuração, um conjunto de filhos de caminho de dados e interface opcionais. O estado por assinante vive em um GenServer para cada identidade, criado sob demanda.

Os principais grupos são:

- **Web e OAM**: a API REST e a interface web (`Twag.Web.Supervisor`).
- **Sessões por identidade**: um GenServer `Twag.Session.Client` para cada assinante, encontrado através de um `Registry` e iniciado por um `DynamicSupervisor`. `Twag.Session.Router` faz a busca ou criação.
- **Armazenamentos compartilhados**: `Twag.RadiusCredentials` (segredos compartilhados duráveis), `Twag.RadiusRejections` (fontes rejeitadas em memória), `Twag.IpPools` (pools de endereços), e os armazenamentos `CryptoState`, `APState` e `ClientUsage` suportados por ETS.
- **Escutadores RADIUS**: `RadiusServer.Auth` (porta 1812) e `RadiusServer.Acct` (porta 1813).
- **Diameter**: `DiameterEx.Supervisor` carrega os pares SWx (vetores de autenticação), STa (reauth e abortar), e Ro/Gy (cobrança online).
- **GTP**: `Twag.Gtp.GtpcS2a` (controle) e `Twag.Gtp.GtpU` (plano de usuário), iniciados apenas quando `gtp.enabled` é verdadeiro.
- **Caminho de dados**: `Twag.Gre.Server`, `Twag.Downlink`, `Twag.DHCP.Server`, `Twag.DHCP.IPv6RA`, e `Twag.NSWO.Handler`, cada um iniciado apenas quando seu bloco de configuração está habilitado.



Árvore de Supervisão

O supervisor raiz utiliza uma estratégia `:one_for_one` (máximo de 10 reinicializações em 60 segundos). Os filhos iniciam nesta ordem, de modo que

os segredos por IP de origem sejam resolvidos antes que o primeiro pacote RADIUS chegue:



Os filhos marcados com `*` são iniciados apenas quando seu bloco de configuração está habilitado (veja Resumo da Configuração).

Referência de Componentes

Componente	Tipo	Iniciado quando
`Twag.Web.Supervisor`	Supervisor	Sempre
`DiameterEx.Supervisor`	Supervisor	Sempre
`Twag.Session.ClientRegistry`	Registry	Sempre
`Twag.Session.ClientSupervisor`	DynamicSupervisor	Sempre
`Twag.Session.Router`	Módulo	Sob demanda
`Twag.Session.Client`	GenServer	Por identidade
`Twag.RadiusCredentials`	GenServer	Sempre

Componente	Tipo	Iniciado quando
`Twag.RadiusRejections`	GenServer	Sempre
`Twag.IpPools`	GenServer	Sempre
`RadiusServer.Auth`	GenServer	Sempre
`RadiusServer.Acct`	GenServer	Sempre
`CryptoState`	GenServer (ETS)	Sempre
`APState`	GenServer (ETS)	Sempre

Componente	Tipo	Iniciado quando
`ClientUsage`	GenServer (ETS)	Sempre
`Twag.Gtp.GtpcS2a`	GenServer	`gtp.enabled`
`Twag.Gtp.GtpU`	GenServer	`gtp.enabled`
`Twag.Downlink`	GenServer	`downlink.enabled`
`Twag.Gre.Server`	GenServer	`gre.enabled`
`Twag.DHCP.Server`	GenServer	`dhcp.enabled`

Componente	Tipo	Iniciado quando
`Twag.DHCP.IPv6RA`	GenServer	`ipv6_ra.enabled`
`Twag.NSWO.Handler`	GenServer	`nswo.enabled`

Modos de Operação

OmniTWAG suporta dois modos de plano de dados. O modo utilizado para uma sessão é resolvido a partir da credencial do AP correspondente (veja Segredos Compartilhados RADIUS), com a configuração global `gtp` / `nswo` como padrão.

Modo Roteado para Casa (Túnel GTP)

O tráfego é tunelado via GTP-C/GTP-U (interface S2a) para o PGW. O PGW aloca o endereço IP da UE, aplica políticas/cobrança e fornece acesso à internet.

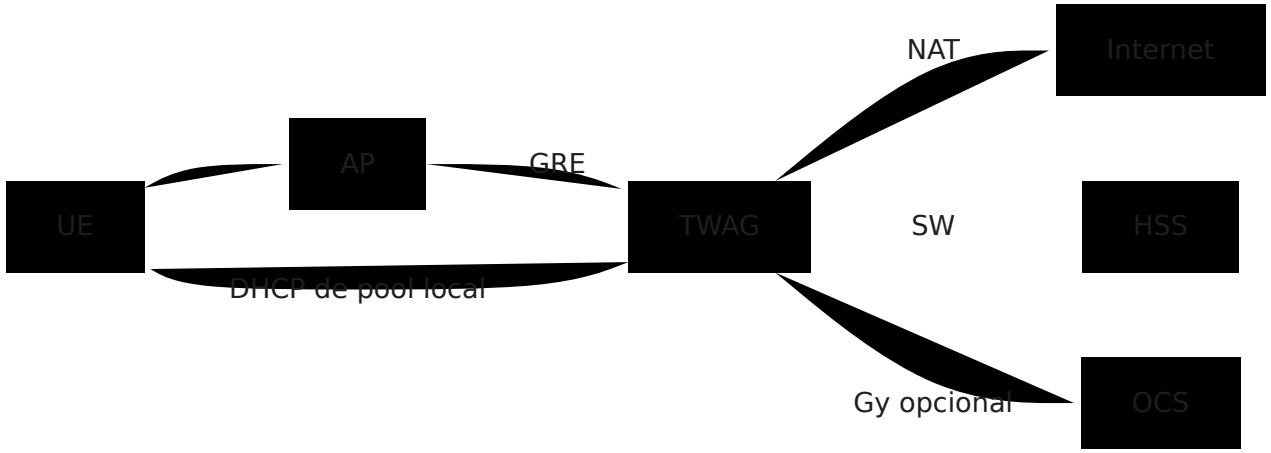


No thanks

- **IP da UE**: Alocado pelo PGW na Resposta de Criação de Sessão (IE PAA).
- **Entrega de IP**: O TWAG fornece DHCP para a UE via GRE, entregando o IP alocado pelo PGW.
- **Uplink**: O AP envia o tráfego da UE via GRE. O TWAG encapsula em GTP-U para o PGW.
- **Downlink**: O PGW envia via GTP-U. O TWAG decapsula e envia via GRE para o AP, depois para a UE.
- **Cobrança**: Tratada pelo PGW (PCEF) via Gy para o OCS.
- **Config**: `gtp.enabled: true`, `gre.enabled: true`.

Modo de Saída Local (NSWO)

O tráfego sai diretamente do TWAG para a internet. Nenhum PGW está envolvido. O TWAG realiza NAT/mascaramento e opcionalmente faz sua própria cobrança online via Gy.



- **IP da UE**: Alocado de um pool de IP local do TWAG (`Twag.IpPools`).
- **Entrega de IP**: O TWAG fornece DHCP para a UE via GRE, do pool.
- **Uplink**: O AP envia via GRE. O TWAG aplica NAT para a internet.
- **Downlink**: O caminho da internet retorna através do NAT do TWAG, depois GRE para o AP, depois para a UE.
- **Cobrança**: O TWAG atua como PCEF via Gy para o OCS (opcional, desativado por padrão). Veja Cobrança Online.
- **Config**: `nswo.enabled: true`, `gtp.enabled: false`.

A Saída Local é selecionada configurando `nswo.enabled: true` (com `gtp.enabled: false`), ou por AP pela credencial `breakout_type`. Quando habilitado, o manipulador NSWO é iniciado, as sessões recebem um endereço de um pool local em vez de um atribuído pelo PGW, e o uplink é NAT/mascarado diretamente para a internet. Veja Plano de Usuário e Caminho de Dados para os detalhes do caminho de dados.

Integração de AP (Túnel GRE)

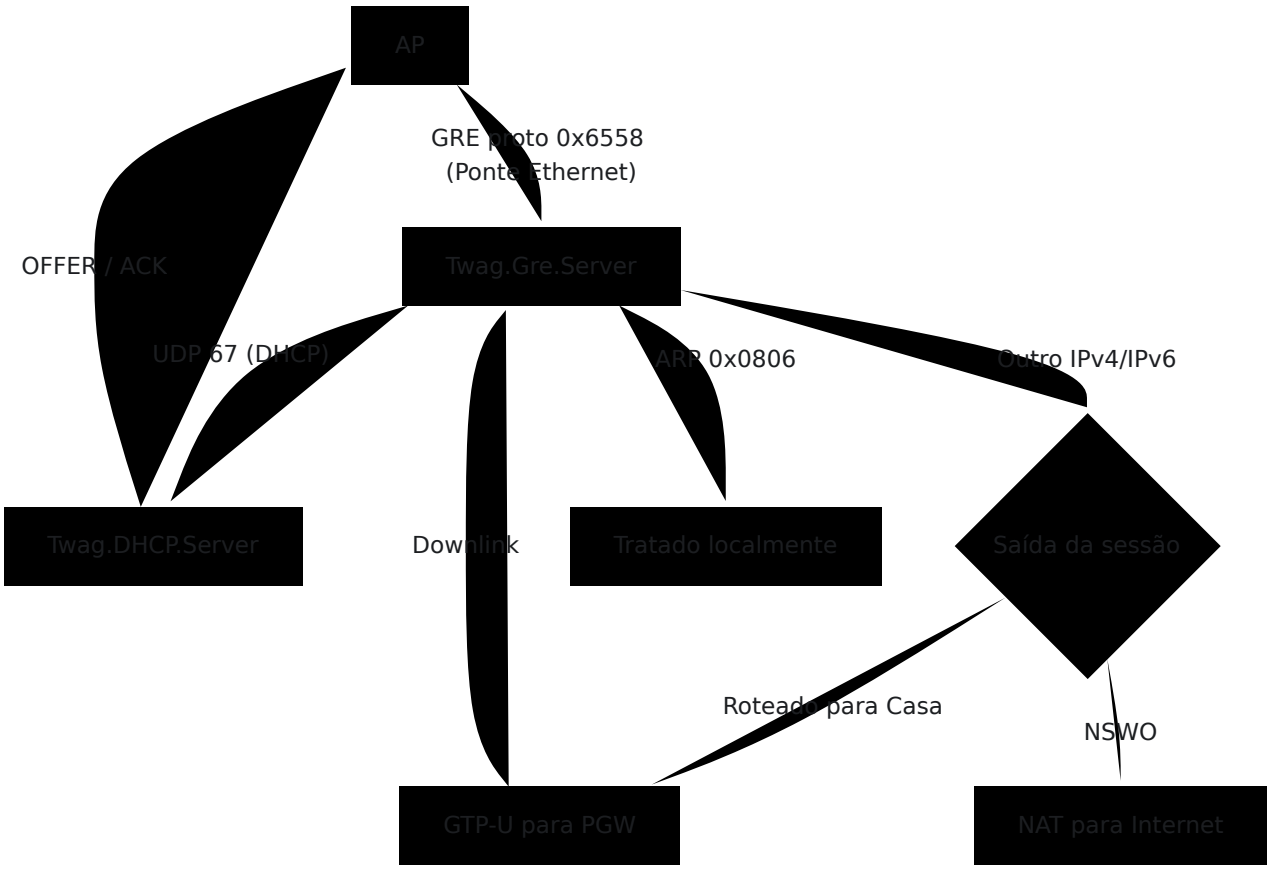
O AP tunela os quadros L2 da UE para o TWAG via GRE (protocolo IP 47). Isso fornece o link ponto a ponto necessário entre a UE e o TWAG conforme a Seção 16.1.2 do TS 23.402.

Configuração do AP Cisco

```
dot11 ssid ons
   tunnel CWAG
   authentication open eap eap_methods
   authentication key-management wpa
   accounting acct_methods

dot11 tunnel CWAG
    type gre
    source {AP_IP}
    destination {TWAG_IP}
    mss 1410
    mtu 1500
```

Fluxo de Dados GRE



Comportamento do DHCP

Modo	Fonte IP	Servidor DHCP	Notas
Roteado para Casa	PGW (PAA na Resposta de Criação de Sessão)	TWAG pré-aloca o IP do PGW, fornece via DHCP sobre GRE	DNS do PGW PCO
Saída Local	Pool local do TWAG (`Twag.IpPools`)	TWAG aloca do pool vinculado no DISCOVER (distribuição de pool)	DNS da configuração do pool

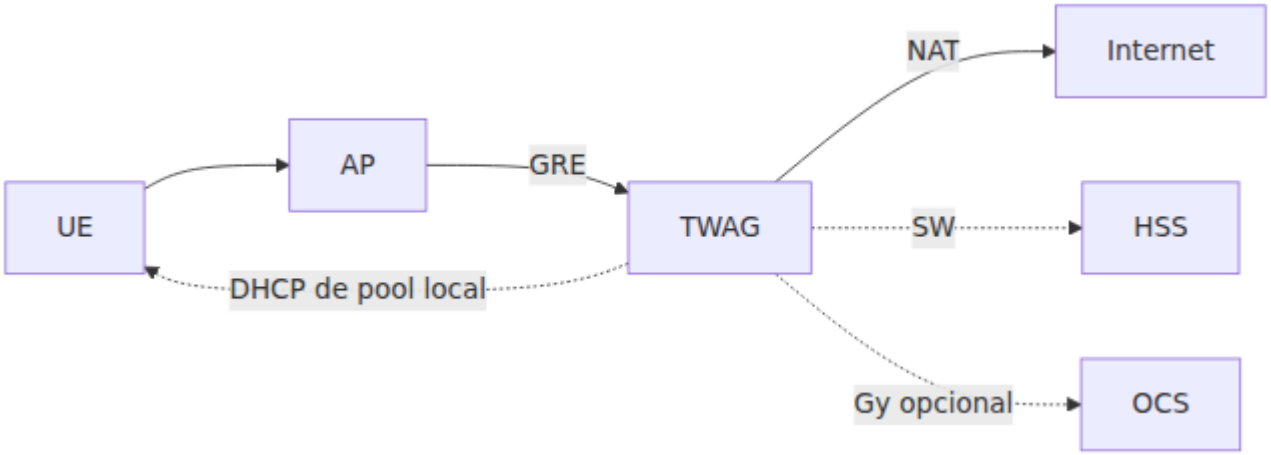
O servidor DHCP do TWAG:

1. Recebe um DHCP DISCOVER da UE via o túnel GRE.
2. Fornece a OFFER. No modo roteado para casa, este é o IP alocado pelo PGW que a sessão pré-alocou. No modo de saída local, quando o MAC está vinculado a um pool, o servidor distribui um endereço novo do pool.
3. Trata REQUEST/ACK para completar a troca.
4. Passa todo o tráfego DHCP através do túnel GRE para e do AP.

Autenticação

OmniTWAG atua como um TWAG combinado e Servidor AAA 3GPP, realizando autenticação EAP-AKA localmente e buscando vetores de autenticação do HSS via Diameter SWx. A máquina de estado EAP de cada assinante é executada dentro de seu próprio `Twag.Session.Client`.

Fluxo de Autenticação



Antes que qualquer pacote seja processado, o escutador RADIUS autoriza o IP de origem contra `Twag.RadiusCredentials`. Uma origem sem sub-rede de credencial correspondente é descartada e registrada como `unauthorized_source`. Uma origem que corresponde, mas falha na verificação do Message-Authenticator, é registrada como `bad_secret`. Veja Rejeições RADIUS.

Modos de Autenticação

Modo	Config	Fonte do Vetor	Processamento EAP
`:swx`	`auth_mode: :swx`	HSS via SWx MAR/MAA	Local (TWAG)
`:local`	`auth_mode: :local`	Chaves de teste locais (Milenage)	Local (TWAG)

Método EAP

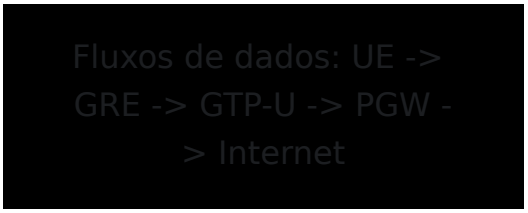
Configurável via `eap_method`. Padrão `:eap_aka`. Se a UE NAKs o método oferecido, o TWAG negocia para o método suportado preferido da UE. EAP-SIM e reautenticação EAP (ERP) também são suportados. Veja Autenticação.

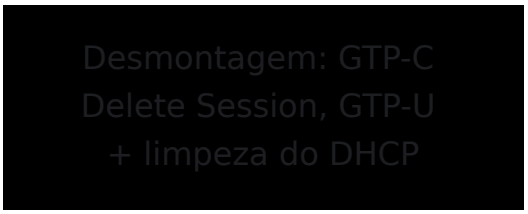
Ciclo de Vida da Sessão (Roteado

para Casa)


EAP Auth -> Access-Accept (MSK para AP)
AP envia Accounting-Start
Pós-aceitação em Twag.Session.Client
GTP-C Create Session Request -> PGW
PGW aloca IP, Resposta de Criação de Sessão
Registrar túnel GTP-U (TEIDs locais + remotos)
Pré-alocar IP do PGW no DHCP para MAC da UE
UE DHCP Discover -> OFFER com IP do PGW







Uma sessão é encerrada por uma das várias razões, cada uma registrada na métrica `twag.session.teardown`: `accounting_stop`, `admin_disconnect`, `network_disconnect` (STa ASR), `identity_timeout` (autenticação iniciada mas nunca completada), e `liveness_timeout`. Veja Referência de Métricas e Sessões.

Interfaces Diameter

Interface	ID do App	Direção	Propósito
SWx	16777265	TWAG -> HSS	Busca de vetor de autenticação (MAR/MAA)
STa	16777250	Entrada para TWAG	RAR (reauth), ASR (abortar), STR (terminar)
Ro/Gy	4	TWAG -> OCS	Cobrança online (apenas NSWO)

Roteamento DRA

O DRA deve rotear SWx para o HSS:

```
%{rule_name: "swx_to_hss", match: :all,
  filters: [{:application_id, 16777265}, {:avp, {296, "
{realm}"}}],
  route: %{peers: ["{hss_host}"]}}
```

Resumo da Configuração

Cada parâmetro é documentado na tabela que segue o bloco.

```
config :omnitwag,
  auth_mode: :swx,
  eap_method: :eap_aka,
  identity_timeout_ms: 600_000,

  gtp: %{
    enabled: true,
    pgw_ip: "10.179.1.21",
    local_ip: "10.179.1.41",
    pgw_port: 2123,
    gtpu_port: 2152,
    echo_enabled: true
  },

  gre: %{
    enabled: true,
    local_ip: "10.179.1.41"
  },

  downlink: %{
    enabled: true
  },

  dhcp: %{
    enabled: true,
    interface: "gre"
  },

  ipv6_ra: %{
    enabled: false
  },

  nswo: %{
    enabled: false,
    local_pool: "10.100.0.0/16",
    nat_interface: "eth0"
  },

  online_charging: %{
    mode: :disabled
  },

  rate_limit: %{
```

```
    max_failures: 10,
    window_ms: 60_000
  },

  prometheus: %{
    port: 9568
  }
```

Parâmetro	Padrão	Descrição
`auth_mode`	`:swx`	`:swx` busca vetores do HSS; `:local` usa chaves de teste Milenage locais.
`eap_method`	`:eap_aka`	Método EAP oferecido (`:eap_aka` ou `:eap_aka_prime`).
`identity_timeout_ms`	`600000`	Tempo que uma sessão pode permanecer na fase de identidade antes de ser encerrada (razão de desmontagem `identity_timeout`).
`gtp.enabled`	`true`	Inicia os filhos GTP-C/GTP-U (modo roteado para casa).
`gtp.pgw_ip`	`127.0.0.1`	Endereço do PGW para GTP-C/GTP-U.
`gtp.local_ip`	nenhum	Endereço local do TWAG para GTP.
`gtp.pgw_port`	`2123`	Porta GTP-C do PGW.
`gtp.gtpu_port`	`2152`	Porta GTP-U local.

Parâmetro	Padrão	Descrição
`gtp.echo_enabled`	`true`	Envia keepalives de eco GTP para o PGW.
`gre.enabled`	`false`	Inicia o servidor GRE para túneis de AP.
`gre.local_ip`	`0.0.0.0`	Endereço local ao qual o servidor GRE se vincula.
`downlink.enabled`	`true`	Inicia o encaminhador de downlink.
`dhcp.enabled`	`false`	Inicia o servidor DHCPv4.
`dhcp.interface`	nenhum	`"gre"` para modo túnel, ou um nome de NIC para entrega direta.
`ipv6_ra.enabled`	`false`	Inicia Anúncios de Roteador IPv6 e DHCPv6.
`nswo.enabled`	`false`	Inicia o manipulador de saída local NSWO.
`nswo.local_pool`	nenhum	CIDR de fallback para

Parâmetro	Padrão	Descrição
		endereçamento de saída local.
`nswo.nat_interface`	nenhum	NIC de saída para NAT/mascaramento.
`online_charging.mode`	`:disabled`	Modo Gy: `:disabled`, `:authorize_only`, ou `:online_charging`.
`rate_limit.max_failures`	`10`	Falhas de autenticação por NAS na janela antes que Access-Requests sejam descartados.
`rate_limit.window_ms`	`60000`	Janela de limite de taxa em milissegundos.
`prometheus.port`	`9568`	Porta de métricas Prometheus.
`radius_config.acct_interim_interval`	`300`	Intervalo de contabilidade provisória (segundos) anunciado ao AP em Access-Accept.

Derivação de Chaves

Para EAP-AKA', conforme o 3GPP TS 33.402 cláusula 6.2 e Anexo A.2, e RFC 5448:

Passo	Nó (`:swx` modo)	Resultado
1	TWAG	SWx MAR com `SIP-Authentication-Scheme` = `EAP-AKA'` e `ANID` = nome da rede
2	HSS	RAND, AUTN com bit de separação AMF 1, XRES
3	HSS	`CK'`, `IK'` = KDF(`CK`, `IK`, FC 0x20, nome da rede, SQN XOR AK)
4	TWAG	`MK` = PRF'(`IK'`, `CK'`, "EAP-AKA'", Identidade)
5	TWAG	`K_aut`, `MSK`, `EMSK` de `MK`

No modo `:local`, o TWAG executa também os passos 2 e 3, com as chaves de teste configuradas. Veja Autenticação para a derivação completa.

Para EAP-AKA (RFC 4187):

```
HSS fornece: RAND, AUTN, XRES, CK, IK
TWAG deriva: MK = SHA1(Identidade || IK || CK)
             K_encr, K_aut, MSK, EMSK de PRF(MK)
```

O nome da rede (ANID) é `WLAN` conforme a cláusula 8.1.1.2 do TS 24.302. Veja Nome da rede.

Páginas Relacionadas

- Autenticação
- Segredos Compartilhados RADIUS

- Rejeições RADIUS
- Pools de IP
- Sessões
- Contabilidade por AP
- Plano de Usuário e Caminho de Dados
- Cobrança Online
- Referência de Métricas

Visão Geral

Autenticação (EAP-AKA / EAP-AKA' / EAP-SIM)

Visão Geral

OmniTWAG autentica cada assinante com um método EAP que é executado sobre RADIUS entre o ponto de acesso e o TWAG. O TWAG atua como um servidor AAA combinado de TWAG e 3GPP. Ele executa a máquina de estados EAP e obtém os vetores de autenticação do HSS via Diameter SWx, ou de chaves de teste locais. Com sucesso, o TWAG entrega a Chave de Sessão Mestre (MSK) ao ponto de acesso como chaves MPPE dentro do RADIUS Access-Accept. O ponto de acesso então completa o handshake de 4 vias 802.11 com o assinante.

Para o contexto mais amplo do sistema, veja Arquitetura. Para como o segredo compartilhado do RADIUS é resolvido por ponto de acesso, veja Segredos Compartilhados do RADIUS. A autenticação é executada primeiro. A política de cobrança e separação por AP é aplicada após a autenticação ser bem-sucedida. Para a sessão autenticada e seu ciclo de vida de status, veja Sessões. Para a sessão GTP-C S2a que o TWAG cria após a autenticação, veja S2a GTP-C. Para a cobrança que segue uma autenticação bem-sucedida, veja Cobrança Online (Gy).

Índice

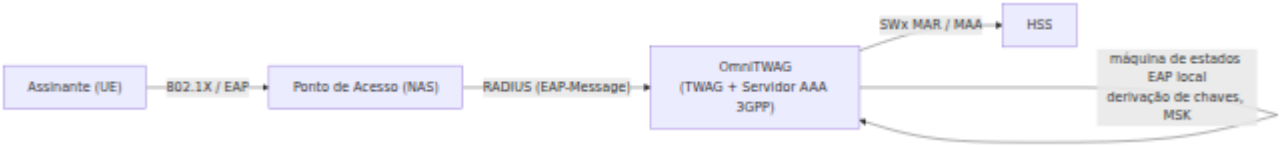
- Visão Geral
- Métodos EAP Suportados
- Máquina de Estados EAP e Fases
- Sequência de Autenticação
- Fonte de Vetores de Autenticação
- Tratamento de Identidade e NAI
- Derivação e Entrega de Chaves
- Resincronização

- Reautenticação Rápida (ERP)
- Tratamento de Falhas e Códigos de Resultado
- Configuração
- Observabilidade
- Solução de Problemas

Visão Geral

O ponto de acesso inicia uma troca EAP IEEE 802.1X com o assinante. O ponto de acesso encapsula cada mensagem EAP em um atributo RADIUS `EAP-Message` e a envia ao TWAG. O TWAG executa uma máquina de estados de autenticação por identidade de assinante. A máquina de estados conduz a troca EAP a um resultado, então retorna um dos três resultados à camada RADIUS.

Resultado	Resposta RADIUS	Significado
Desafio	`Access-Challenge`	A troca precisa de outra rodada EAP.
Aceitar	`Access-Accept`	A autenticação foi bem-sucedida. A resposta carrega a MSK como chaves MPPE.
Rejeitar	`Access-Reject`	A autenticação falhou. A resposta carrega um EAP-Failure.



O TWAG mantém o estado criptográfico por identidade. Ele mantém a resposta esperada (XRES), a chave de autenticação (`K_aut`), o RAND, a MSK e a MSK estendida (EMSK) entre as rodadas EAP. Após o TWAG entregar as chaves MPPE, o TWAG zera o material sensível da chave.

Métodos EAP Suportados

O TWAG suporta três métodos EAP. Cada método usa o mesmo formato de atributo TLV, mas um valor de tipo EAP diferente e uma hierarquia de chaves diferente.

Método	Tipo EAP	Especificação	Vetores	K_aut	MAC	Deri C
EAP-AKA'	50 (`0x32`)	RFC 5448	quinteto UMTS	32 bytes	HMAC-SHA-256, truncado para 16 bytes	`CK'`/`I` PRF' (SHA-2
EAP-AKA	23 (`0x17`)	RFC 4187	quinteto UMTS	16 bytes	HMAC-SHA1, truncado para 16 bytes	`MK =` `SHA1(` `| IK` então 186-2
EAP-SIM	18 (`0x12`)	RFC 4186	2 ou 3 trios GSM	16 bytes	HMAC-SHA1, truncado para 16 bytes	`MK =` `SHA1(` `| Kc.` `NONCE` `versã` PRF F

Seleção e Negociação de Método

O TWAG oferece um método padrão. Defina o método com a chave de configuração `eap_method`. O padrão é `:eap_aka_prime` (EAP-AKA').

O assinante pode rejeitar o método oferecido com um EAP-NAK (tipo EAP 3). O NAK lista os tipos EAP que o assinante prefere. O TWAG lê a lista e seleciona o

primeiro tipo que suporta. O TWAG suporta esses tipos no NAK: `50` (EAP-AKA'), `23` (EAP-AKA) e `18` (EAP-SIM).

- Se o NAK listar um tipo suportado, o TWAG reinicia a troca com esse método. Para EAP-AKA' e EAP-AKA, o TWAG envia uma solicitação de identidade AKA com `AT_PERMANENT_ID_REQ`. Para EAP-SIM, o TWAG envia uma solicitação EAP-SIM/Start.
- Se o NAK não listar nenhum tipo suportado, o TWAG rejeita o assinante.

Máquina de Estados EAP e Fases

O TWAG executa uma máquina de estados por identidade de assinante, dentro de um processo de cliente por identidade. A máquina de estados rastreia uma fase EAP. A fase determina qual mensagem EAP o TWAG espera a seguir e qual telemetria ele emite.

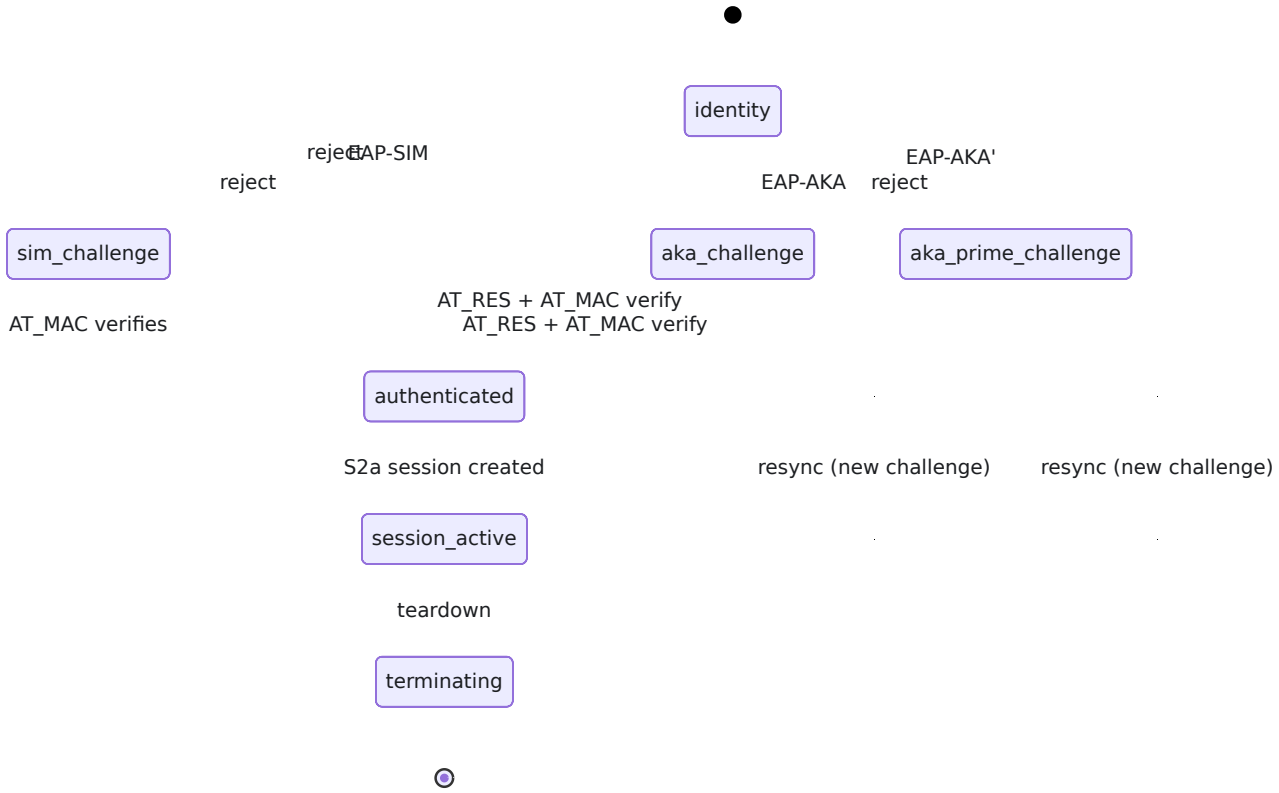
Fase (`eap_phase`)	Significado
`identity`	A fase inicial. O TWAG espera um EAP-Identity, ou a primeira rodada AKA-Identity ou EAP-SIM/Start. Uma rodada falhada retorna o estado para `identity`.
`aka_challenge`	O TWAG enviou um desafio EAP-AKA e aguarda a resposta AKA-Challenge.
`aka_prime_challenge`	O TWAG enviou um desafio EAP-AKA' e aguarda a resposta AKA-Challenge.
`sim_challenge`	O TWAG enviou um desafio EAP-SIM e aguarda a resposta SIM/Challenge.
`authenticated`	A troca EAP foi bem-sucedida. O TWAG tem a MSK e está prestes a estabelecer a sessão S2a.
`session_active`	A sessão S2a está ativa. O assinante pode passar dados do usuário. A contabilidade RADIUS também pode promover uma sessão para esta fase.

Existem dois valores de fase adicionais no código:

- `resyncing` é reservado como uma fase de autenticação em andamento. Na prática, uma rodada de resincronização reconstrói o desafio e retorna a fase para `aka_challenge` ou `aka_prime_challenge`, de modo que o estado não se estabiliza em `resyncing`.
- `terminating` marca uma sessão que está sendo encerrada.

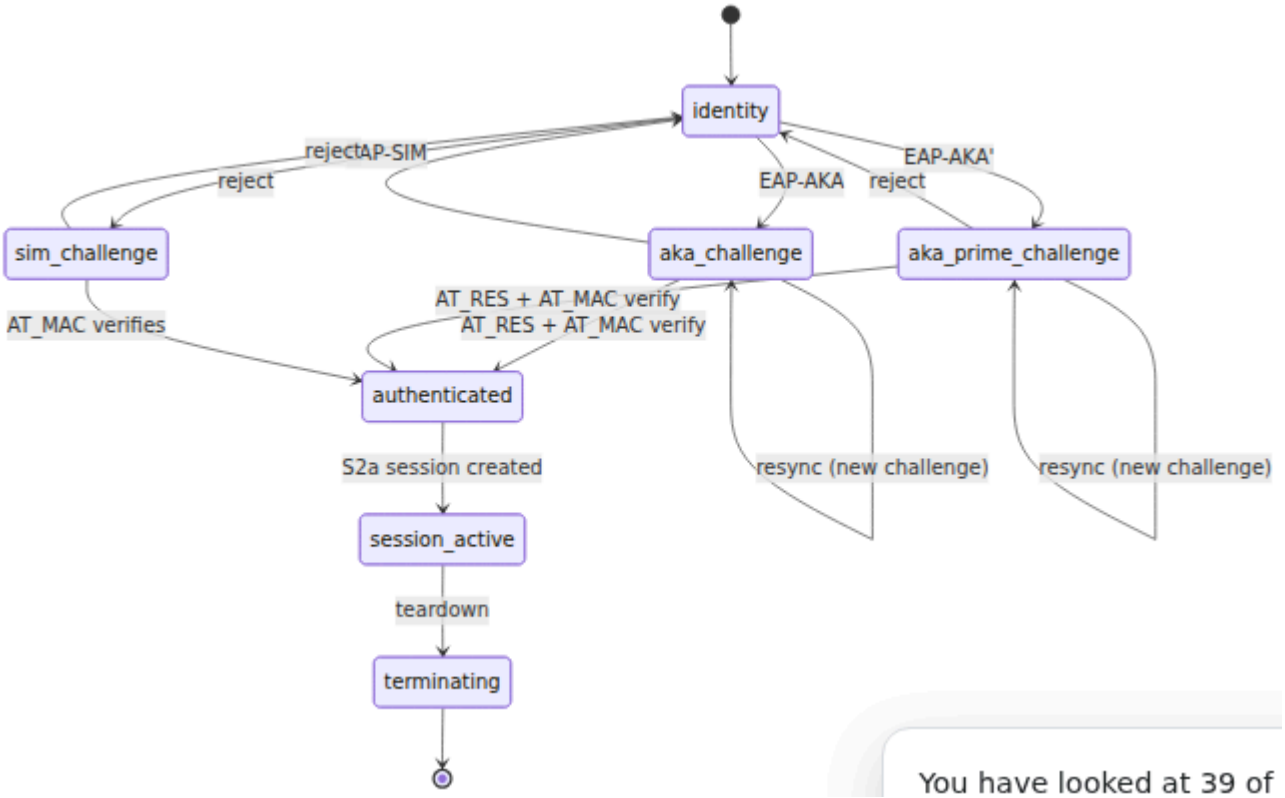
Uma sessão que permanece em uma fase de autenticação em andamento (`identity`, `aka_challenge`, `aka_prime_challenge`, `sim_challenge` ou `resyncing`) além do tempo limite de identidade é encerrada. Defina o tempo limite com `identity_timeout_ms`. Veja Configuração.

A página Sessões mapeia essas fases para o status de autenticação da sessão mostrado na interface do usuário e na API.



Sequência de Autenticação

A sequência completa tem uma fase de identidade e uma fase de desafio. O exemplo abaixo mostra EAP-AKA'. EAP-AKA tem a mesma forma. EAP-SIM usa uma rodada Start no lugar da rodada AKA-Identity, e não envia `AT_RES`.



Fase de Identidade

1. O assinante envia uma resposta EAP-Identity que carrega o Identificador de Acesso à Rede (NAI).
2. Para EAP-AKA' ou EAP-AKA, o TWAG responde com uma solicitação de identidade AKA que carrega `AT_PERMANENT_ID_REQ`. Isso pede ao assinante a identidade permanente.
3. Para EAP-SIM, o TWAG responde com uma solicitação EAP-SIM/Start que carrega a `AT_VERSION_LIST` (apenas versão 1).
4. O assinante retorna a identidade permanente (AKA-Identity com `AT_IDENTITY`), ou a resposta Start com `AT_NONCE_MT` para EAP-SIM.

Fase de Desafio

1. O TWAG extrai o IMSI da identidade e obtém os vetores de autenticação. Veja Fonte de Vetores de Autenticação.
2. O TWAG deriva as chaves de sessão, incluindo a MSK. Veja Derivação e Entrega de Chaves.
3. O TWAG constrói o desafio. Para EAP-AKA' o desafio carrega `AT_RAND`, `AT_AUTN`, `AT_KDF_INPUT` (o nome da rede), `AT_KDF`, `AT_BIDDING` e `AT_MAC`.

Para EAP-AKA o desafio carrega `AT_RAND`, `AT_AUTN` e `AT_MAC`. Para EAP-SIM o desafio carrega `AT_RAND` (2 ou 3 RANDs) e `AT_MAC`.

4. O assinante verifica a rede (AUTN) e o MAC, então retorna a resposta.
5. O TWAG verifica a resposta. Veja Verificação de Resposta.
6. Com sucesso, o TWAG retorna um EAP-Success e a MSK. Em caso de falha, o TWAG retorna um EAP-Failure.

Verificação de Resposta

O TWAG verifica a resposta ao desafio em uma ordem fixa. A ordem é a mesma para EAP-AKA e EAP-AKA'.

1. **Verificação de estado.** Se o `K_aut` armazenado ou XRES estiver ausente, o TWAG rejeita.
2. **Verificação de atributo.** Se `AT_RES` ou `AT_MAC` estiver ausente, o TWAG rejeita.
3. **Verificação de MAC.** O TWAG zera o campo `AT_MAC`, recalcula o MAC sobre o pacote com `K_aut` e compara. EAP-AKA usa HMAC-SHA1. EAP-AKA' usa HMAC-SHA-256. Ambos truncam para 16 bytes.
4. **Verificação de resposta.** O TWAG compara `AT_RES` com o XRES armazenado. A comparação é **tempo constante** para prevenir ataques de tempo. O TWAG usa uma verificação de comprimento e uma comparação de bytes em tempo constante.
5. **Resultado.** Se tanto o MAC quanto a resposta corresponderem, o TWAG aceita. Caso contrário, o TWAG rejeita.

Para EAP-SIM, o assinante não envia `AT_RES`. A verificação de `AT_MAC` é a única prova de que o assinante possui as chaves corretas. Se o MAC for verificado, o TWAG aceita.

Fonte de Vetores de Autenticação

O TWAG obtém os vetores de autenticação de uma das duas fontes. Defina a fonte com a chave de configuração `auth_mode`.

Modo	`auth_mode`	Fonte do vetor	Uso
SWx	`:swx`	HSS, via Diameter SWx (MAR/MAA)	Produção
Local	qualquer valor diferente de `:swx` (padrão `:local`)	Milenage local de chaves de teste provisionadas	Teste

Em ambos os modos, o TWAG executa a máquina de estados EAP localmente. Apenas a fonte do vetor difere. Qualquer valor de `auth_mode` que não seja `:swx` seleciona o caminho Milenage local.

> **O pass-through STa é um caminho separado.** Um provedor `Twag.Auth.StaAuth` pode retransmitir cargas EAP brutas para um Servidor AAA 3GPP via Diameter STa (DER/DEA), de modo que o Servidor AAA execute a troca EAP-AKA e retorne a MSK. Este provedor existe, mas não está no caminho de autenticação RADIUS ao vivo. O caminho ao vivo busca vetores com `:swx` ou `:local` e executa a máquina de estados EAP no TWAG.

SWx para o HSS

No modo `:swx`, o TWAG envia uma Solicitação de Autenticação Multimídia (MAR) ao HSS via a interface SWx (3GPP TS 29.273, ID de Aplicação `16777265`). O MAR carrega:

- `User-Name`: o IMSI.
- `SIP-Auth-Data-Item` com `SIP-Authentication-Scheme` definido para o método EAP em uso: `EAP-AKA'` ou `EAP-AKA`. Para uma resincronização, o item também carrega `SIP-Authorization`. Isso contém o RAND seguido pelo AUTS.
- `SIP-Number-Auth-Items`: 1.
- `ANID`: o nome da rede. O padrão é `WLAN`. Para `EAP-AKA'`, o HSS o usa para derivar `CK'`/`IK'`. Veja 3GPP TS 29.273 cláusula 8.1.2.1.
- `RAT-Type`: 0.

- `Auth-Session-State`: 1 (NO_STATE_MAINTAINED).
- `Vendor-Specific-Application-Id` com o ID de Aplicação de Autenticação SWx `16777265` e Vendor-Id `10415`.

O HSS retorna uma Resposta de Autenticação Multimídia (MAA). Quando o `Result-Code` é `2001` (sucesso), o TWAG lê o primeiro `SIP-Auth-Data-Item` e extrai o vetor:

- `SIP-Authentication-Scheme` = o esquema do vetor: `EAP-AKA'` ou `EAP-AKA`.
- `SIP-Authenticate` = RAND (16 bytes) seguido por AUTN (16 bytes).
- `SIP-Authorization` = XRES.
- `Confidentiality-Key` = CK para `EAP-AKA`, ou CK' para `EAP-AKA'`.
- `Integrity-Key` = IK para `EAP-AKA`, ou IK' para `EAP-AKA'`.

Se o `Result-Code` não for `2001`, ou os campos do vetor estiverem ausentes, o TWAG trata a resposta como uma negação de autenticação. O TWAG rejeita o assinante com um EAP-Failure e registra uma negação, não uma falha.

Se o TWAG pediu `EAP-AKA'` e o vetor não for um vetor EAP-AKA', o TWAG também rejeita o assinante. Isso se aplica quando o esquema é `EAP-AKA`, ou quando o bit de separação AMF no AUTN é 0. Veja hierarquia de chaves EAP-AKA'.

Sinalização de erro SWx (Experimental-Result)

O SWx sinaliza a maioria dos erros em um AVP `Experimental-Result` agrupado (Vendor-Id `10415`), não no `Result-Code` base. Uma MAA falhada pode, portanto, não carregar nenhum `Result-Code` base. O TWAG lê o `Experimental-Result-Code` também, e registra o nome legível por humanos para que o operador veja a verdadeira causa. Veja Solução de Problemas para os códigos.

Um caso comum é `DIAMETER_ERROR_USER_UNKNOWN` (5001) ou `DIAMETER_ERROR_USER_NO_NON_3GPP_SUBSCRIPTION` (5450): o assinante não está habilitado para acesso não-3GPP no SWx, portanto a autenticação falha, mesmo que o mesmo assinante possa ser conhecido no S6a para acesso EPC.

O DRA deve rotear a aplicação SWx para o HSS. Veja Arquitetura: Roteamento DRA.

Milenage Local

No modo `:local`, o TWAG gera o vetor com o algoritmo Milenage (3GPP TS 35.206) a partir das chaves de teste provisionadas (`test_keys`). As chaves são `ki`, `opc`, `amf` e `sqn`. O TWAG computa RAND, AUTN, XRES, CK e IK no processo. Use este modo apenas para testes em laboratório.

Vetores para EAP-SIM

O EAP-SIM precisa de 2 ou 3 trios GSM. O TWAG deriva os trios dos mesmos vetores Milenage. Para cada vetor, o TWAG pega RAND, pega os primeiros 4 bytes de XRES como SRES e deriva o `Kc` GSM de CK e IK com a função de conversão c3 (3GPP TS 33.102 Anexo C.3): `Kc = CK[0..7] XOR CK[8..15] XOR IK[0..7] XOR IK[8..15]`.

Tratamento de Identidade e NAI

A identidade do assinante é um Identificador de Acesso à Rede (NAI). O NAI tem a forma `{username}@{realm}`, por exemplo `0{IMSI}@wlan.mnc{MNC}.mcc{MCC}.3gppnetwork.org`.

Extração de IMSI

O TWAG extrai o IMSI do NAI em duas etapas:

1. O TWAG pega o nome de usuário, que é a parte antes do `@`.
2. O TWAG remove um dígito de tipo de identidade à frente. Um `0` à frente (identidade permanente, EAP-AKA/AKA') ou um `6` à frente mapeiam para o IMSI puro. Os dígitos restantes são o IMSI.

O TWAG aceita um IMSI de 14 ou 15 dígitos. O IMSI é o `User-Name` no SWx MAR. O TWAG também usa o IMSI para derivar o MCC e o MNC para a sessão GTP-C: o MCC são os primeiros 3 dígitos, e o MNC são os próximos 2 dígitos.

Nome da rede (EAP-AKA')

O EAP-AKA' vincula as chaves ao nome da rede. O 3GPP chama esse valor de **Identidade da Rede de Acesso (ANID)**. O TWAG envia o nome da rede ao assinante em `AT_KDF_INPUT`. O TWAG envia a mesma string ao HSS no AVP `ANID` do SWx MAR. O HSS e o assinante ambos o usam para derivar `CK'`/`IK'`, portanto os dois valores devem ser idênticos.

O padrão é `WLAN`. Este é o ANID para uma rede de acesso WLAN na 3GPP TS 24.302 cláusula 8.1.1.2, sem string ANID adicional. A chave de configuração `network_name` substitui o padrão. Mantenha a chave não definida, a menos que o HSS espere um ANID diferente.

Derivação e Entrega de Chaves

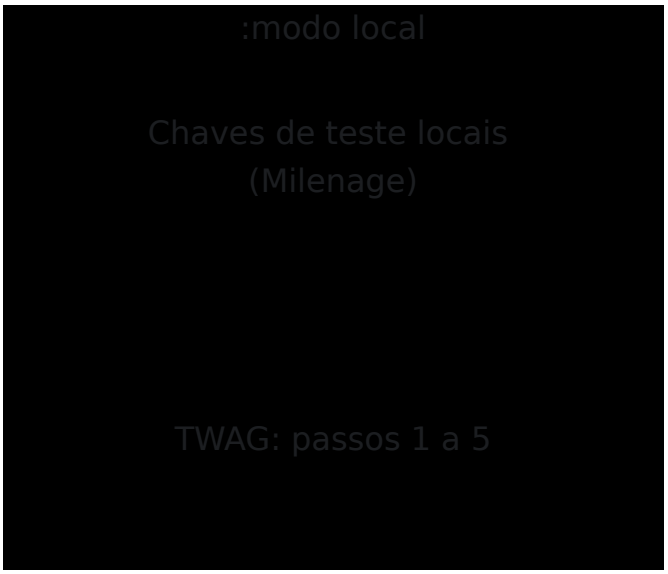
Cada método deriva uma Chave de Sessão Mestre (MSK) de 64 bytes e uma MSK Estendida (EMSK) de 64 bytes. O TWAG entrega a MSK ao ponto de acesso. O TWAG mantém a EMSK para reautenticação rápida.

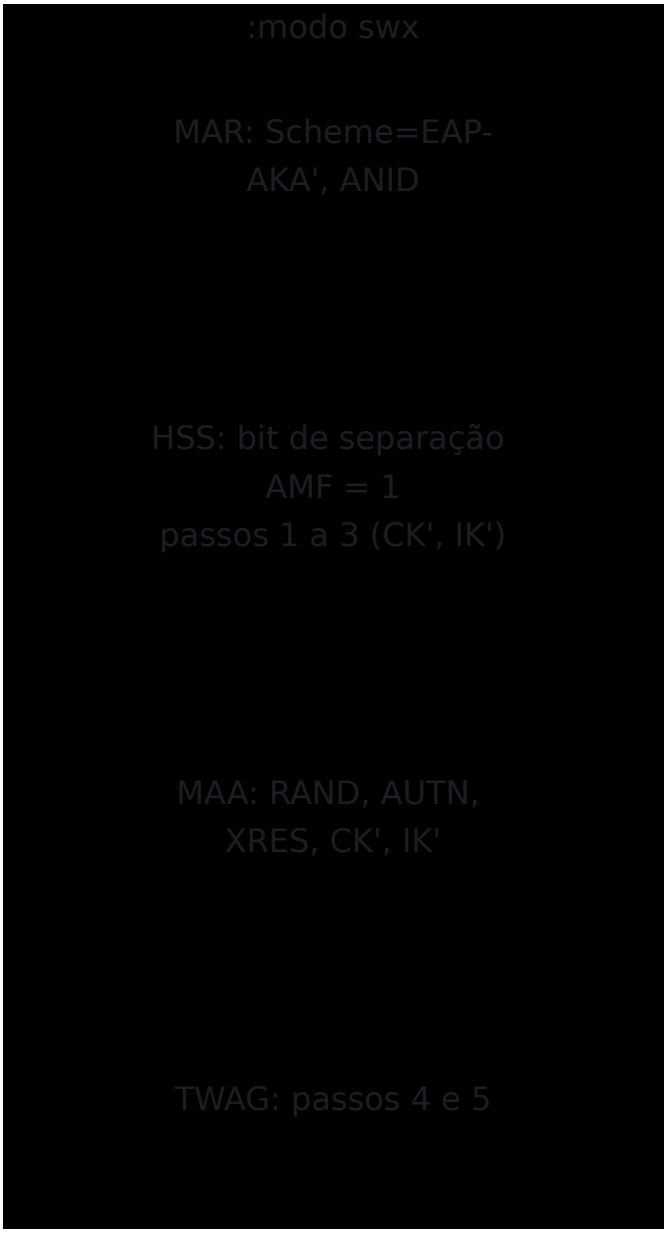
Hierarquia de chaves EAP-AKA'

De acordo com RFC 5448 e 3GPP TS 33.402 cláusula 6.2 e Anexo A:

1. `Key = CK \| IK`.
2. `S = FC(0x20) \| network_name \| len(network_name) \| (SQN XOR AK) \| len(SQN XOR AK)`.
3. `CK' \| IK' = HMAC-SHA-256(Key, S)`. `CK'` são os primeiros 16 bytes. `IK'` são os últimos 16 bytes.
4. `MK = PRF'(IK' \| CK', "EAP-AKA'" \| Identidade)`, 208 bytes. PRF' usa HMAC-SHA-256.
5. O TWAG divide `MK`: `K_encr` (16), `K_aut` (32), `K_re` (32), `MSK` (64), `EMSK` (64).

A fonte do vetor decide onde os passos 1 a 3 são executados:


:modo local
Chaves de teste locais
(Milenage)
TWAG: passos 1 a 5


:modo swx
MAR: Scheme=EAP-
AKA', ANID
HSS: bit de separação
AMF = 1
passos 1 a 3 (CK', IK')
MAA: RAND, AUTN,
XRES, CK', IK'
TWAG: passos 4 e 5

No modo `:swx`, o HSS executa os passos 1 a 3. O TWAG pede um vetor EAP-AKA' no MAR. O HSS define o bit de separação AMF como 1 e deriva `CK'`/`IK'` com o ANID. A MAA carrega `CK'` em `Confidentiality-Key` e `IK'` em `Integrity-Key`. O TWAG usa essas chaves conforme recebidas e executa os passos 4 e 5. `CK` e `IK` não saem do HSS para um vetor AKA'. Veja 3GPP TS 33.402 cláusula 6.2, passo 10.

No modo `:local`, o TWAG é a fonte do vetor. O TWAG executa todos os cinco passos com as chaves de teste configuradas.

Se o HSS responder a um pedido EAP-AKA' com o esquema `EAP-AKA`, ou com o bit de separação AMF 0, o TWAG rejeita o assinante com um EAP-Failure. A linha de log começa com `HSS vector unusable for EAP-AKA'`. O assinante rejeita tal desafio em qualquer caso, conforme a cláusula 6.2 passo 15 do TS 33.402. A rejeição antecipada nomeia o HSS como a causa em vez de um `Client-Error` do dispositivo.

Hierarquia de chaves EAP-AKA

De acordo com RFC 4187:

1. `MK = SHA1(Identidade \| IK \| CK)`.
2. O TWAG expande `MK` para 160 bytes com o PRF FIPS 186-2.
3. O TWAG divide a saída: `K_encr` (16), `K_aut` (16), `MSK` (64), `EMSK` (64).

Hierarquia de chaves EAP-SIM

De acordo com RFC 4186:

1. `MK = SHA1(Identidade \| Kc1 \| Kc2 \| [Kc3] \| NONCE_MT \| Version_List \| Selected_Version)`.
2. O TWAG expande `MK` para 160 bytes com o PRF FIPS 186-2.
3. O TWAG divide a saída: `K_encr` (16), `K_aut` (16), `MSK` (64), `EMSK` (64).

MSK para MPPE para o ponto de acesso

O TWAG entrega a MSK de 64 bytes ao ponto de acesso no RADIUS Access-Accept. O TWAG divide a MSK e codifica cada metade como uma chave MPPE Microsoft (RFC 2548, RFC 3748):

- `MS-MPPE-Recv-Key` = `MSK[0..31]` (primeiros 32 bytes).
- `MS-MPPE-Send-Key` = `MSK[32..63]` (últimos 32 bytes).

O TWAG criptografa cada chave com o segredo compartilhado do RADIUS e o Autenticador de Requisição, conforme o RFC 2548. O TWAG usa um sal de 2 octetos fresco com o bit alto definido para cada chave. O TWAG assina o Access-Accept com o segredo compartilhado. Veja RADIUS Shared Secrets para como o TWAG resolve o segredo.

O ponto de acesso descriptografa as chaves MPPE, recupera o MSK e usa o MSK como a Chave Mestre Par a Par para o handshake de 4 vias 802.11 com o assinante.

O Access-Accept requer um MSK válido de 64 bytes. Se o MSK estiver ausente ou tiver o tamanho errado, o TWAG envia um Access-Reject em vez disso.

Ressincronização

O assinante pode relatar que seu número de sequência (SQN) está fora de sincronia com a rede. O assinante envia um AKA-Synchronization-Failure (subtipo `0x04`) com `AT_AUTS`. `AT_AUTS` tem 14 bytes: 6 bytes de SQN oculto e 8 bytes de MAC-S.

O TWAG lida com a ressincronização da seguinte forma:

1. O TWAG constrói as informações de ressincronização: o RAND armazenado concatenado com os 14 bytes de AUTS.
2. O TWAG executa a ressincronização. Isso recupera o SQN do assinante com a função Milenage f5*, verifica o MAC-S e retorna um número de sequência atualizado.
3. Em caso de sucesso, o TWAG gera um novo vetor com o SQN atualizado, constrói um novo desafio e envia um novo Access-Challenge.

4. Em caso de falha, o TWAG rejeita o assinante com um EAP-Failure.

O TWAG lida com a ressincronização tanto para EAP-AKA quanto para EAP-AKA'. A ressincronização usa as chaves Milenage locais. O valor `AT_AUTS` deve ter exatamente 14 bytes, caso contrário, o TWAG gera um erro e a rodada falha.

Reautenticação Rápida (ERP)

O TWAG suporta o Protocolo de Reautenticação EAP (ERP) conforme RFC 6696. O ERP permite que um assinante reauthentique sem uma troca completa de EAP. A economia vem de uma cadeia de chaves que o TWAG deriva do EMSK durante a primeira autenticação completa.

Após uma autenticação completa, o TWAG deriva e armazena as chaves de reautenticação:

- `rRK` = Chave Raiz de Reautenticação, do EMSK.
- `rIK` = Chave de Integridade de Reautenticação, do rRK.

Todas as derivações de chave ERP usam HMAC-SHA-256.

Quando o assinante envia uma mensagem EAP-Initiate/Re-auth (tipo EAP 41), o TWAG faz o seguinte:

1. Se o TWAG não tiver `rRK` ou `rIK` armazenados, o TWAG volta a uma autenticação completa. O TWAG envia uma solicitação de identidade AKA fresca.
2. Se o TWAG tiver as chaves, o TWAG analisa a mensagem e verifica a tag de autenticação com `rIK`.
3. Se a tag for verificada, o TWAG deriva o MSK de reautenticação (`rMSK`) de `rRK` e do número de sequência da mensagem, constrói uma resposta EAP-Finish/Re-auth (tipo EAP 42) e aceita. O `rMSK` se torna o novo MSK para as chaves MPPE.
4. Se a tag não for verificada ou a análise falhar, o TWAG volta a uma autenticação completa.

Tratamento de Falhas e Códigos de

Resultado

Resultados do EAP

Condição	Ação do TWAG
Falha no SWx MAR, ou vetores estão incompletos	A busca do vetor é negada. O TWAG rejeita o assinante de forma limpa com um EAP-Failure e registra uma negação em vez de uma falha.
EAP-AKA' solicitado, mas o esquema MAA é `EAP-AKA`	Access-Reject com EAP-Failure. A linha de log começa com `HSS vector unusable for EAP-AKA'`. O HSS não suporta vetores EAP-AKA'.
EAP-AKA' solicitado, mas o bit de separação AMF no AUTN é 0	Access-Reject com EAP-Failure. Mesma linha de log. O HSS emitiu um vetor que o assinante deve rejeitar (TS 33.402 cláusula 6.2 passo 15).
Falha na verificação de `AT_MAC`	Access-Reject com EAP-Failure.
`AT_RES` não corresponde a XRES	Access-Reject com EAP-Failure.
Ausência de `K_aut` ou estado XRES	Access-Reject com EAP-Failure.
Ausência do atributo `AT_RES` ou `AT_MAC`	Access-Reject com EAP-Failure.
Assinante envia AKA-Authentication-Reject (subtipo `0x02`)	Access-Reject. O assinante rejeitou a rede.

Condição	Ação do TWAG
Assinante envia AKA-Client-Error (subtipo `0x0E`)	Access-Reject.
EAP-NAK sem método suportado	Access-Reject.
MSK ausente ou não 64 bytes no Accept	Access-Reject.
Mensagem EAP não tratada	Access-Reject com EAP-Failure.

Códigos de resultado Diameter do SWx

O código de sucesso é um `Result-Code` base. Os códigos de erro abaixo chegam no AVP `Experimental-Result`, não no `Result-Code` base. Veja a semaforização de erro SWx e Resolução de Problemas.

Código	Nome	Signific
2001	DIAMETER_SUCCESS	O MAA carrega vetor válido.
5001	DIAMETER_ERROR_USER_UNKNOWN	O HSS n tem assinatu não-3GPP/SW para est IMSI. O assinant pode ain ser conhecic no S6a/E
5004	DIAMETER_ERROR_NOT_SUPPORTED_USER_DATA	Os dado usuário solicitad não são suportac
5420	DIAMETER_ERROR_UNKNOWN_EPS_SUBSCRIPTION	Nenhum assinatu EPS para este assinant
5450	DIAMETER_ERROR_USER_NO_NON_3GPP_SUBSCRIPTION	O assina não tem assinatu não-3GP

Código	Nome	Signific
Outro	(erro Diameter)	A busca vetor fal A rodada rejeitada

Referência: RFC 6733 Seção 7.1, 3GPP TS 29.273, 3GPP TS 29.229.

Eventos iniciados pela rede (STa)

O Servidor AAA 3GPP pode enviar eventos para o TWAG através da interface STa (3GPP TS 29.273, ID da Aplicação `16777250`). O TWAG responde a cada evento com `Result-Code` `2001`.

Mensagem	Gatilho	Ação do TWAG
ASR (Abort-Session-Request)	O Servidor AAA deseja uma desconexão.	O TWAG desconecta a sessão do assinante e, em seguida, responde ASA.
STR (Session-Termination-Request)	Desconexão iniciada pelo HSS.	O TWAG desconecta a sessão do assinante e, em seguida, responde STA.
RAR (Re-Auth-Request)	O Servidor AAA deseja uma reautenticação.	O TWAG responde RAA com sucesso e registra a solicitação. A reautenticação completa da sessão ativa ainda não é acionada a partir do RAR.

Todas as três respostas carregam o Origin-Host e Origin-Realm da identidade Diameter. Veja Configuração.

Configuração

Defina os parâmetros de autenticação na configuração da aplicação `:omnitwag`. A identidade Diameter (`host` e `realm`) vem da configuração `:diameter_ex`, e o TWAG a usa como Origin-Host e Origin-Realm no SWx e STa.

```
config :omnitwag,
  # Fonte do vetor: :swx (produção) ou :local (teste)
  auth_mode: :swx,

  # Método EAP oferecido: :eap_aka_prime (padrão), :eap_aka ou
:eap_sim
  eap_method: :eap_aka_prime,

  # Nome da rede EAP-AKA' (ANID). O TWAG o envia em AT_KDF_INPUT e
no
  # SWx MAR. O padrão é "WLAN" (TS 24.302 cláusula 8.1.1.2).
Mantenha não definido em
  # implantações normais.
  network_name: "WLAN",

  # Recolha uma sessão ainda presa em uma fase de autenticação em
andamento após
  # este número de milissegundos. Padrão 600000 (10 minutos).
  identity_timeout_ms: 600_000,

  # Campos do Identificador TWAN, adicionados ao STa DER/AAR
quando definidos
  twan_ssid: "ons",
  twan_bssid: "00:11:22:33:44:55",

  # Chaves de teste locais (usadas apenas quando auth_mode:
:local)
  test_keys: %{
    ki: "11111111111111111111111111111111",
    opc: "22222222222222222222222222222222",
    amf: "8000",
    sqn: 1200
  }
```

Parâmetro	Tipo	Necessário	Padrão	Desc
`auth_mode`	Átomo	Não	`:local`	Fonte do `:swx` ob vetores via SWx. gera vet partir de `test_ke` Defina `:` produçã
`eap_method`	Átomo	Não	`:eap_aka_prime`	O métod oferecid `:eap_ak` `:eap_ak` `:eap_si` assinant negociar método suportad diferente EAP-NAK
`network_name`	String	Não	`WLAN`	O nome EAP-AKA O TWAG em `AT_KDF_` como o A `ANID` no MAR. `WL` valor qu 24.302 c 8.1.1.2 c para ace WLAN.

Parâmetro	Tipo	Necessário	Padrão	Desc
`identity_timeout_ms`	Inteiro	Não	`600000`	Tempo e milisseg que uma pode fic uma fas autentic andame antes qu TWAG a
`twan_ssid`	String	Não	-	O SSID W Quando o TWAG um Iden TWAN às mensag DER e A
`twan_bssid`	String	Não	-	O BSSID ponto de Combina `twan_ss` formar o Identific TWAN.
`test_keys`	Mapa	Não	padrões embutidos	Chaves I locais. U apenas `auth_mo` `:local`. Parâmet chave de

Parâmetros de chave de teste

O TWAG lê `test_keys` apenas no modo `:local`. Não use `test_keys` em produção.

Parâmetro	Tipo	Necessário	Padrão	Descrição
`ki`	String (hex)	Sim (modo local)	`1111...`	Chave do assinante `Ki`, 16 bytes como 32 caracteres hexadecimais.
`opc`	String (hex)	Sim (modo local)	`2222...`	Chave variante do operador `OPc`, 16 bytes como 32 caracteres hexadecimais.
`amf`	String (hex)	Sim (modo local)	`8000`	Campo de Gestão de Autenticação, 2 bytes como 4 caracteres hexadecimais.
`sqn`	Inteiro	Sim (modo local)	`1200`	O número de sequência inicial.

Observabilidade

O TWAG emite contadores de telemetria através da máquina de estados EAP e do manipulador STa. O endpoint Prometheus expõe os contadores. Veja a Referência de Métricas para o endpoint e exemplos de consulta.

Métricas EAP

Métrica	Significado
`eap_aka_identity_count`	Trocas de identidade EAP-AKA/AKA' (Fase 1).
`eap_aka_challenge_count`	Trocas de desafio EAP-AKA/AKA' (Fase 2).
`eap_aka_sync_failure_count`	Falhas de sincronização AKA (eventos de ressincronização).
`eap_aka_auth_success_count`	Autenticações EAP-AKA/AKA' bem-sucedidas.
`eap_aka_auth_reject_count`	Autenticações EAP-AKA/AKA' rejeitadas.

O TWAG também emite eventos de telemetria para EAP-SIM (`start`, `auth_success`, `auth_reject`), para AKA `auth_reject` e `client_error`, e para ERP (`initiate`, `success`, `fallback`, `mac_failure`). Veja a Referência de Métricas.

Métricas de resposta RADIUS

Métrica	Significado
`radius_access_challenge_count`	Respostas Access-Challenge (rodada EAP em progresso).
`radius_access_accept_count`	Respostas Access-Accept (autenticação bem-sucedida).
`radius_access_reject_count`	Respostas Access-Reject (autenticação falhada).

Métricas de eventos STa

O manipulador STa emite um contador de telemetria para cada evento iniciado pela rede: `[:sta, :asr, :received, :count]`, `[:sta, :str, :received, :count]`, e `[:sta, :rar, :received, :count]`, além de contadores de sessão `disconnect` e `reauth`.

Contexto de Log

O TWAG define os metadados de log `identity` e `imsi` para cada sessão. Use esses campos para rastrear um assinante através do fluxo de autenticação.

Resolução de Problemas

Sintoma	Causa provável	
A autenticação falha e o log mostra `DIAMETER_ERROR_USER_UNKNOWN (5001)`	O HSS não tem assinatura não-3GPP/SWx para este IMSI. O assinante pode ser conhecido no S6a/EPC, mas não está habilitado para acesso não-3GPP.	Habilit não-3G assinar
A autenticação falha e o log mostra `DIAMETER_ERROR_USER_NO_NON_3GPP_SUBSCRIPTION (5450)`	O perfil do assinante não tem assinatura não-3GPP.	Provisi assinat no HSS
O log mostra `Result-Code=nil` em um MAA falhado	O SWx retornou um erro no AVP `Experimental-Result`, não no `Result-Code` base.	Leia o `Result` mesma O TWA nome l human
A autenticação nunca é concluída e a sessão é recolhida	O assinante parou em uma fase de autenticação em andamento	Verifiqu troca. A `identi` se nec

Sintoma	Causa provável	
	além do tempo limite de identidade.	
O assinante envia AKA-Authentication-Reject	O UE falhou em autenticar a rede (verificação AUTN).	Verifiqu da rede AMF co SIM no ou se o HSS es modo
Falha repetida de AKA-Synchronization-Failure	O SQN do assinante se desviou do SQN da rede.	O TWA ressinc automa persist manus pelo HS Ressin
Access-Reject com um EAP-Success válido esperado	O MSK estava ausente ou não tinha 64 bytes no Accept.	Confirr do vet chaves Verifiqu ou o co chaves
A autenticação EAP-AKA' falha antes do desafio e o log mostra `HSS vector unusable for EAP-AKA'`	O HSS respondeu ao MAR com um vetor EAP-AKA: esquema `EAP-AKA`, ou bit de separação AMF 0. O HSS	Atualiz o HSS vetores confor 8.1.2.1 Como s alterna

Sintoma	Causa provável	
	não suporta vetores EAP-AKA', ou ignorou o `ANID`.	`eap_me` `:eap_a`
O dispositivo responde ao desafio EAP-AKA' com AKA-Client-Error	O dispositivo não conseguiu verificar o desafio. A causa usual é um desvio no nome da rede: o HSS derivou `CK'`/`IK'` com um ANID diferente do valor `AT_KDF_INPUT`.	Confirr usa o Manter `networ` definid ambos `WLAN`.

Para rejeições na camada RADIUS que ocorrem antes ou durante a troca EAP (segredo compartilhado ruim, ponto de acesso desconhecido), veja RADIUS Shared Secrets e a Referência de Status.

Fluxos de Chamada

Visão Geral

Esta página mostra os fluxos de mensagens de ponta a ponta para o OmniTWAG. Cada fluxo é um diagrama de sequência entre o equipamento do usuário (UE), o ponto de acesso WiFi (AP), o TWAG e os pares da rede central: o HSS (SWx), o PGW (S2a) e o OCS (Gy).

Leia as páginas por recurso para detalhes sobre cada etapa: Autenticação, S2a GTP-C, Plano do Usuário, Gerenciamento de Endereços IP, Pools de IP, Cobrança Online, Credenciais RADIUS, e Sessões.

Índice

- Atores e interfaces
- Anexação bem-sucedida - tunelada, cobrada
- Anexação - NSWO com um pool de IP
- Anexação - NSWO com distribuição DHCP
- Anexação - quebra local (plano de dados do AP)
- Falha de autenticação - assinante não provisionado para SWx
- Contabilidade e relatório de uso
- Exaustão de cota e reatribuição
- Desmontagem - desconexão do assinante
- Desmontagem - expulsão do operador pela API
- Desmontagem - tempo limite de atividade
- Variações do modo de cobrança

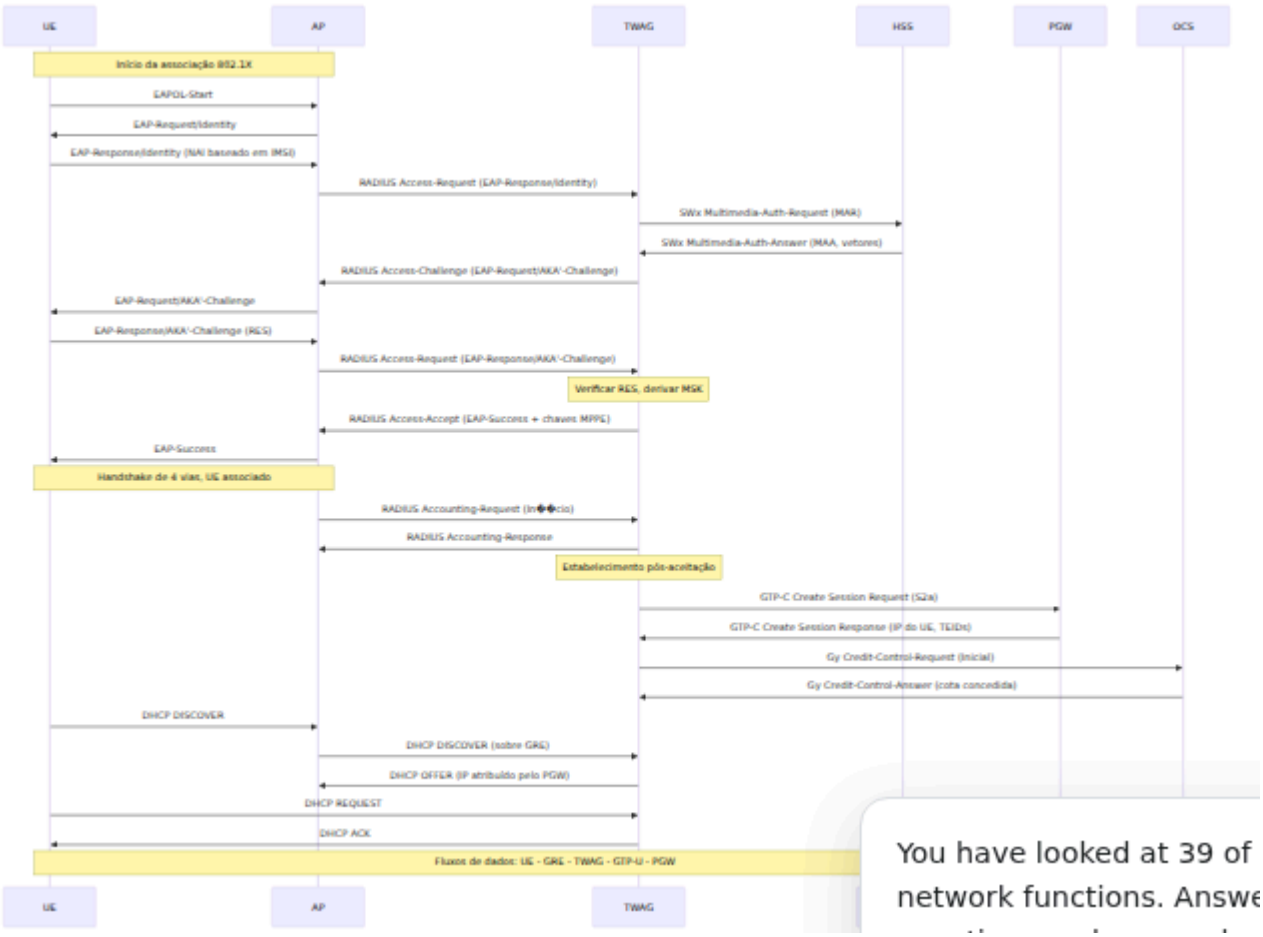
Atores e interfaces

Ator	Papel	Interface para o TWAG
UE	Dispositivo assinante com um SIM	802.11 / EAPOL (via o AP)
AP	Ponto de acesso WiFi e cliente RADIUS	RADIUS (UDP 1812 autenticação, 1813 contabilidade)
TWAG	Gateway de acesso WiFi confiável e servidor AAA 3GPP	-
HSS	Servidor de assinante doméstico	Diameter SWx (vetores de autenticação)
PGW	Gateway de pacotes	GTP-C/GTP-U sobre S2a (somente tunelado)
OCS	Sistema de cobrança online	Diameter Gy (somente cobrado)

O tipo de quebra e o modo de cobrança de uma sessão vêm das credenciais do AP. Veja Credenciais RADIUS.

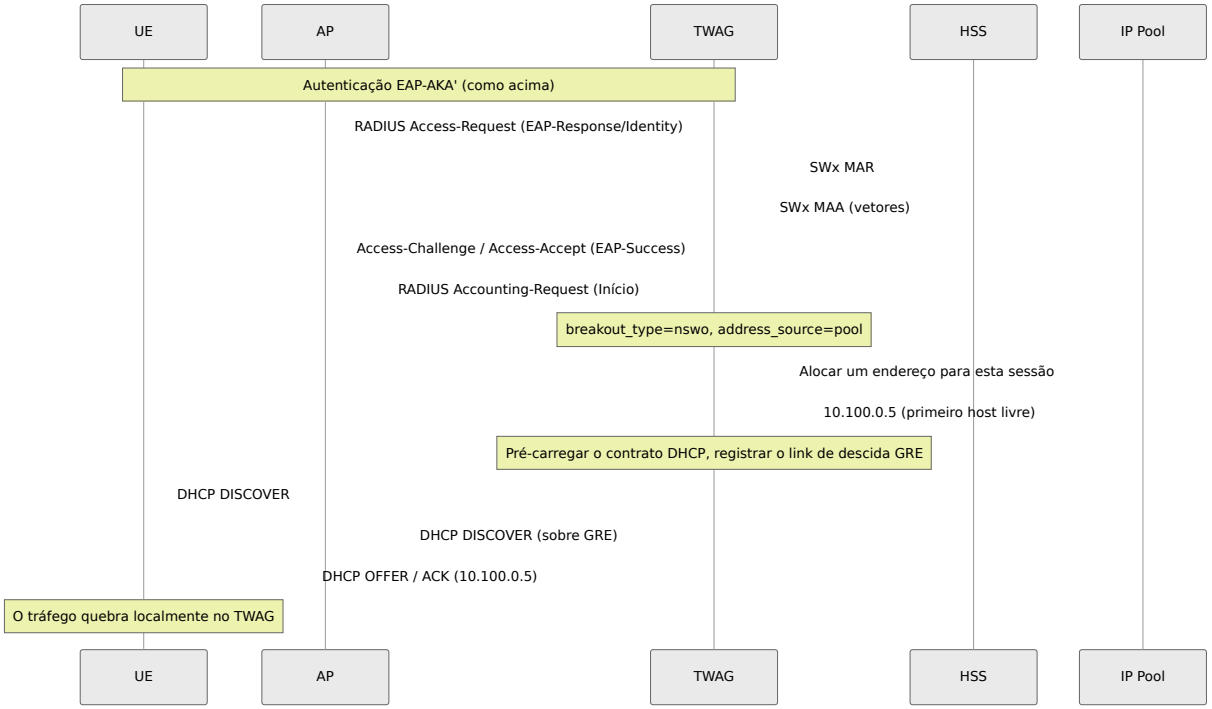
Anexação bem-sucedida - tunelada, cobrada

Este é o caminho feliz completo para `breakout_type: tunneled` e `charging_mode: charged`. O TWAG autentica o assinante com EAP-AKA', estabelece uma sessão GTP com o PGW, entrega o IP atribuído pelo PGW via DHCP e abre uma sessão de cobrança Gy.



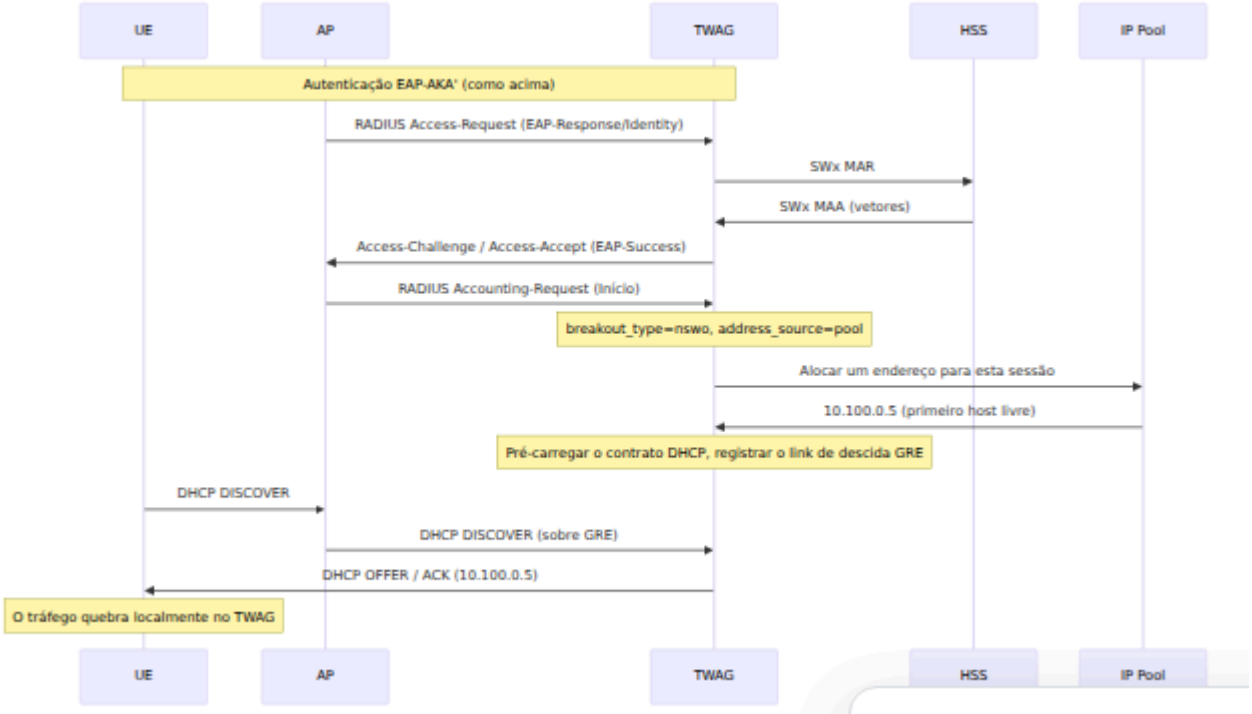
Anexação - NSWO com um pool de IP

Para `breakout_type: nswo` e `address_source: pool`, o TWAG não contata o PGW. O TWAG pré-aloca um endereço fixo do pool de IP associado e o entrega via DHCP. O tráfego do assinante quebra localmente.



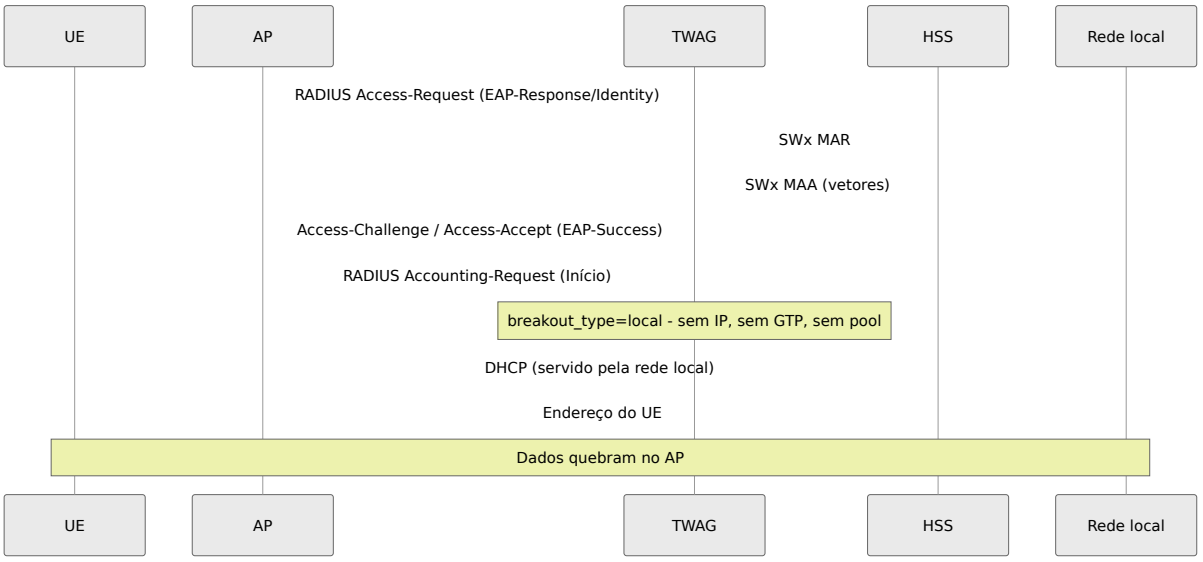
Anexação - NSWO com distribuição DHCP

Para `breakout_type: nswo` e `address_source: dhcp`, o TWAG vincula o MAC do cliente ao pool. O servidor DHCP distribui o próximo endereço livre do intervalo do pool quando o UE envia um DISCOVER. Veja Pools de IP.



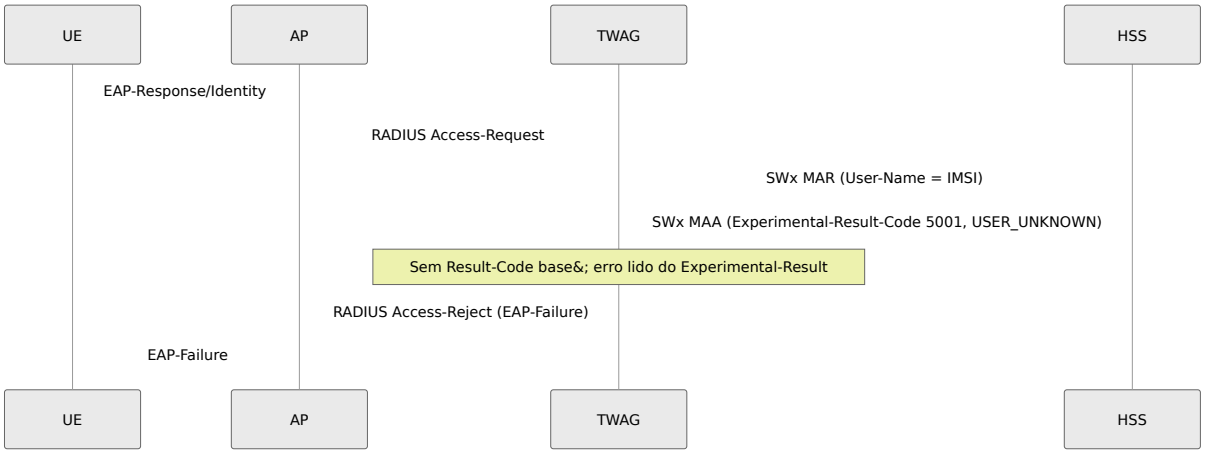
Anexação - quebra local (plano de dados do AP)

Para `breakout_type: local`, o AP possui o plano de dados. O TWAG autentica o assinante e não atribui IP. O UE obtém seu endereço da rede local. O TWAG não estabelece sessão GTP e nem contrato DHCP.



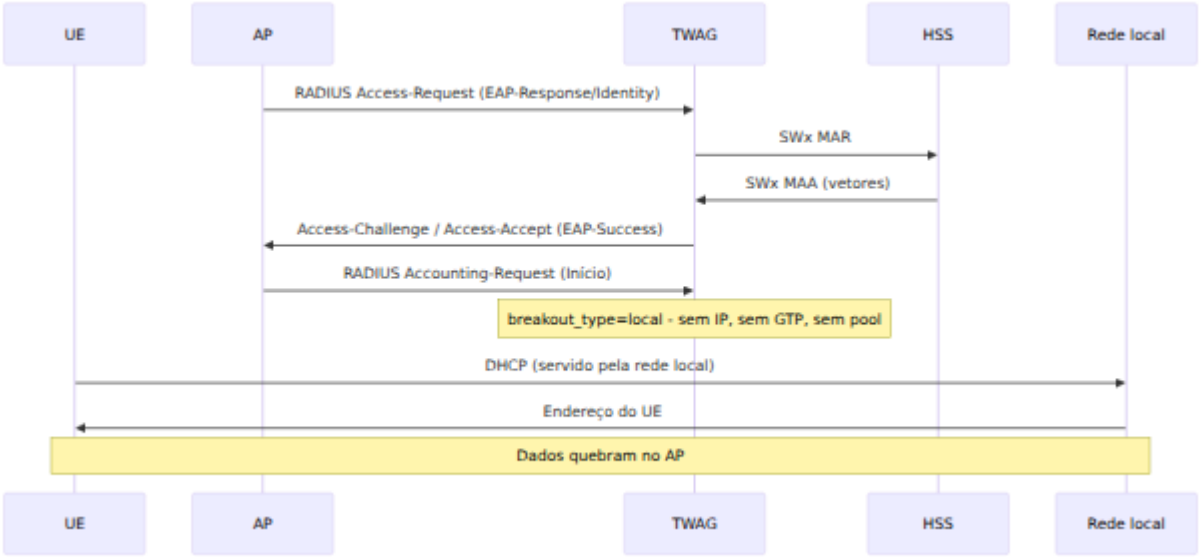
Falha de autenticação - assinante não provisionado para SWx

O HSS sinaliza um erro SWx no AVP `Experimental-Result` agrupado, não no `Result-Code` base. Um assinante conhecido no S6a ainda pode falhar no SWx se o HSS não tiver uma assinatura não-3GPP para o IMSI. O TWAG rejeita a sessão. Veja Autenticação.



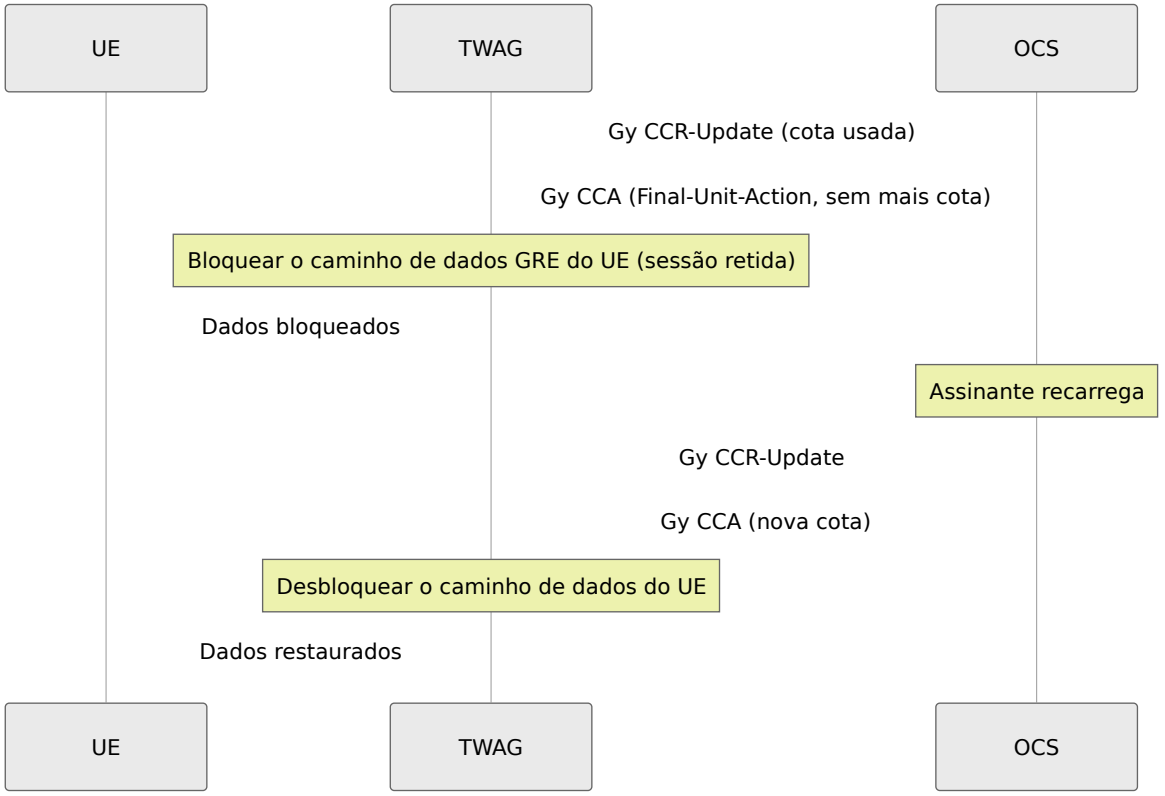
Contabilidade e relatório de uso

O AP envia contabilidade RADIUS. O TWAG mescla o tráfego na sessão (veja Sessões) e, no modo `charged`, relata o uso ao OCS em um limite de cota ou expiração de temporizador de validade, não em cada atualização intermediária. Veja Cobrança Online.



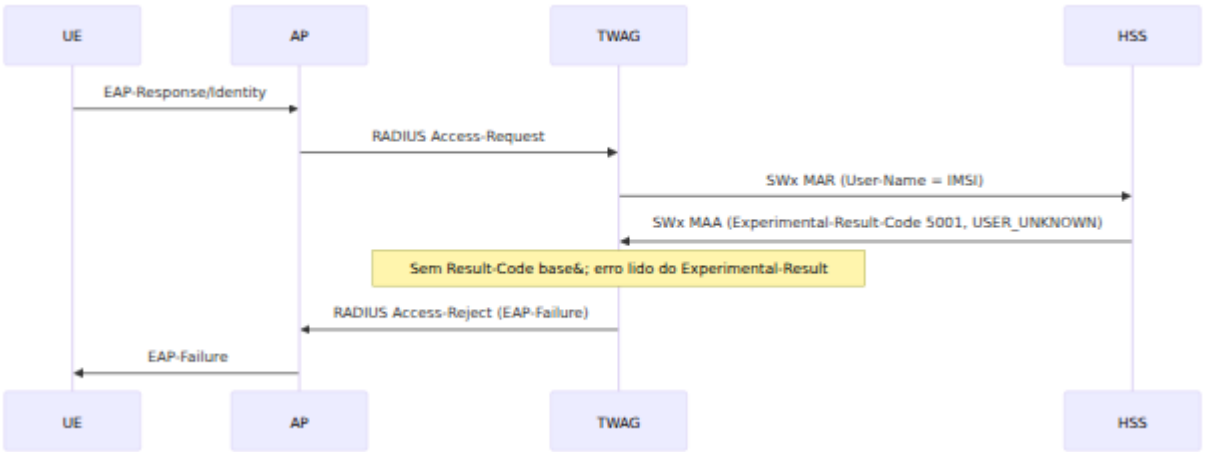
Exaustão de cota e reatribuição

Quando o OCS não concede mais cota, ou define uma Ação Final de Unidade, o TWAG bloqueia o caminho de dados do UE. O TWAG mantém a sessão, o contrato e a contabilidade para que um recarregamento possa desbloquear o UE. Veja Cobrança Online.



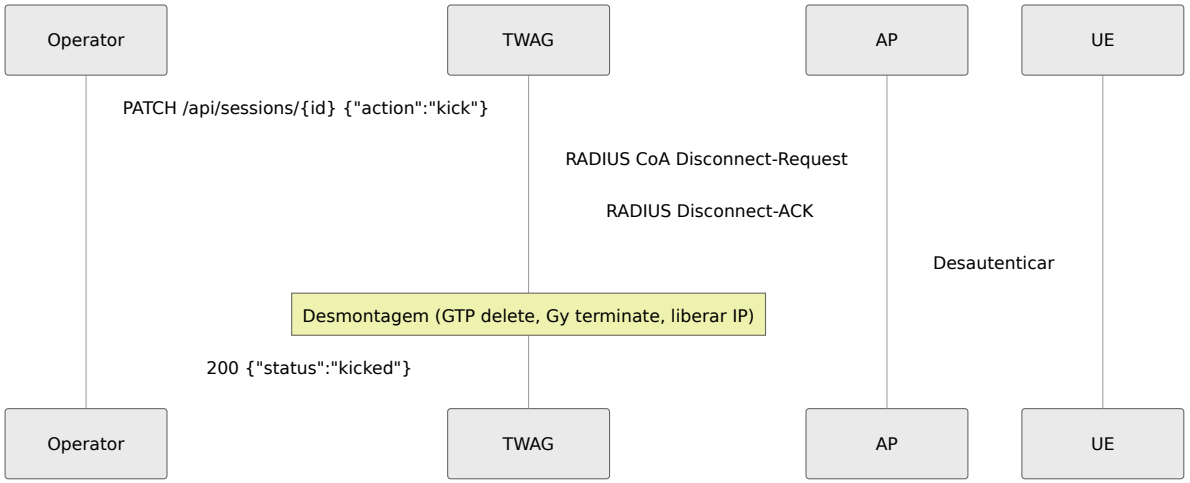
Desmontagem - desconexão do assinante

Quando o UE se desconecta, o AP envia um Accounting-Stop. O TWAG desmonta a sessão: ele exclui a sessão GTP (tunelada), envia um Gy CCR-Terminate (cobrado), libera o endereço e limpa o estado da sessão.



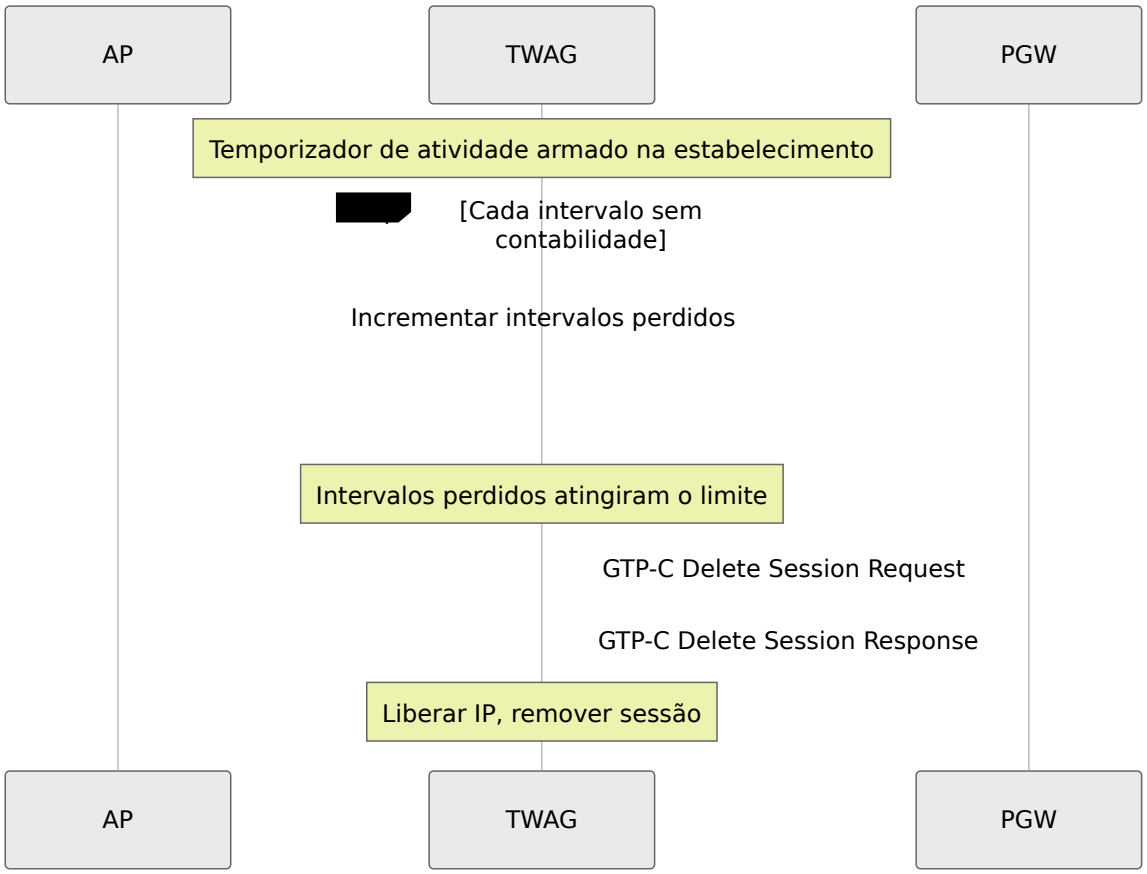
Desmontagem - expulsão do operador pela API

Um operador pode remover uma sessão através da API. O TWAG envia um RADIUS CoA-Disconnect (Disconnect-Message) para o AP, que desconecta o UE, e então desmonta a sessão. Veja Sessões.



Desmontagem - tempo limite de atividade

O TWAG monitora uma sessão estabelecida com um temporizador de atividade. Atualizações intermediárias de contabilidade redefinem o temporizador. Se o AP não enviar atualizações pelo número configurado de intervalos, o TWAG declara a sessão como morta e a desmonta. Uma sessão que nunca completa a autenticação é eliminada por um tempo limite de identidade separado. Veja Sessões.



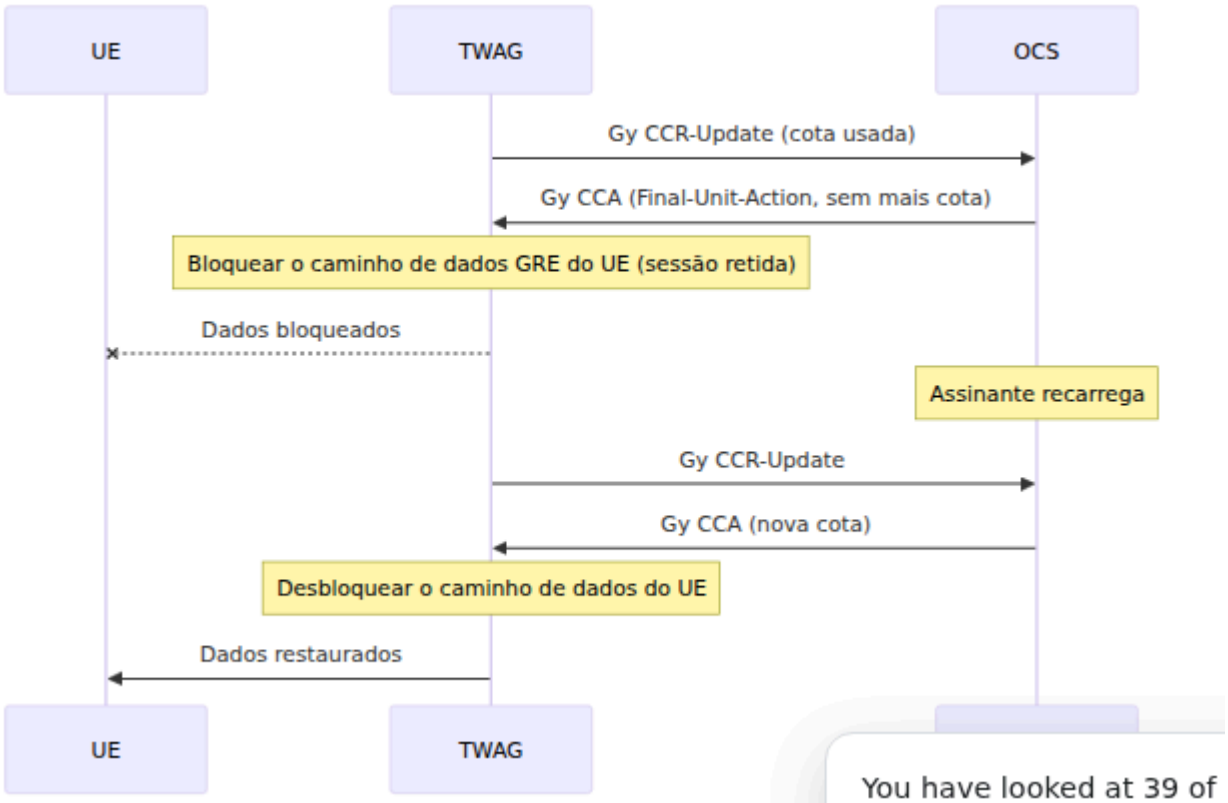
Variações do modo de cobrança

O `charging_mode` nas credenciais do AP altera os passos pós-aceitação. Veja Credenciais RADIUS.

charging_mode	Sessão de dados	Gy
`charged`	Estabelecida (por `breakout_type`)	CCR-Initial / Update / Terminate
`uncharged`	Estabelecida (por `breakout_type`)	Sem sessão Gy
`auth_only`	Não estabelecida	Sem sessão Gy

Para `uncharged`, o fluxo é a anexação tunelada ou NSWO acima, mas o TWAG não envia mensagens Gy. Para `auth_only`, o fluxo para após o Access-Accept e

Accounting-Start: o TWAG não estabelece sessão de dados.



Visão Geral

Modos de Conexão, WLCP e Mobilidade

Visão Geral

Este guia explica os modos de conexão do TWAG, o Protocolo de Controle WLAN (WLCP) e o comportamento de transferência. Ele descreve o que o software OmniTWAG faz hoje. Também marca cada área que é parcial ou ainda não integrada, para que os operadores não planejem em torno de comportamentos que o código não executa.

O TWAG segue o modelo WLAN Confiável na Seção 16 do 3GPP TS 23.402. Para o plano de dados, autenticação e sinalização S2a que esses modos utilizam, veja:

- Arquitetura: plano de dados, ciclo de vida da sessão e interfaces
- Autenticação: EAP-AKA / EAP-AKA' e material de chave
- S2a GTP-C: Criar Sessão, Deletar Sessão e sinalização de bearer
- Sessões: a sessão autenticada e seu ciclo de vida de status
- Plano do Usuário: entrega GRE, GTP-U e DHCP
- Segredos Compartilhados RADIUS: breakout por AP e cobrança aplicada após a autenticação

Índice

- Visão Geral e Comparação de Modos
- Negociação de Modos
- Modo de Conexão Única Transparente (TSCM)
- Modo de Conexão Única (SCM)
- Modo de Múltiplas Conexões (MCM) e WLCP
- Transferência e Mobilidade
- Configuração
- Observabilidade

- Status da Implementação

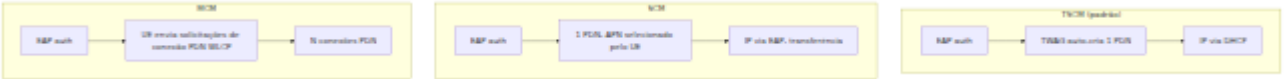
Visão Geral e Comparação de Modos

O TWAG suporta três modos de conexão para um UE que acessa o EPC através de uma WLAN Confiável. O modo controla como o UE aprende seu endereço IP, se o UE pode sinalizar um APN e se o UE pode manter mais de uma conexão PDN.

Propriedade	TSCM	SCM	MCM
Código do modo	`0`	`1`	`2`
Nome completo	Modo de Conexão Única Transparente	Modo de Conexão Única	Modo de Múltiplas Conexões
Conexões PDN por UE	Uma	Uma	Muitas
Entrega de endereço IP	DHCP	EAP	EAP / WLCP
Seleção de APN sinalizada pelo UE	Não	Sim	Sim (por conexão)
Transferência com preservação de IP	Não	Sim	Sim
Bearers dedicados	Não	Sim	Sim
Protocolo de controle do UE	Nenhum	Nenhum	WLCP (TS 24.244)
UE ciente do S2a	Não	Sim	Sim

TSCM é o modo padrão. TSCM corresponde ao comportamento anterior à Release-12, onde o UE não sinaliza a rede. O TWAG seleciona o APN, cria uma conexão PDN e entrega o endereço IP alocado pelo PGW através do DHCP.

Os códigos de modo `0`, `1` e `2` seguem o AVP TWAN-Connection-Mode no 3GPP TS 29.273.

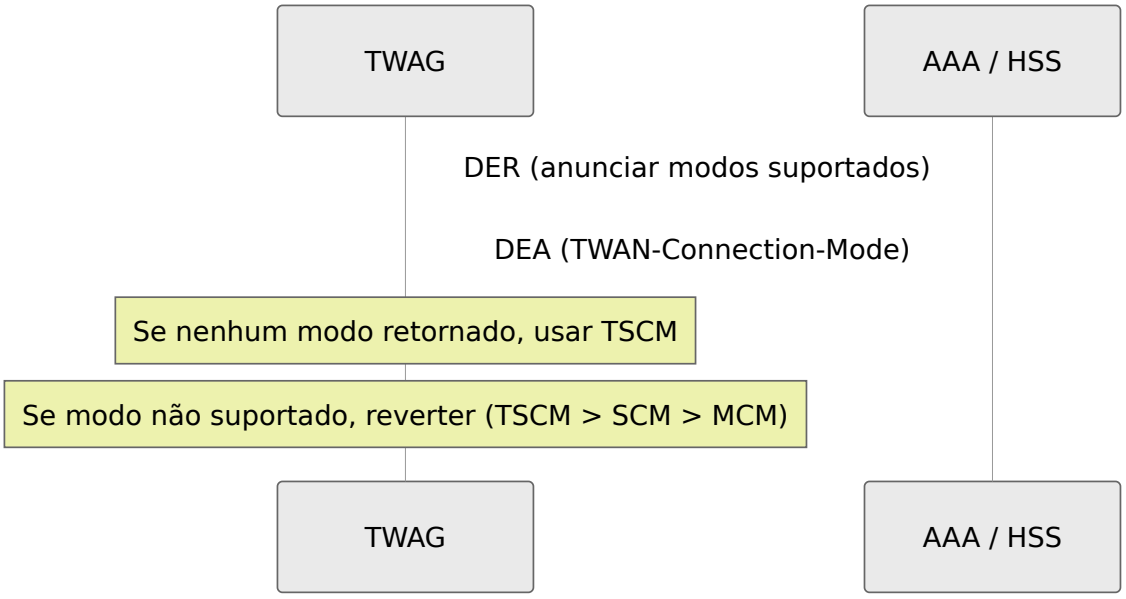


Negociação de Modos

O modo de conexão é negociado durante a autenticação, na troca STa que transporta EAP. Veja Autenticação para o fluxo EAP em si. O design é:

1. O TWAG anuncia os modos que suporta ao servidor AAA. O conjunto anunciado vem da chave de configuração `connection_modes`.
2. O servidor AAA retorna o modo selecionado na resposta STa DEA, no AVP TWAN-Connection-Mode.
3. Se o servidor AAA não indicar um modo, o TWAG usa TSCM como padrão.

Se o servidor AAA selecionar um modo que o TWAG não suporta, o TWAG reverte. A ordem de fallback prefere TSCM, depois SCM, e por último MCM. O TWAG seleciona o primeiro modo suportado dessa ordem.



O TWAG lê o modo selecionado de qualquer uma dessas chaves na resposta AAA, na seguinte ordem: `twan_connection_mode`, `TWAN-Connection-Mode`, `connection_mode` ou `Connection-Mode`. O valor pode ser um código numérico (`0`/`1`/`2`) ou um átomo (`:tscm`/`:scm`/`:mcm`). Um valor não reconhecido padrão é TSCM.

No caminho STa, o STa DER anuncia o modo suportado preferido do TWAG no AVP `TWAN-Connection-Mode`. A leitura do STa DEA analisa o modo negociado e padrão para TSCM quando o DEA não transporta nenhum.

> **Status: o modo negociado não é aplicado no caminho da sessão ativa.** O cliente STa (`StaClient`) anuncia o modo suportado no DER e lê o modo negociado do DEA. O módulo de modo de conexão está presente e testado unitariamente. No entanto, o caminho de autenticação de passagem STa não é o caminho de autenticação RADIUS ativo, e o cliente ativo nunca aplica o modo negociado: o ponto de entrada `store_connection_mode` não tem chamador no caminho de anexação, e o caminho de anexação e dados sempre se comporta como TSCM (IP através do DHCP, sem WLCP, sem transferência sinalizada pelo UE ou APN). Hoje, cada sessão ativa segue o caminho TSCM. Veja Status da Implementação.

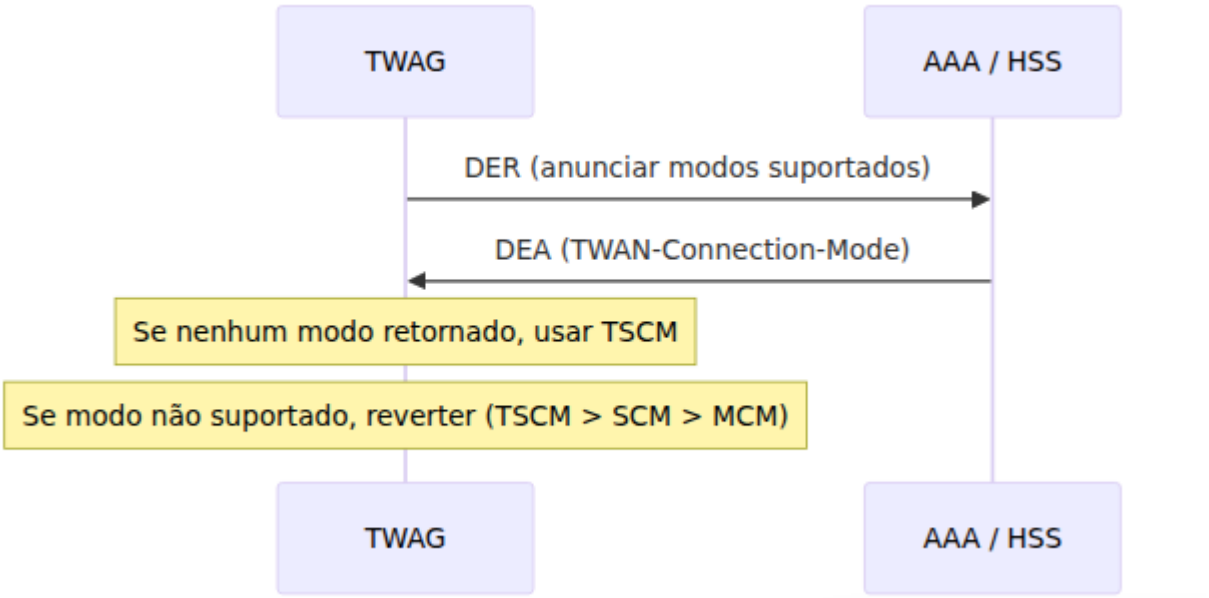
Modo de Conexão Única Transparente (TSCM)

TSCM é o modo padrão e o comportamento que o caminho da sessão ativa executa hoje. O UE não sinaliza a rede. TSCM é descrito no ciclo de vida da sessão da Arquitetura e é resumido aqui.

No TSCM:

- O UE mantém uma conexão PDN.
- O TWAG seleciona o APN. O UE não pode solicitar um APN.
- O UE recebe seu endereço IP através do DHCP, servido pelo TWAG sobre o túnel GRE. Veja Plano do Usuário.
- Não há sinalização WLCP.
- Transferência com preservação de IP, bearers dedicados e seleção de APN sinalizada pelo UE não estão disponíveis.

O TWAG cria a sessão S2a automaticamente após a autenticação. Ele envia uma Solicitação de Criação de Sessão ao PGW com a flag de Indicação de Transferência limpa. Em seguida, pré-aloca o endereço IP atribuído pelo PGW em seu servidor DHCP para o endereço MAC do UE. Veja S2a GTP-C para os detalhes da Criação de Sessão e Sessões para a sessão resultante e seu ciclo de vida de status. A cobrança pela sessão, quando habilitada, segue através do Gy. Veja Cobrança Online (Gy).



Modo de Conexão Única (SCM)

SCM mantém uma única conexão PDN por UE, mas o UE está ciente do S2a. O SCM adiciona essas capacidades sobre o TSCM:

- O UE recebe seu endereço IP através do EAP, não do DHCP.
- O UE pode selecionar um APN não padrão.
- O UE mantém seu endereço IP durante a transferência de 3GPP para WLAN e de WLAN para 3GPP.
- O PGW pode criar bearers dedicados.

O módulo de modo de conexão relata essas capacidades do SCM. Ele classifica o SCM como um modo que entrega IP através do EAP, suporta transferência, suporta seleção de APN e suporta bearers dedicados.

> **Status: o SCM ainda não é um caminho de execução distinto.** O modelo de capacidade para SCM existe, mas o código de estabelecimento de sessão ainda não ramifica no modo negociado. Um UE que o servidor AAA coloca no SCM ainda é servido através do caminho TSCM (PDN auto-criada, IP através do DHCP). Não confie em IP entregue por EAP ou seleção de APN sinalizada pelo UE para SCM nesta versão.

Modo de Múltiplas Conexões (MCM) e WLCP

O MCM permite que um UE mantenha várias conexões PDN ao mesmo tempo. O UE gerencia cada conexão com o Protocolo de Controle WLAN (WLCP), definido no 3GPP TS 24.244. Após a autenticação, o TWAG não cria automaticamente uma conexão PDN. Em vez disso, o UE envia Solicitações de Conexão PDN WLCP. O TWAG mapeia cada conexão PDN para uma sessão S2a GTP-C separada em direção ao PGW.

Transporte WLCP

O TWAG executa um servidor WLCP na porta UDP `2156`, a porta padrão do WLCP. O servidor roteia cada pacote para um manipulador de sessão WLCP por UE, baseado na identidade do UE. O servidor mapeia o IP de origem do UE e a porta para a identidade após a autenticação.

Na produção, a sinalização WLCP é realizada sobre DTLS conforme a Seção 16.1.4A.3.1 do TS 23.402. O servidor atual lida apenas com a camada UDP simples. DTLS ainda não foi aplicado.

Formato da Mensagem WLCP

Cada mensagem WLCP tem um cabeçalho fixo, seguido por uma lista de Elementos de Informação (IEs).

```
Tipo de Mensagem (1 byte) | Comprimento (2 bytes) | ID da
Transação (1 byte) | IEs...
```

Cada IE tem este formato:

```
Tipo de IE (1 byte) | Comprimento do IE (2 bytes) | Valor do IE
(variável)
```

O campo `Comprimento` conta a mensagem completa, incluindo o cabeçalho de 4 bytes.

Tipos de Mensagens WLCP

Código	Mensagem	Direção	Propósito
`0x01`	Solicitação de Conexão PDN	UE para TWAG	Solicitar uma nova conexão PDN
`0x02`	Aceitação de Conexão PDN	TWAG para UE	Conexão criada, com IP / DNS / QoS
`0x03`	Rejeição de Conexão PDN	TWAG para UE	Conexão recusada, com uma causa
`0x04`	Solicitação de Desconexão PDN	UE para TWAG	Liberar uma conexão PDN
`0x05`	Aceitação de Desconexão PDN	TWAG para UE	Conexão liberada
`0x06`	Notificação de Dados de Downlink	TWAG para UE	Notificar o UE sobre dados de downlink pendentes

Elementos de Informação WLCP

Código	IE	Notas
`0x01`	APN	String do Nome do Ponto de Acesso
`0x02`	Tipo de PDN	`1` IPv4, `2` IPv6, `3` IPv4v6
`0x03`	PCO	Opções de Configuração de Protocolo
`0x04`	Endereço PDN	Endereço IP alocado do UE (IPv4, IPv6 ou IPv4v6)
`0x05`	QoS do Bearer	Valores de QCI e MBR / GBR
`0x06`	TFT	Modelo de Fluxo de Tráfego
`0x07`	Causa	Valor da causa do resultado
`0x08`	ID do Bearer	Identidade do Bearer EPS (EBI)
`0x09`	Endereço DNS	Um ou mais endereços de servidor DNS

Valores de Causa WLCP

Código	Causa	Significado
`0x10`	request_accepted	Solicitação aceita
`0x20`	apn_not_allowed	APN não permitido para este UE
`0x21`	no_resources	Nenhum recurso disponível
`0x22`	unknown_apn	APN desconhecida
`0x23`	user_auth_failed	Falha na autenticação do usuário
`0x24`	network_failure	Falha na rede (por exemplo, falha no S2a)
`0x25`	pdn_type_not_supported	Tipo de PDN solicitado não suportado
`0x30`	request_rejected	Solicitação rejeitada

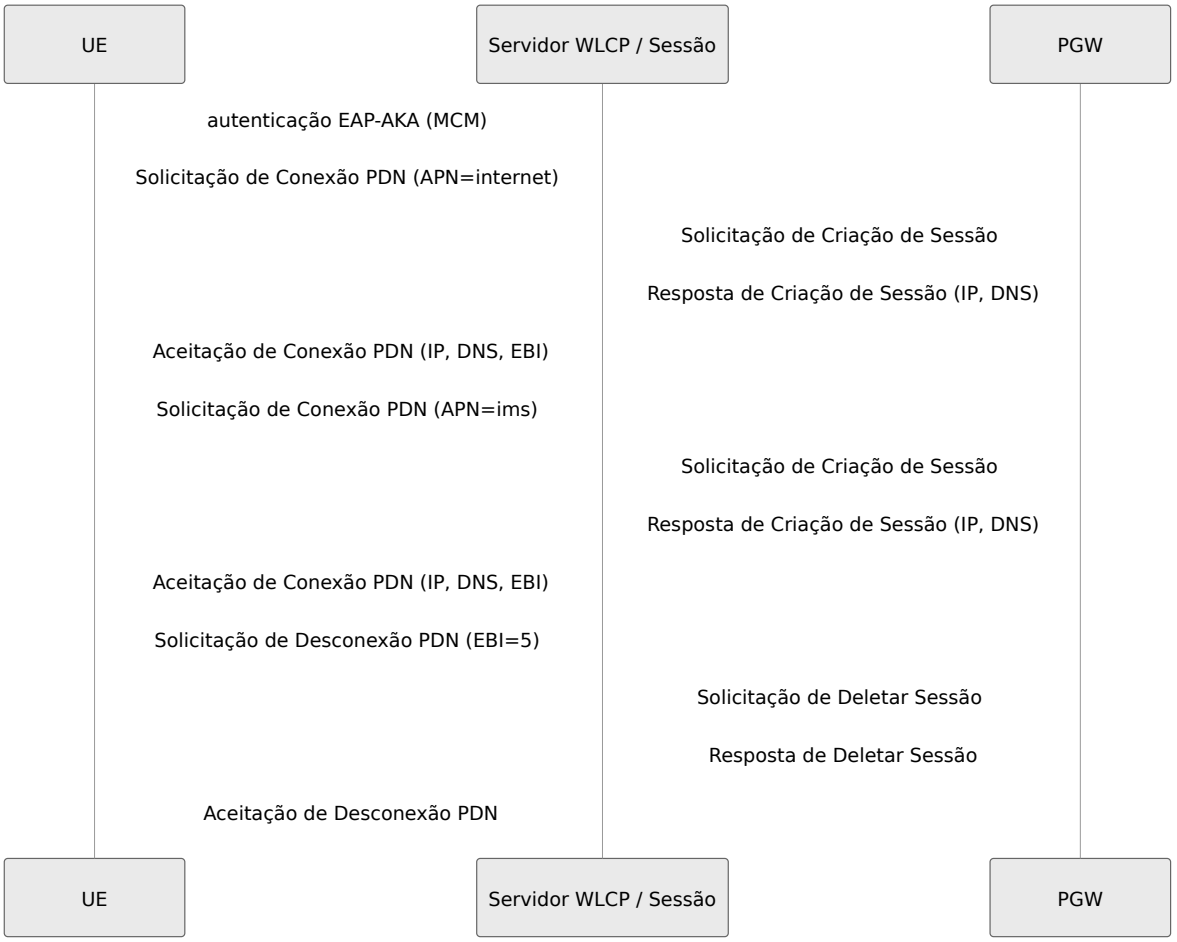
Configuração da Conexão PDN

Quando uma Solicitação de Conexão PDN chega, o manipulador de sessão WLCP realiza os seguintes passos:

1. Lê o APN, tipo de PDN e PCO da mensagem. Se o APN estiver ausente, usa `internet`. Se o tipo de PDN estiver ausente, usa IPv4.
2. Verifica se o APN é permitido. Veja `allowed_apns` em Configuração.
3. Se o UE não tiver um IMSI conhecido, rejeita a solicitação com a causa `user_auth_failed`.
4. Cria uma sessão S2a GTP-C em direção ao PGW para aquele APN, com tipo de RAT `3` (WLAN) e a flag de Indicação de Transferência limpa.
5. Em caso de sucesso, registra o túnel GTP-U e armazena a conexão PDN, com base no EBI. Em seguida, envia uma Aceitação de Conexão PDN com o

endereço IP do UE e, quando presente, os servidores DNS, QoS do bearer e EBI.

6. Em caso de falha no S2a, envia uma Rejeição de Conexão PDN com a causa `network_failure`.



Desconexão da Conexão PDN

Quando uma Solicitação de Desconexão PDN chega, o manipulador encontra a conexão PDN pelo EBI. Se o EBI for desconhecido, responde com uma Rejeição de Conexão PDN e causa `request_rejected`. Se o EBI for conhecido, deleta a sessão S2a GTP-C, desregistra o túnel GTP-U, remove a conexão e responde com uma Aceitação de Desconexão PDN. Quando o UE não tiver mais conexões PDN, o manipulador de sessão WLCP para.

Se o PGW deletar um bearer, o manipulador remove a conexão PDN correspondente e libera o túnel GTP-U local. Ele não envia uma Deletar Sessão neste caso, porque o PGW já liberou o bearer.

Derivação de Chave WLCP

A chave WLCP `K_WLCP` protege a sinalização WLCP. Ela é derivada da Chave de Sessão Mestre (MSK) que a autenticação EAP produz. A derivação segue a Seção 6.2 e o Anexo A do 3GPP TS 33.402.

```
K_WLCP = HMAC-SHA-256(MSK, FC | P0 | L0)

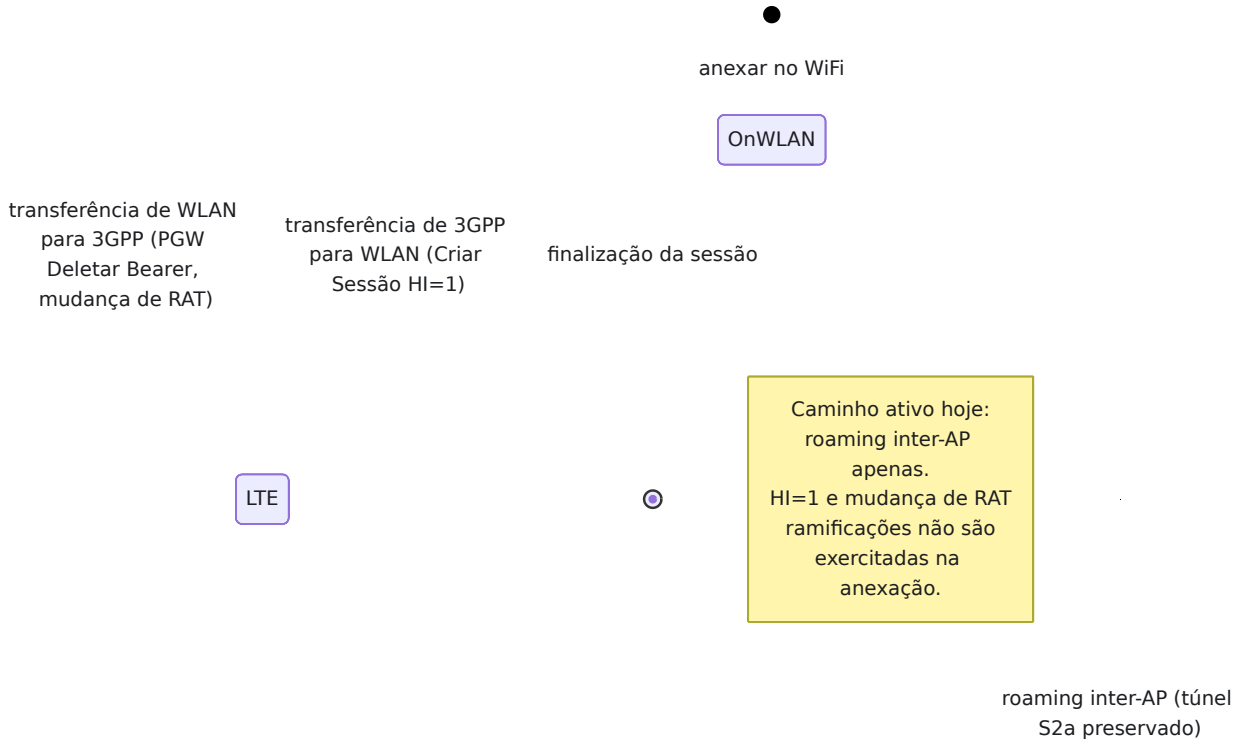
FC = 0x21
P0 = endereço IP do TWAG (4 bytes para IPv4, 16 bytes para IPv6)
L0 = comprimento de P0 (2 bytes, big-endian)
```

A saída é de 32 bytes (256 bits). O TWAG vincula a chave ao seu pr��prio endereço IP através do parâmetro `P0`. Uma variante simplificada deriva a chave da MSK e de um rótulo fixo quando o IP do TWAG ainda não é conhecido.

> **Status: MCM e WLCP são parciais e não integrados.** O codec WLCP, o servidor UDP WLCP, o manipulador de sessão por UE e a derivação de chave existem e são testados unitariamente. Eles não são iniciados pela aplicação e não são acionados pelo fluxo de autenticação. Os seguintes itens não estão implementados:
>
> - Proteção DTLS para WLCP. Apenas UDP simples é tratado.
> - Gerenciamento de bearer dedicado dentro do MCM. O codec transporta IEs de QoS e TFT, mas o manipulador não atua na criação de bearer dedicado iniciada pelo PGW.
> - Transferência dentro do MCM.
> - A mensagem de Notificação de Dados de Downlink é definida no codec, mas não é gerada pelo manipulador de sessão.
>
> Trate o MCM como uma prévia. Não o habilite para UEs de produção.

Transferência e Mobilidade

O TWAG modela três cenários de mobilidade conforme o TS 23.402. Apenas a mobilidade inter-AP está ativa no caminho da sessão ativa hoje.



Transferência de 3GPP para WLAN

Quando um UE se move de LTE para WiFi, o TWAG deve manter a sessão IP existente. Para fazer isso, o TWAG envia uma Solicitação de Criação de Sessão com a flag de Indicação de Transferência (HI) definida. O PGW então preserva o endereço IP existente e o contexto do bearer, em vez de alocar um novo endereço. Veja S2a GTP-C para os detalhes da Criação de Sessão e do IE de Indicação.

A Solicitação de Criação de Sessão S2a suporta a flag de Indicação de Transferência. O manipulador de transferência pode rotear um evento de entrada de transferência para o processo cliente do UE com a flag de transferência definida.

> **Status: o caminho da sessão principal não define a flag de transferência.** O estabelecimento automático de sessão sempre envia Criação de Sessão com a flag de Indicação de Transferência limpa. Como resultado, a preservação de IP na transferência de 3GPP para WLAN n��o é exercitada pelo caminho de anexação normal nesta versão.

Transferência de WLAN para 3GPP

Quando um UE se move de WiFi para LTE, o PGW envia uma Solicitação de Deletar Bearer. O valor da causa indica uma mudança de RAT. O TWAG trata esses valores de causa como uma transferência de mudança de RAT:

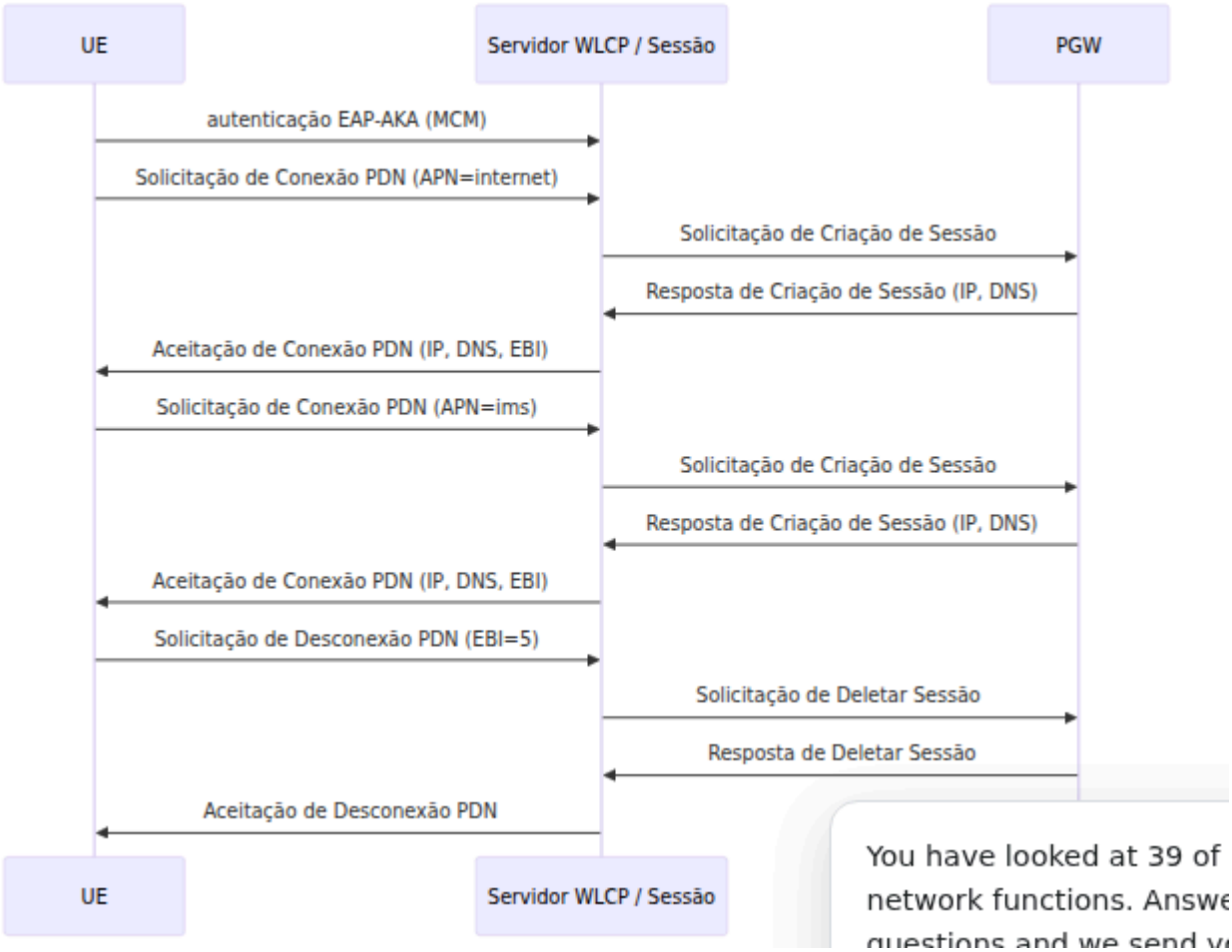
Código de Causa	Nome
`4`	RAT alterado (3GPP para não-3GPP)
`5`	desativação do ISR
`10`	Acesso alterado de não-3GPP para 3GPP

Em uma Deletar Bearer de mudança de RAT, o TWAG limpa a sessão WiFi localmente. Ele não envia uma Deletar Sessão GTP-C, porque o PGW já liberou o bearer. O UE mantém seu endereço IP no lado LTE.

> **Status: o manipulador de deletar-bearer não ramifica na causa hoje.** O processo cliente do UE lida com uma Deletar Bearer do PGW executando a limpeza local e, em seguida, parando, para qualquer causa. A classificação de mudança de RAT existe como um auxiliar, mas o cliente não usa a causa para escolher um caminho de finalização diferente.

Mobilidade Inter-AP

Quando um UE se move entre APs WiFi dentro do mesmo TWAN, o túnel S2a GTP permanece ativo. O UE mantém a mesma identidade, mas chega com um NAS-IP-Address e Called-Station-Id diferentes. O TWAG atualiza os dados de rastreamento de AP no processo cliente do UE e não finaliza a sessão. Isso mantém o endereço IP do UE estável durante a mudança de AP.



A mobilidade inter-AP é o único caminho de mobilidade que o código da sessão ativa executa. Ele atualiza os campos do processo cliente do UE para NAS-IP e Called-Station-Id, e registra o novo AP no estado de rastreamento de AP. Veja Contabilidade por AP para os dados de rastreamento de AP.

Configuração

Modos de Conexão

Defina os modos que o TWAG anuncia com a chave `connection_modes`. O valor é uma lista de átomos de modo. O padrão é `[:tscm]`.

```
config :omnitwag,
  # Modos de conexão que este TWAG anuncia ao servidor AAA.
  connection_modes: [:tscm, :scm]
```

Parâmetro	Tipo	Requerido	Padrão	Descrição
`connection_modes`	Lista	Não	`[:tscm]`	Modos que o TWAG anuncia. Use `:tscm`, `:scm` e `:mcm`. A ordem não importa. Os códigos anunciados são ordenados antes do uso.

> **Nota.** O conjunto anunciado é lido pelo módulo de modo de conexão e o cliente STa envia o modo preferido no STa DER. O caminho da sessão RADIUS ativa ainda não age no modo negociado. Veja Status da Implementação. Mantenha o padrão `[:tscm]` para produção até que SCM e MCM sejam integrados.

Servidor WLCP (MCM)

As configurações do servidor WLCP estão sob a chave `:wlcp`. O servidor WLCP é relevante apenas para MCM.

```
config :omnitwag,
  wlcp: %{
    enabled: true,
    port: 2156,
    allowed_apns: :all
  }
```

Parâmetro	Tipo	Requerido	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Habilitar o servidor WLCP. Deixe `false` a menos que você teste o MCM.
`port`	Inteiro	Não	`2156`	Porta UDP para WLCP. `2156` é a porta padrão do WLCP.
`allowed_apns`	Átomo ou Lista	Não	`:all`	Lista de permissão de APN para solicitações PDN WLCP. Use `:all` para permitir todos os APNs, ou uma lista de strings de APN, por exemplo, `["internet", "ims"]`. Uma solicitação para um APN não na lista é rejeitada com a causa `apn_not_allowed`.

> **Nota.** O servidor WLCP não é iniciado pelo supervisor da aplicação nesta versão. A chave `enabled` é definida para uso futuro.

Configuração Relacionada

O caminho de configuração da PDN do MCM reutiliza as configurações GTP para suas sessões S2a. Veja o resumo de configuração da Arquitetura para o bloco `gtp`, incluindo `pgw_ip`, `local_ip`, `mcc` e `mnc`.

Observabilidade

O servidor WLCP e os manipuladores de sessão emitem eventos de telemetria. Esses eventos alimentam o pipeline de métricas. Veja Métricas para o catálogo de métricas.

Evento de Telemetria	Emitido quando
`[:wlcp, :packet, :received]`	Um pacote WLCP chega. Transporta `bytes`.
`[:wlcp, :packet, :malformed]`	Um pacote falha ao decodificar.
`[:wlcp, :packet, :unregistered]`	Um pacote chega de um endereço de origem não mapeado.
`[:wlcp, :pdn, :request]`	Uma Solicitação de Conexão PDN é recebida.
`[:wlcp, :pdn, :accept]`	Uma conexão PDN é estabelecida.
`[:wlcp, :pdn, :reject]`	Uma conexão PDN é rejeitada. Transporta a causa.
`[:wlcp, :pdn, :disconnect_request]`	Uma Solicitação de Desconexão PDN é recebida.
`[:wlcp, :pdn, :disconnect_accept]`	Uma conexão PDN é liberada.
`[:twag, :handover, :in]`	Um evento de entrada de transferência de 3GPP para WLAN é roteado. Transporta `source`.
`[:twag, :handover, :out]`	Uma mudança de RAT de WLAN para 3GPP é tratada. Transporta `cause`.
`[:twag, :handover, :ap_change]`	Um evento de mobilidade inter-AP é tratado.

O servidor WLCP e os manipuladores de sessão também registram em nível de informação. Eles registram cada solicitação de conexão PDN, aceitação,

rejeição e desconexão, com a identidade do UE, APN e EBI. O manipulador de transferência registra cada entrada de transferência, mudança de RAT e mudança de AP, com a identidade do UE.

Status da Implementação

Esta tabela resume o que está em execução no caminho da sessão ativa, o que existe, mas não está integrado, e o que não está implementado. Use-a para planejar em torno do comportamento real.

Recurso	Status
TSCM (padrão, PDN auto-criada, IP via DHCP)	Ativo
Mobilidade inter-AP (túnel S2a preservado)	Ativo
S2a Criar Sessão com flag de Indicação de Transferência	Suportado no codificador S2a, mas o caminho de anexação sempre limpa
Lógica de negociação de modo de conexão e fallback	Presente como um módulo, testado unitariamente
STa DER anuncia modos suportados	Conectado (STa DER transporta o TWAN-Connection-Mode preferido, além de SSID / BSSID)
STa DEA lê modo selecionado	Conectado (StaClient analisa o modo negociado, padrão para TSCM)
Modo negociado aplicado à sessão ativa	Não conectado (`store_connection_mode` não tem chamador no caminho de anexação)
SCM como um caminho de execução distinto (EAP IP, seleção de APN)	Não integrado (servido como TSCM)
Codec WLCP (todos os tipos de mensagem e IEs)	Presente, testado unitariamente
Servidor UDP WLCP e manipulador de sessão por UE	Presente, não iniciado pela aplicação

Recurso	Status
Derivação de chave WLCP (`K_WLCP`, FC=0x21)	Presente, testado unitariamente
MCM várias conexões PDN simultâneas	Presente no manipulador, não integrado de ponta a ponta
Proteção DTLS para WLCP	Não implementado (apenas UDP simples)
Gerenciamento de bearer dedicado no MCM	Não implementado
Geração de Notificação de Dados de Downlink WLCP	Não implementado (apenas codec)
Transferência no MCM	Não implementado
Preservação de IP de 3GPP para WLAN no caminho de anexação	Não exercitado (anexação limpa a flag HI)
Finalização de WLAN para 3GPP pela causa de mudança de RAT	Não ramificado (limpeza de deletar-bearer é independente da causa)

Gerenciamento de Endereço IP (DHCP / DHCPv6 / IPv6 RA / PCO)

Visão Geral

OmniTWAG fornece a cada UE um endereço IP e os parâmetros de rede que o acompanham. Este subsistema cobre DHCPv4, DHCPv6, Anúncio de Roteador IPv6 (SLAAC) e a entrega de endereços DNS e P-CSCF. O subsistema não inventa um endereço aleatoriamente. Ele entrega um endereço que outra parte do sistema já escolheu. A fonte depende do modo por AP.

- No **modo tunelado (roteado para casa)**, o PGW aloca o endereço sobre S2a GTP, e o OmniTWAG entrega esse mesmo endereço para a UE. Este design segue 3GPP TS 23.402 Seção 16.2.
- No **modo NSWO (quebra local)**, não há PGW. O OmniTWAG aloca o endereço a partir de um **pool de IP** local que o operador define. Veja Pools de IP.

O modo, a fonte do endereço e o pool vêm todos da credencial do AP que corresponde à fonte RADIUS. Veja Credenciais RADIUS para como um AP mapeia para um `breakout_type`, um `address_source` e um `ip_pool_id`.

Para os modos de operação e o caminho de dados GRE, veja a referência User Plane. Para o plano de controle GTP-C que transporta o PAA e PCO, veja S2a GTP-C. Para a máquina de estados por UE que dirige a alocação, veja Sessions.

Índice

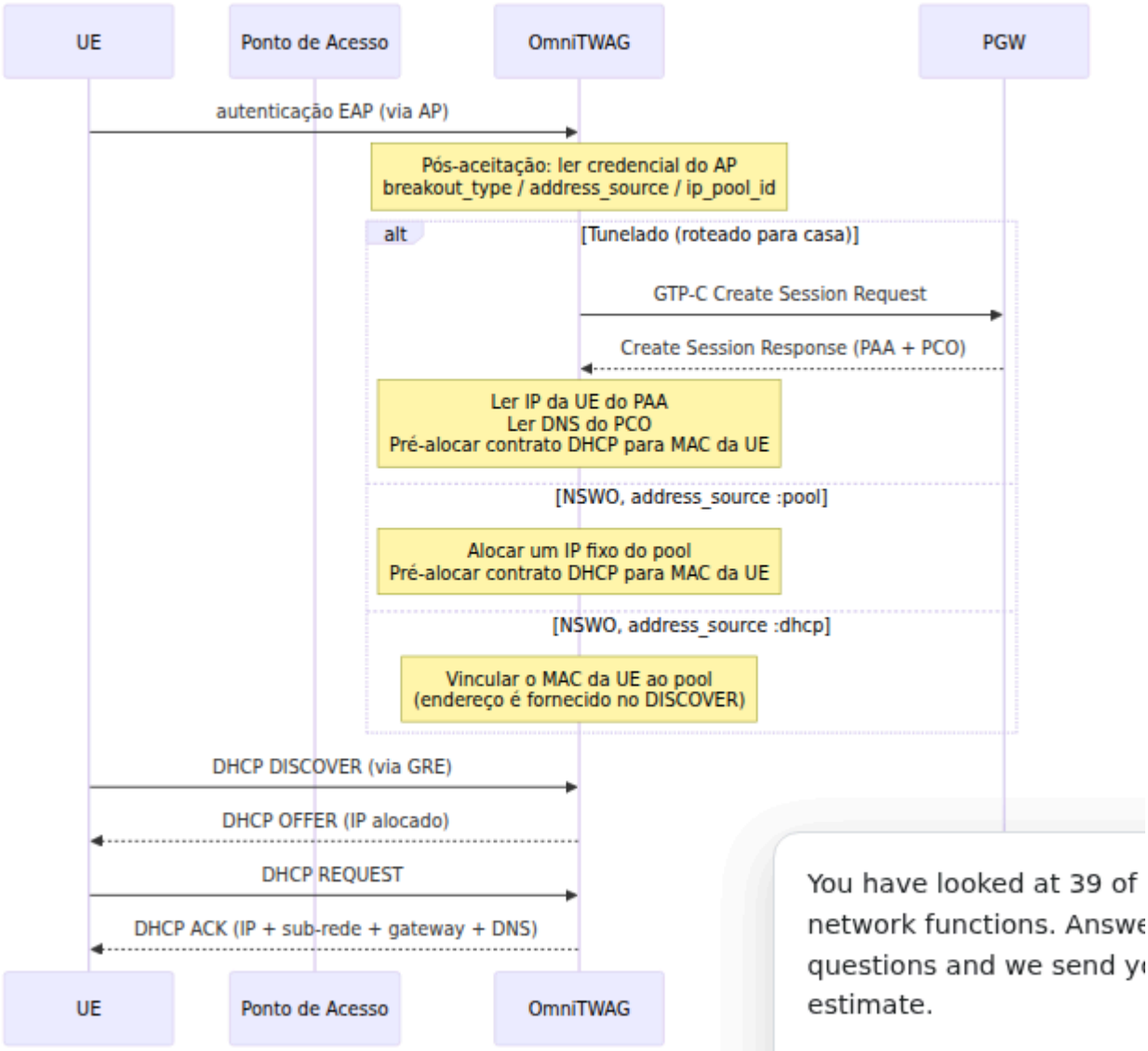
- Visão Geral: quem aloca o endereço
- Fontes de endereço

- Sessões tuneladas: endereço atribuído pelo PGW
- Sessões NSWO: pool e dhcp
- DHCPv4
- DHCPv6
- Anúncio de Roteador IPv6 (SLAAC)
- Entrega de DNS e PCO
- Pools de endereços e exaustão
- Configuração
- Observabilidade
- Solução de Problemas

Visão Geral: quem aloca o endereço

OmniTWAG executa um servidor DHCPv4 leve, um servidor DHCPv6 sem estado e um servidor de Anúncio de Roteador IPv6. Esses servidores não escaneiam uma faixa em busca de um endereço livre no caso tunelado. Eles entregam o endereço que o PGW já alocou. No modo NSWO, o mesmo servidor DHCPv4 serve um endereço que o OmniTWAG retirou de um pool de IP local.

O fluxo de endereços começa quando a UE se autentica. Após a aceitação de acesso, o processo de sessão por UE executa uma etapa de estabelecimento pós-aceitação. Essa etapa lê a credencial do AP para decidir o modo e, em seguida, aloca o endereço pelo caminho correto.



Fontes de endereço

A credencial do AP define a fonte do endereço através de dois campos:

- `breakout_type`: `:tunneled`, `:nswo` ou `:local`.
- `address_source`: `:pool` ou `:dhcp`. Isso se aplica apenas quando `breakout_type` é `:nswo`.

Modo	Credencial	Quem escolhe o endereço da UE	Como chega à UE
Tunelado (roteado para casa)	`breakout_type: :tunneled`	O PGW, na resposta de criação de sessão GTP-C PAA.	Contrato DHCPv4 pré-alocado, DHCPv6 ou RA IPv6.
NSWO, fixo	`breakout_type: :nswo`, `address_source: :pool`	OmniTWAG, do pool `ip_pool_id`, no estabelecimento.	Contrato DHCPv4 pré-alocado.
NSWO, fornecido	`breakout_type: :nswo`, `address_source: :dhcp`	OmniTWAG, do pool `ip_pool_id`, no DHCP DISCOVER.	Fornecimento do pool DHCPv4 no DISCOVER.
AP-local	`breakout_type: :local`	A rede local atrás do AP.	OmniTWAG não atribui nenhum endereço.

Para os campos de credencial e como uma fonte RADIUS mapeia para uma credencial, veja Credenciais RADIUS. Para o armazenamento do pool, veja Pools de IP.

Sessões tuneladas: endereço atribuído pelo PGW

Para uma sessão tunelada, o OmniTWAG estabelece uma sessão S2a com o PGW com uma solicitação de criação de sessão GTP-C. O PGW responde com uma resposta de criação de sessão. A resposta carrega duas coisas que este subsistema usa:

- O **PDN Address Allocation (PAA)**. Este é o endereço IP da UE.

- As **Protocol Configuration Options (PCO)**. Isso carrega servidores DNS, endereços P-CSCF e o MTU do link.

OmniTWAG lê o PAA e o PCO da resposta. O OmniTWAG então pré-aloca um contrato DHCP para o endereço MAC da UE. Quando a UE envia uma solicitação DHCP, o OmniTWAG responde com esse endereço pré-alocado. A UE nunca recebe um endereço que o PGW não alocou. Veja S2a GTP-C para a troca de criação de sessão.

O tipo de PAA define qual endereço o OmniTWAG entrega.

Tipo de PAA	Endereço entregue à UE
`ipv4`	O endereço IPv4 do PAA.
`ipv6`	O endereço IPv6 do PAA.
`ipv4v6`	O endereço IPv4 do PAA.

Se a resposta de criação de sessão não tiver PAA, o OmniTWAG não pré-aloca um contrato. Nesse caso, a UE não recebe nenhum endereço.

Sessões NSWO: pool e dhcp

Uma sessão NSWO não tem PGW, então o OmniTWAG possui o espaço de endereços. O endereço vem do pool de IP nomeado pela credencial `ip_pool_id`. Um pool contém um CIDR IPv4, um gateway, uma lista de servidores DNS e um tempo de contrato. Veja Pools de IP.

O campo `address_source` da credencial seleciona como o endereço do pool chega à UE.

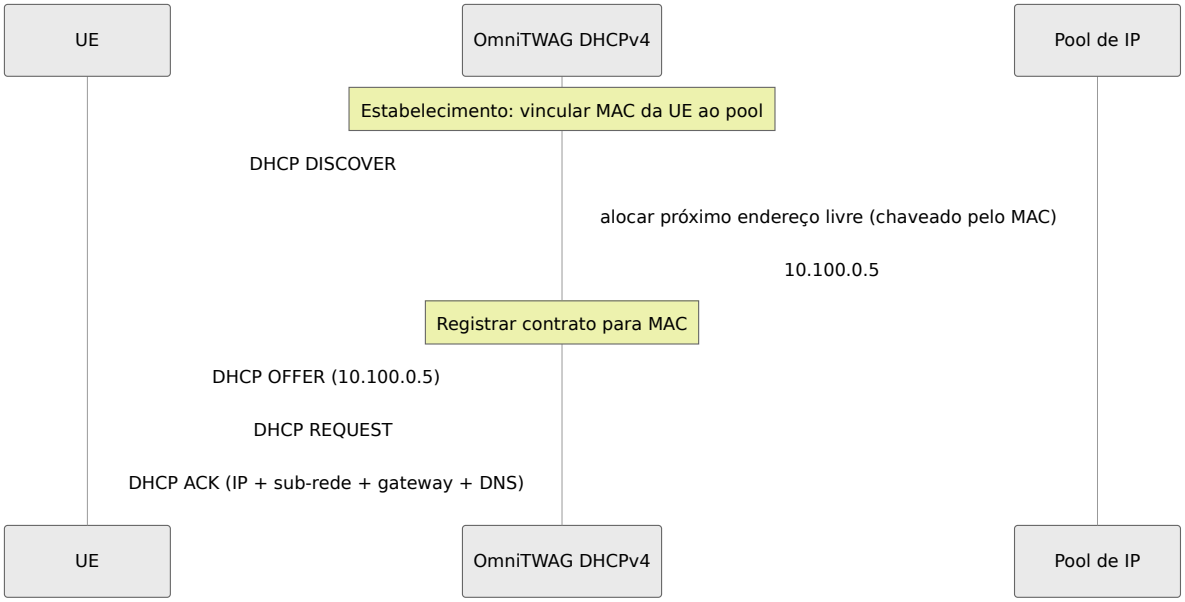
`address_source: :pool` (fixo)

No estabelecimento pós-aceitação, o OmniTWAG aloca um endereço do pool, chaveado pela identidade do assinante. A alocação é idempotente, então a mesma identidade mantém o mesmo endereço em re-estabelecimentos. O

OmniTWAG pré-aloca um contrato DHCPv4 para o MAC da UE com esse endereço, o gateway do pool, os servidores DNS do pool e uma máscara de sub-rede derivada do CIDR do pool. Ele também registra a UE no caminho de dados GRE neste ponto. Quando a UE envia DISCOVER, o servidor oferece o endereço pré-alocado.

`address_source: :dhcp` (fornecido no DISCOVER)

No estabelecimento, o OmniTWAG ainda não aloca um endereço. Ele **vincula** o MAC da UE ao pool. Na primeira DISCOVER DHCP daquele MAC, o servidor DHCPv4 aloca o próximo endereço livre da faixa do pool, registra um contrato e o oferece. Este é o novo caminho dinâmico de fornecimento. O endereço é escolhido no momento do DISCOVER, não no estabelecimento.



Na desmontagem da sessão, o OmniTWAG libera a alocação do pool e, para `:dhcp`, remove a vinculação do pool e o contrato dinâmico. Se o pool estiver ausente ou exaurido, ou se o AP não tiver `ip_pool_id`, o OmniTWAG não atribui nenhum endereço e a UE se auto-endereça.

DHCPv4

O servidor DHCPv4 escuta na porta UDP `67`. Ele armazena cada contrato em uma tabela ETS na memória, chaveada pelo endereço MAC da UE. Um contrato chega à tabela por um dos dois caminhos.

- **Pré-alocado.** O processo da sessão insere um contrato quando o PGW aloca um endereço (tunelado) ou quando o OmniTWAG aloca um endereço fixo do pool (NSWO `address_source: :pool`).
- **Fornecido no DISCOVER.** Para NSWO `address_source: :dhcp`, o processo da sessão em vez disso vincula o MAC da UE a um pool. O servidor mantém uma vinculação por MAC na memória. Na primeira DISCOVER de um MAC vinculado sem contrato, o servidor aloca o próximo endereço livre do pool, registra o contrato e o oferece.

Ciclo de vida do contrato

Para um contrato pré-alocado, o servidor serve o endereço que já existe na tabela para o MAC solicitante. Para um MAC vinculado, o servidor cria o contrato no DISCOVER fornecendo do pool.

Parse error on line 2: ...ar (PGW PAA ou pool :pool) [*] --> V ----------^ Expecting 'SPACE', 'NL', 'HIDE_EMPTY', 'scale', 'COMPOSIT_STATE', 'STRUCT_STOP', 'STATE_DESCR', 'ID', 'FORK', 'JOIN', 'CHOICE', 'CONCURRENT', 'note', 'acc_title', 'acc_descr', 'acc_descr_multiline_value', 'CLICK', 'classDef', 'style', 'class', 'direction_tb', 'direction_bt', 'direction_rl', 'direction_lr', 'EDGE_STATE', got 'DESCR'

Tentar novamente

OmniTWAG lida com esses tipos de mensagens DHCP.

Mensagem	Código do tipo	Comportamento do OmniTWAG
DISCOVER	1	Se um contrato existir para o MAC, o OmniTWAG envia uma OFFER. Se nenhum contrato existir, mas o MAC estiver vinculado a um pool, o OmniTWAG fornece o próximo endereço livre do pool, registra um contrato e envia uma OFFER. Se não houver contrato e nenhuma vinculação, o OmniTWAG registra um aviso e não envia nada.
REQUEST	3	Se um contrato existir e o IP solicitado (opção 50) corresponder ao contrato, ou a solicitação não tiver IP solicitado, o OmniTWAG envia um ACK. Se o IP solicitado não corresponder, o OmniTWAG envia um NAK. Se nenhum contrato existir, o OmniTWAG envia um NAK.
RELEASE	7	O OmniTWAG marca o contrato como `liberado`. O OmniTWAG mantém o registro do contrato. A conexão PDN permanece ativa, conforme TS 23.402 §16.1.5.2 NOTA 2.
DECLINE	4	O OmniTWAG registra um aviso. O OmniTWAG mantém o registro do contrato.
INFORM	8	O OmniTWAG envia um ACK com configuração apenas. O ACK carrega o gateway, a máscara de sub-rede e o DNS, mas nenhum endereço (`yiaddr` é `0.0.0.0`).

Para INFORM, o OmniTWAG usa o `default_gateway` e `default_subnet_mask` da configuração. O OmniTWAG usa `8.8.8.8` como o servidor DNS na resposta INFORM.

Conteúdos do contrato

Cada contrato contém o IP da UE, a máscara de sub-rede, o gateway, os servidores DNS e o tempo de contrato. Para uma sessão tunelada, o processo da sessão define o IP da UE e os servidores DNS a partir da resposta do PGW, e a máscara de sub-rede, o gateway e o tempo de contrato a partir da configuração DHCP. Para uma sessão NSWO, esses valores vêm do pool de IP: o IP da faixa do pool, a máscara de sub-rede do prefixo CIDR do pool e o gateway, servidores DNS e tempo de contrato do registro do pool. Veja Pools de IP e Configuração.

Um registro de contrato também tem um tempo `allocated_at` e um tempo `expires_at`. O tempo `expires_at` é `allocated_at` mais o tempo de contrato. O OmniTWAG usa esses valores para relatar o status do contrato. Um reaper em segundo plano é executado a cada 60 segundos e exclui qualquer contrato cujo `expires_at` tenha passado. Um contrato com um tempo de contrato de zero não tem `expires_at` e nunca expira. Um contrato também é removido quando a sessão é desmontada.

Opções DHCP em uma resposta

O OmniTWAG constrói cada OFFER e ACK com essas opções DHCP.

Opção	Código	Conteúdo
Tipo de Mensagem	53	OFFER ou ACK.
Identificador do Servidor	54	O `server_ip` DHCP.
Máscara de Sub-rede	1	A máscara de sub-rede do contrato.
Roteador	3	O gateway do contrato.
Servidor de Nome DNS	6	Os servidores DNS do contrato. Omitido se a lista estiver vazia.
Tempo de Contrato de Endereço IP	51	O tempo de contrato em segundos. Omitido se o tempo de contrato for zero.
Endereço de Broadcast	28	Calculado a partir do IP e da máscara de sub-rede. Enviado apenas se a UE listar esta opção em sua Lista de Solicitação de Parâmetros (opção 55).

Transporte DHCP sobre GRE

Um ponto de acesso que usa um túnel GRE encaminha a solicitação DHCP da UE dentro de um quadro Ethernet encapsulado em GRE. O servidor GRE extrai a carga útil DHCP e a passa para o servidor DHCPv4. O OmniTWAG marca o pacote com o endereço do AP de origem. Quando o OmniTWAG constrói a resposta, ele envolve a carga útil DHCP em um quadro UDP/IP/Ethernet e envia o quadro de volta através do túnel GRE para o mesmo AP. A origem é `10.0.0.1:67` e o destino é o endereço de broadcast na porta do cliente `68`.

Uma solicitação DHCP que chega diretamente ao socket UDP (não através do GRE) é respondida no socket UDP. Se a porta de origem não for a porta padrão

do cliente `68`, o OmniTWAG responde ao endereço e porta de origem. Este caminho suporta ferramentas de teste e relés. Caso contrário, o OmniTWAG segue RFC 2131: ele usa o endereço de broadcast se a flag de broadcast estiver definida ou `ciaddr` for zero, e usa `ciaddr` se esse endereço estiver definido.

DHCPv6

O servidor DHCPv6 escuta na porta UDP `547` sobre IPv6. Ele suporta tanto um modo sem estado quanto um modo com estado.

- **Modo sem estado** responde a uma Solicitação de Informação (tipo 11) com uma Resposta de Informação (tipo 12). A resposta carrega os servidores DNS. Isso segue RFC 3736 e RFC 3646. Uma UE usa este modo quando define a flag de Outra Configuração (O) a partir de seu processamento de RA.
- **Modo com estado** responde a uma Solicitação (tipo 1) com um Anúncio (tipo 2), e uma Solicitação (tipo 3) com uma Resposta (tipo 7). Este modo entrega um endereço IPv6 pré-alocado em uma opção IA_NA. Isso segue RFC 3315.

O servidor pré-aloca um endereço com estado por DUID do cliente. O plano de controle GTP insere a alocação. O servidor mantém essas alocações em uma tabela na memória, chaveada pelo DUID do cliente.

Manipulação de mensagens DHCPv6

Mensagem	Tipo	Comportamento do OmniTWAG
Solicitação de Informação	11	O OmniTWAG envia uma Resposta de Informação com o DUID do Servidor, o Client ID ecoado e os servidores DNS.
Solicitação	1	O OmniTWAG envia um Anúncio. Se o DUID do cliente tiver uma alocação, o Anúncio carrega o endereço IA_NA. Se não, o Anúncio carrega um status `NoAddrsAvail` (código 2).
Solicitação	3	O OmniTWAG envia uma Resposta. A Resposta confirma o endereço IA_NA para o DUID do cliente.
Outros tipos	-	O OmniTWAG ignora a mensagem.

Cada resposta ecoa o ID de transação do cliente, conforme RFC 3315 §15. Cada resposta que inclui um endereço alocado define os temporizadores DHCPv6 a partir do tempo de contrato. T1 é metade do tempo de contrato. T2 é quatro quintos do tempo de contrato.

DUID do Servidor

OmniTWAG constrói um DUID do Servidor do tipo DUID-EN (Número da Empresa). Ele usa o número da empresa 3GPP `10415`. O identificador é derivado do nome do nó, portanto é estável para um dado nó.

Opções DHCPv6 em uma resposta

Opção	Código	Conteúdo
Identificador do Cliente	1	O Client ID ecoado da solicitação, se presente.
Identificador do Servidor	2	O DUID do Servidor.
Associação de Identidade para Endereços Não Temporários (IA_NA)	3	O endereço IPv6 alocado, ou um status `NoAddrsAvail`. Somente no modo com estado.
Servidor de Nome DNS Recursivo	23	Os servidores DNS configurados. Omitido se a lista estiver vazia.
Lista de Pesquisa de Domínio	24	A lista de pesquisa configurada. Enviada apenas se `dhcpv6_domain_list` estiver definida.

Anúncio de Roteador IPv6 (SLAAC)

O servidor RA IPv6 entrega prefixos IPv6 alocados pelo PGW para UEs com Autoconfiguração de Endereço Sem Estado (SLAAC). Ele implementa o Anúncio de Roteador ICMPv6 conforme RFC 4861.

O plano de controle GTP adiciona um prefixo por interface. O servidor mantém os prefixos na memória, agrupados por nome de interface. O servidor envia um Anúncio de Roteador em dois casos:

1. Em um temporizador periódico. O intervalo padrão é `30` segundos.
2. Sob demanda, quando o plano de controle solicita um RA em uma interface.

O servidor também analisa mensagens de Solicitação de Roteador (ICMPv6 tipo 133).

Conteúdos do Anúncio de Roteador

Cada Anúncio de Roteador carrega esses campos e opções.

Campo ou opção	Valor
Limite de Salto Atual	De `hop_limit`. Padrão `64`.
Tempo de Vida do Roteador	De `router_lifetime`. Padrão `1800` segundos.
Flags M e O	Ambas `0`. O RA não sinaliza configuração de endereço gerenciado ou outra configuração.
Endereço de Camada de Link de Origem (opção 1)	O MAC de origem, se for um valor de 6 bytes.
MTU (opção 5)	Incluído apenas se um MTU for fornecido.
Informação de Prefixo (opção 3)	Uma por prefixo. As flags L (em link) e A (autônoma) estão ambas definidas.
Servidor DNS Recursivo, RDNSS (opção 25)	Os servidores DNS dos prefixos, conforme RFC 8106. Omitido se não houver servidores DNS.

Cada opção de Informação de Prefixo carrega o prefixo, o comprimento do prefixo, o tempo de vida válido (padrão `86400` segundos) e o tempo de vida preferido (padrão `14400` segundos). Como a flag A está definida, a UE forma seu próprio endereço IPv6 a partir do prefixo. O tempo de vida do RDNSS é o maior tempo de vida válido entre os prefixos.

Entrega de DNS e PCO

OmniTWAG entrega informações de DNS e P-CSCF de mais de uma maneira. A fonte é o PCO do PGW sempre que possível, e a configuração caso contrário.

PCO do PGW

O analisador PCO lê as Opções de Configuração de Protocolo da Resposta de Criação de Sessão, conforme 3GPP TS 24.008. O analisador retorna esses campos.

Campo	ID do protocolo PCO	Significado
`dns_ipv4`	`0x000D`	Endereços de servidores DNS IPv4.
`dns_ipv6`	`0x0003`	Endereços de servidores DNS IPv6.
`pcscf_ipv4`	`0x000C`	Endereços P-CSCF IPv4.
`pcscf_ipv6`	`0x0001`	Endereços P-CSCF IPv6.
`mtu`	`0x0010`	MTU de link IPv4.
`ip_alloc_nas`	`0x000A`	Uma flag. É `true` se o PGW sinaliza alocação de IP através de sinalização NAS.

O analisador também mantém cada opção analisada em uma lista `raw`. Se o binário PCO estiver malformado em algum ponto, o analisador retorna os campos que leu até agora. Ele não falha na análise completa.

OmniTWAG também pode construir uma solicitação PCO para a Solicitação de Criação de Sessão. A solicitação lista as opções que o OmniTWAG deseja que o PGW retorne, por exemplo, os servidores DNS IPv4 e o MTU do link.

Entrega de DNS para a UE

Os servidores DNS chegam à UE pelo transporte que a UE utiliza.

Transporte	Fonte DNS	Notas
DHCPv4 (opção 6)	Os servidores DNS IPv4 do PCO.	Se o PCO não tiver servidor DNS IPv4, o OmniTWAG usa `8.8.8.8`.
DHCPv6 (opção 23)	A lista de servidores DNS DHCPv6.	Definida a partir do PCO do PGW ou de `dhcpv6_dns_servers`. Padrão para DNS Público do Google IPv6 se não definido.
RA IPv6 (RDNSS, opção 25)	Os servidores DNS IPv6 anexados ao prefixo anunciado.	Entregue com o prefixo.

O contrato DHCPv4 pega os servidores DNS IPv4 da lista PCO `dns_ipv4`. Se essa lista estiver vazia, o OmniTWAG recorre a `8.8.8.8` para o contrato.

> **Nota sobre P-CSCF.** O analisador PCO lê os endereços P-CSCF do PGW. Os caminhos de entrega atuais de DHCP, DHCPv6 e RA não adicionam uma opção DHCP P-CSCF ou opção RA. Os valores P-CSCF são analisados e estão disponíveis, mas não são entregues à UE através deste subsistema no código auditado.

Pools de endereços e exaustão

O comportamento de exaustão depende do modo.

- **Tunelado (roteado para casa).** Não há pool de endereços local. O PGW possui o espaço de endereços. O OmniTWAG entrega exatamente o endereço que o PGW aloca para cada sessão. Não há faixa local para

exaurir. Se o PGW não retornar um PAA, a UE não recebe nenhum endereço, e o OmniTWAG registra a condição.

- **NSWO.** O OmniTWAG aloca de um pool de IP local, então um pool pode se esgotar. Um pool é um CIDR IPv4 de no máximo um `/30`. O alocador ignora os endereços de rede e de broadcast e entrega o primeiro host livre. Quando todos os hosts estão em uso, o alocador retorna `pool_exhausted`. Para `address_source: :pool`, a UE então se auto-endereça. Para `address_source: :dhcp`, o DISCOVER não recebe resposta e o OmniTWAG registra a exaustão. As alocações estão na memória e são reconstruídas à medida que as sessões se re-estabelecem, assim como os contratos DHCP. Veja Pools de IP para definição e capacidade do pool.

O helper `Twag.Cidr` faz a matemática de sub-rede para a plataforma. Ele analisa strings CIDR IPv4 e IPv6 e testa se um endereço está dentro de uma sub-rede. O subsistema RADIUS usa esse helper para autorização de IP de origem. O armazenamento do pool de IP o utiliza para derivar a máscara de sub-rede do pool, a faixa de hosts utilizáveis e o gateway padrão.

Configuração

Os servidores de gerenciamento de IP leem sua configuração do config da aplicação `:omnitwag`. Os principais blocos são `dhcp` e `ipv6_ra`. O DHCPv6 lê um pequeno conjunto de chaves de nível superior.

```
config :omnitwag,
  # Servidor DHCPv4
  dhcp: %{
    enabled: true,
    interface: "wlan0",
    server_ip: "10.0.0.1",
    default_lease_time: 3600,
    default_subnet_mask: "255.255.255.0",
    default_gateway: "10.0.0.1"
  },

  # Anúncio de Roteador IPv6
  ipv6_ra: %{
    enabled: true,
    interface: "wlan0",
    ra_interval: 30,
    hop_limit: 64,
    router_lifetime: 1800
  },

  # DHCPv6 (chaves de nível superior opcionais)
  dhcpv6_dns_servers: ["2001:4860:4860::8888",
"2001:4860:4860::8844"],
  dhcpv6_domain_list: ["example.com"]
```

Parâmetros DHCPv4 (`dhcp`)

Parâmetro	Tipo	Necessário	Padrão	Desc
`enabled`	Booleano	Não	`true`	Habilit servid DHCPv Defina `false` desabi por exe em um ambier teste.
`interface`	String	Não	`wlan0`	Interfa rede pa serviç DHCP. `gre` pa modo túnel G
`server_ip`	String	Não	`10.0.0.1`	O identifi do serv DHCP. OmniT envia e valor n opção 54 e c origem respos encaps em GR

Parâmetro	Tipo	Necessário	Padrão	Desc
`default_lease_time`	Inteiro	Não	`3600`	Tempo contrat segund OmniT aplica valor a contrat alocad
`default_subnet_mask`	String	Não	`255.255.255.0`	Máscar sub-re para o contrat para o cálculo broadc confor RFC 21
`default_gateway`	String	Não	`10.0.0.1`	Gatew padrã o contr (opção 3).

Parâmetros RA IPv6 (`ipv6_ra`)

Parâmetro	Tipo	Necessário	Padrão	Descrição
`enabled`	Booleano	Não	`true`	Habilita o servidor RA IPv6. Defina como `false` para desabilitá-lo.
`interface`	String	Não	`wlan0`	Interface de rede para Anúncios de Roteador.
`ra_interval`	Inteiro	Não	`30`	Intervalo em segundos entre Anúncios de Roteador periódicos.
`hop_limit`	Inteiro	Não	`64`	Campo Limite de Salto Atual no Anúncio de Roteador.
`router_lifetime`	Inteiro	Não	`1800`	Tempo de Vida do Roteador em segundos no Anúncio de Roteador.

Parâmetros DHCPv6 (nível superior)

Parâmetro	Tipo	Necessário	Padrão	D
`dhcpv6_dns_servers`	Lista de String	Não	DNS Público do Google IPv6 (`2001:4860:4860::8888`, `2001:4860:4860::8844`)	Se DI pa re DI PC PC ta po de es en de ex
`dhcpv6_domain_list`	Lista de String	Não	(não definido)	Li pe DI a DI (o 24 nã de O or op

Observabilidade

Interface Web

O painel de controle tem uma página **DHCP Pool** em `/dhcp`. A página lista cada contrato DHCPv4 com o endereço MAC da UE, o endereço IP, o horário de início do contrato, o horário de expiração do contrato e o status. O status é `Ativo` ou `Expirado`. A página mostra uma contagem regressiva ao vivo para cada contrato ativo. Selecione uma linha de contrato para ver os detalhes completos do contrato: máscara de sub-rede, gateway, servidores DNS, duração do contrato e timestamps. A página é atualizada a cada segundo. Se o servidor DHCPv4 não estiver em execução, a página mostra `Servidor Não Está Executando`.

API REST

A API REST serve dados de contratos DHCPv4 sob a tag `DHCP`.

Método	Caminho	Descrição
GET	`/dhcp` (índice)	Lista todos os contratos DHCPv4 com uma contagem resumida.
GET	`/dhcp/{id}` (mostrar)	Obtém um contrato por endereço MAC. O `id` é o MAC separado por dois pontos.

A resposta do índice tem uma flag `enabled`, a lista `leases`, uma `count` total, uma contagem `active` e uma contagem `expired`. Se o servidor DHCPv4 não estiver em execução, o índice retorna `enabled: false` com uma lista vazia. O endpoint de mostrar retorna `404` se nenhum contrato corresponder ao MAC, e `503` se o servidor não estiver em execução.

Cada objeto de contrato tem esses campos.

Campo	Descrição
`mac`	Endereço MAC da UE, separado por dois pontos.
`ip`	Endereço IPv4 atribuído.
`subnet_mask`	Máscara de sub-rede do contrato.
`gateway`	Gateway do contrato.
`dns_servers`	Lista de servidores DNS.
`lease_time`	Duração do contrato em segundos.
`allocated_at`	Horário de início do contrato, em segundos desde a época Unix.
`expires_at`	Horário de expiração do contrato, em segundos desde a época Unix.
`status`	`ativo` ou `expirado`.
`time_remaining`	Segundos até a expiração, ou `0` se expirado.

Eventos de Telemetria

Os servidores de gerenciamento de IP emitem eventos de telemetria. Um operador pode anexar esses eventos a um exportador de métricas. Veja a Referência de Métricas para o endpoint Prometheus.

Evento	Fonte	Significado
`[:dhcp, :lease, :allocate]`	DHCPv4	Um contrato foi pré-alocado para um MAC.
`[:dhcp, :lease, :deallocate]`	DHCPv4	Um contrato foi removido na desmontagem.
`[:dhcp, :lease, :reap]`	DHCPv4	Um contrato expirado foi removido pelo reaper em segundo plano.
`[:dhcp, :discover]`	DHCPv4	Um DISCOVER foi respondido com uma OFFER de um contrato existente.
`[:dhcp, :discover, :pool_allocated]`	DHCPv4	Um DISCOVER de um MAC vinculado ao pool foi respondido fornecendo um novo endereço do pool (NSWO `:dhcp`).
`[:dhcp, :discover, :no_allocation]`	DHCPv4	Um DISCOVER chegou para um MAC sem contrato e sem vinculação ao pool.
`[:dhcp, :ack]`	DHCPv4	Uma REQUEST foi respondida com um ACK.
`[:dhcp, :nak]`	DHCPv4	Uma REQUEST foi respondida com um NAK.
`[:dhcp, :release]`	DHCPv4	Um cliente liberou seu contrato.
`[:dhcp, :decline]`	DHCPv4	Um cliente recusou seu contrato.
`[:dhcp, :inform]`	DHCPv4	Um INFORM foi respondido.
`[:dhcp, :packet, :malformed]`	DHCPv4	Um pacote DHCP malformado foi recebido.

Evento	Fonte	Significado
`[:dhcpv6, :info_request, :received]`	DHCPv6	Uma Solicitação de Informação foi recebida.
`[:dhcpv6, :solicit, :received]`	DHCPv6	Uma Solicitação foi recebida.
`[:dhcpv6, :request, :received]`	DHCPv6	Uma Solicitação foi recebida.
`[:dhcpv6, :info_reply, :sent]`	DHCPv6	Uma Resposta de Informação foi enviada.
`[:dhcpv6, :advertise, :sent]`	DHCPv6	Um Anúncio foi enviado.
`[:dhcpv6, :reply, :sent]`	DHCPv6	Uma Resposta foi enviada.
`[:ipv6_ra, :prefix, :add]`	RA IPv6	Um prefixo foi adicionado a uma interface.
`[:ipv6_ra, :prefix, :remove]`	RA IPv6	Um prefixo foi removido de uma interface.
`[:ipv6_ra, :advertisement, :sent]`	RA IPv6	Um Anúncio de Roteador foi enviado.

Solução de Problemas

A UE não recebe endereço IP

Sintomas: A UE se autentica, mas não recebe um endereço IPv4. A página do Pool DHCP mostra nenhum contrato para o MAC da UE. O evento `[:dhcp, :discover, :no_allocation]` é acionado.

Possíveis causas (tunelado):

- A resposta de criação de sessão GTP não teve PAA, então o OmniTWAG não pré-alocou um contrato.
- A sessão GTP-C não foi estabelecida, então nenhum contrato foi criado.

Possíveis causas (NSWO):

- A credencial do AP não tem `ip_pool_id`, então o OmniTWAG não atribui nenhum endereço.
- O pool está exaurido, então nenhum endereço livre pode ser fornecido.
- Para `address_source: :dhcp`, o DISCOVER chegou antes que o MAC fosse vinculado ao pool, ou a vinculação não foi executada porque a sessão ainda não tinha MAC.

Causa comum:

- O servidor DHCPv4 não está em execução.

Resolução:

1. Verifique os logs do OmniTWAG para o caminho de estabelecimento (tunelado versus NSWO) e o IP da UE.
2. Para sessões tuneladas, confirme que o PGW retorna um PAA para o APN.
3. Para sessões NSWO, confirme que a credencial do AP nomeia um `ip_pool_id` válido e que o pool tem capacidade livre. Veja Pools de IP e Credenciais RADIUS.
4. Confirme que a página do Pool DHCP mostra o servidor DHCPv4 como em execução.

A UE recebe um endereço, mas sem DNS

Sintomas: A UE tem um endereço IPv4, mas não consegue resolver nomes.

Possíveis causas:

- O PCO do PGW não retornou servidor DNS IPv4. Nesse caso, o OmniTWAG usa `8.8.8.8`.
- A UE não consegue alcançar o servidor DNS que o PGW forneceu.

Resolução:

1. Leia os servidores DNS do contrato na página do Pool DHCP ou através da API REST.
2. Confirme que o PCO do PGW retorna o servidor DNS esperado.
3. Confirme que o servidor DNS é acessível a partir do caminho de dados da UE.

A UE recebe um NAK

Sintomas: A UE envia um REQUEST e recebe um NAK. O evento `[:dhcp, :nak]` é acionado.

Possíveis causas:

- A UE solicitou um IP (opção 50) que não corresponde ao seu contrato pré-alocado.
- O contrato foi removido, por exemplo, após uma desmontagem de sessão.

Resolução:

1. Confirme que um contrato existe para o MAC da UE.
2. Confirme que a UE solicita o mesmo IP que o PGW ou o pool alocou.
3. Se a sessão foi desmontada, deixe a UE reiniciar a troca DHCP após se re-autenticar.

Páginas relacionadas

- Visão Geral
- Pools de IP
- Credenciais RADIUS
- Sessions
- S2a GTP-C
- User Plane
- Cobrança Online (Gy)

Visão Geral

Pools de IP

Visão Geral

OmniTWAG atribui endereços IPv4 a sessões NSWO (Non-Seamless WLAN Offload) a partir de um **pool de IP**. NSWO é um breakout local ancorado no TWAG: OmniTWAG termina o tráfego do usuário e o quebra localmente. Um pool de IP define um bloco CIDR IPv4, um gateway, servidores DNS e um tempo de locação. O operador adiciona, altera e remove pools sob demanda através da API REST ou da interface web do painel de controle.

As definições de pool são persistentes. OmniTWAG as armazena em um arquivo DETS, de modo que os pools sobrevivem a uma reinicialização. As alocações de endereços são mantidas na memória. OmniTWAG reconstrói as alocações à medida que as sessões se restabelecem após uma reinicialização.

Índice

- Conceitos
- Como uma sessão obtém um endereço
- Alocação de endereços
- Modelo de pool
- API REST
- Interface web
- Solução de problemas

Conceitos

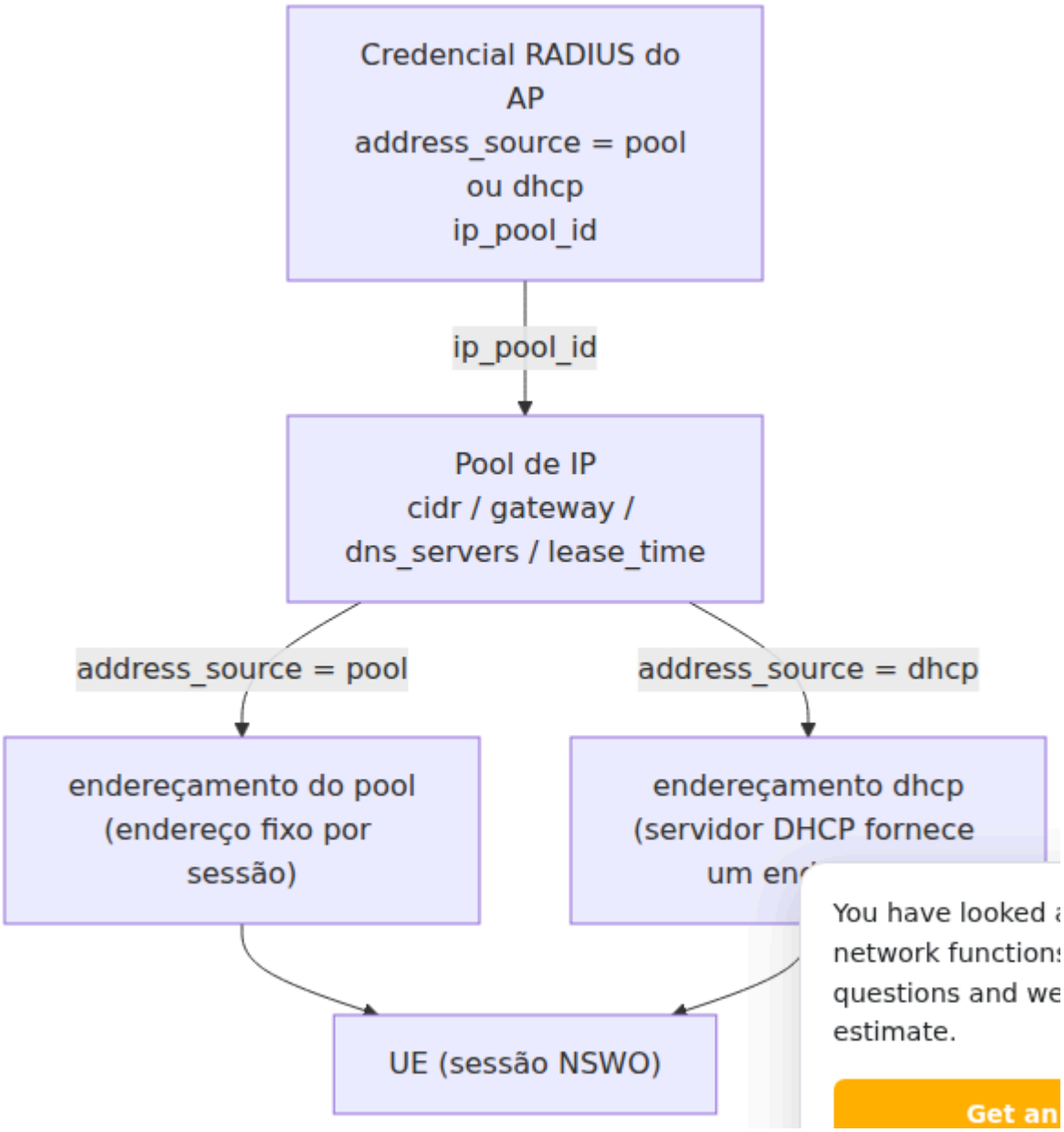
Um **pool** é um conjunto de endereços de um bloco CIDR IPv4. Cada pool tem um gateway, uma lista de servidores DNS e um tempo de locação.

Um pool de IP está vinculado a um ponto de acesso através das credenciais RADIUS do AP. A credencial possui um campo `ip_pool_id`. Este campo contém o `id` do pool. Veja Segredos Compartilhados RADIUS para o modelo de

credenciais. As sessões NSWO desse ponto de acesso são endereçadas a partir do pool vinculado.

Cada credencial também possui um campo `address_source`. Este campo seleciona como o OmniTWAG fornece o endereço ao UE (equipamento do usuário). Existem duas fontes de endereço:

Fonte de endereço	Comportamento
`pool`	OmniTWAG pré-aloca um endereço fixo por sessão a partir do pool. O endereço é escolhido quando a sessão é configurada.
`dhcp`	O servidor DHCP do OmniTWAG fornece um endereço a partir da faixa do pool em resposta a um DHCP DISCOVER do UE.

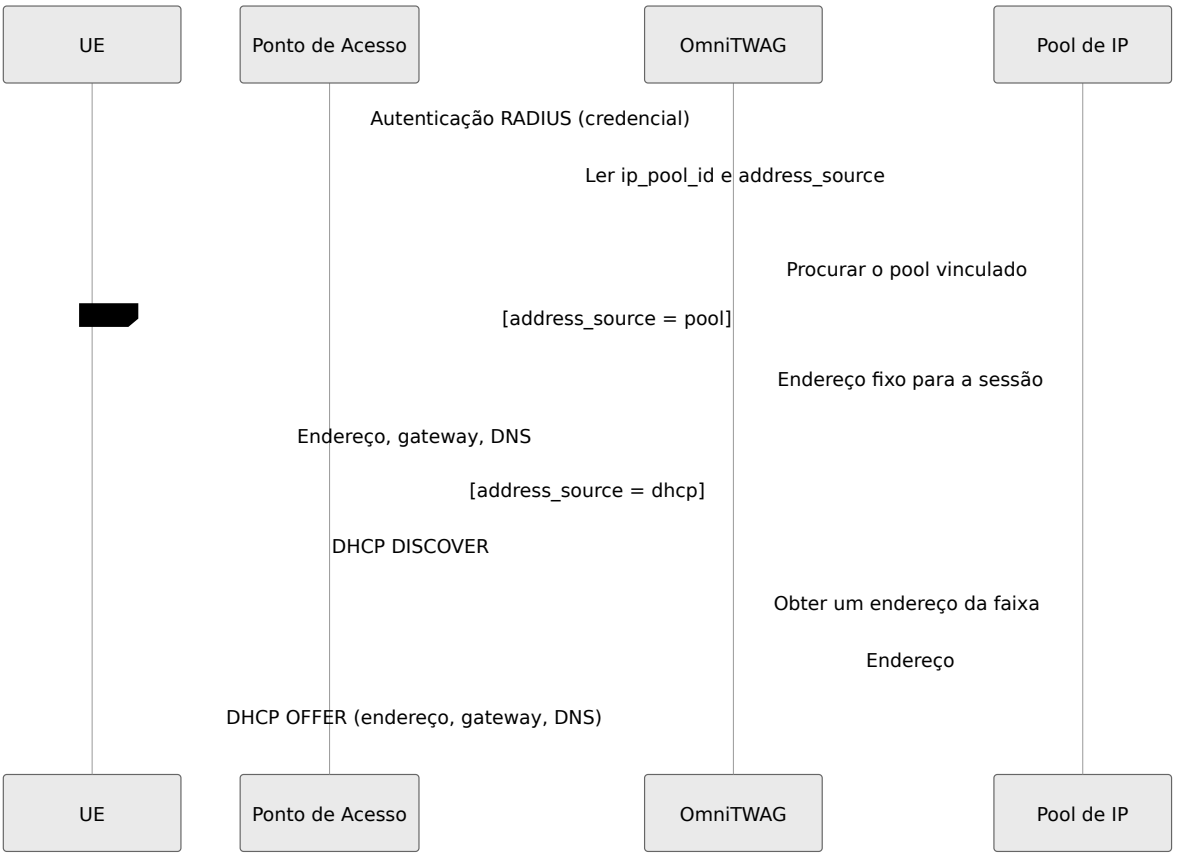


Como uma sessão obtém um endereço

OmniTWAG endereça cada sessão NSWO a partir do pool que está vinculado à credencial do AP. Os passos são:

1. O ponto de acesso se autentica através de sua credencial RADIUS.
2. OmniTWAG lê o campo `ip_pool_id` na credencial para encontrar o pool.
3. OmniTWAG lê o campo `address_source` na credencial.

4. Se `address_source` for `pool`, OmniTWAG pré-aloca um endereço fixo para a sessão a partir do pool.
5. Se `address_source` for `dhcp`, OmniTWAG aguarda um DHCP DISCOVER do UE. O servidor DHCP então fornece um endereço a partir da faixa do pool.
6. OmniTWAG fornece ao UE o endereço, o gateway e os servidores DNS do pool.



Alocação de endereços

OmniTWAG aloca hosts a partir do CIDR do pool. A alocação ignora o endereço de rede e o endereço de broadcast. O primeiro host utilizável é o endereço de rede mais um.

O prefixo CIDR do pool deve ser `/30` ou maior para ter hosts utilizáveis. Um bloco `/30` tem 2 hosts utilizáveis. Um prefixo menor tem menos hosts utilizáveis, portanto, OmniTWAG o rejeita.

Quando todos os hosts utilizáveis estão alocados, OmniTWAG retorna `pool_exhausted`. Nenhuma nova sessão do ponto de acesso vinculado pode

obter um endereço até que uma alocação seja liberada.

O campo `allocated` no pool mantém a contagem atual de alocações. OmniTWAG calcula essa contagem a partir das alocações na memória.

Modelo de pool

Cada pool tem estes campos.

Campo	Tipo	Descrição
`id`	String	Identificador único (UUID). O servidor define este valor.
`name`	String	Rótulo legível por humanos, por exemplo `WiFi para convidados de Melbourne`.
`cidr`	String	Bloco CIDR IPv4 para o pool, por exemplo `10.60.0.0/24`. O prefixo deve ser `/30` ou maior para ter hosts utilizáveis.
`gateway`	String	Endereço de gateway IPv4 fornecido ao UE. Por padrão, é o primeiro host utilizável do CIDR se omitido.
`dns_servers`	Lista de String	Endereços de servidores DNS IPv4 fornecidos ao UE.
`lease_time`	Inteiro	Tempo de locação em segundos. Por padrão, é `3600`.
`default`	Booleano	Quando `true`, este pool é a faixa padrão do DHCP. O servidor DHCP fornece a partir dele para qualquer cliente sem locação pré-alocada e sem vinculação de pool. Apenas um pool pode ser o padrão ao mesmo tempo. Por padrão, é `false`.
`allocated`	Inteiro	Contagem das alocações atuais. O servidor calcula este valor.
`inserted_at`	Inteiro	Tempo em que o pool foi criado, em milissegundos desde a época Unix. O servidor define este valor.

Campo	Tipo	Descrição
`updated_at`	Inteiro	Tempo em que o pool foi alterado pela última vez, em milissegundos desde a época Unix. O servidor define este valor.

O operador define `name`, `cidr`, `gateway`, `dns_servers` e `lease_time`. O servidor define `id`, `allocated`, `inserted_at` e `updated_at`.

API REST

A API REST do OmniTWAG serve estes endpoints sobre HTTPS na porta API configurada (`8444` por padrão). O caminho base é `/ip_pools`.

Método	Caminho	Ação	Descrição
GET	`/ip_pools`	index	Lista todos os pools. O resultado é paginado e pesquisável por `name` ou `cidr`.
GET	`/ip_pools/{id}`	show	Obtém um pool por `id`.
POST	`/ip_pools`	create	Cria um pool.
PUT	`/ip_pools/{id}`	update	Altera um pool.
DELETE	`/ip_pools/{id}`	delete	Remove um pool.

Corpo da solicitação

`POST` e `PUT` aceitam um corpo JSON com estes campos.

Campo	Tipo	Obrigatório	Descrição
`name`	String	Não	Rótulo legível por humanos.
`cidr`	String	Sim	Bloco CIDR IPv4. O prefixo deve ser `/30` ou maior.
`gateway`	String	Não	Endereço de gateway IPv4. Por padrão, é o primeiro host utilizável do CIDR se omitido.
`dns_servers`	Lista de String	Não	Endereços de servidores DNS IPv4.
`lease_time`	Inteiro	Não	Tempo de locação em segundos. Por padrão, é `3600`.
`default`	Booleano	Não	Marcar este pool como a faixa padrão do DHCP. Defini-lo limpa a marca em qualquer outro pool.

`PUT` altera apenas os campos no corpo. Campos que não estão no corpo mantêm seu valor atual.

Respostas

Status	Significado
`200 OK`	A solicitação foi bem-sucedida (index, show, update, delete).
`201 Created`	O pool foi criado.
`404 Not Found`	Nenhum pool possui o `id` fornecido.
`422 Unprocessable Entity`	O `cidr` não é um bloco CIDR IPv4 válido ou o prefixo é muito pequeno para ter hosts utilizáveis.

Exemplo: listar pools

Solicitação:

```
GET /ip_pools
```

Resposta:

```
{
  "count": 1,
  "ip_pools": [
    {
      "id": "0d1e2f3a-4b5c-6d7e-8f90-1a2b3c4d5e6f",
      "name": "WiFi para convidados de Melbourne",
      "cidr": "10.60.0.0/24",
      "gateway": "10.60.0.1",
      "dns_servers": ["8.8.8.8", "8.8.4.4"],
      "lease_time": 3600,
      "allocated": 12,
      "inserted_at": 1734512400000,
      "updated_at": 1734512400000
    }
  ]
}
```


Exemplo: criar um pool

Solicitação:

```
POST /ip_pools
```


```
{
  "name": "WiFi para convidados de Melbourne",
  "cidr": "10.60.0.0/24",
  "gateway": "10.60.0.1",
  "dns_servers": ["8.8.8.8", "8.8.4.4"],
  "lease_time": 3600
}
```


Uma solicitação bem-sucedida retorna `201 Created` com o novo pool, incluindo seu `id`. Se `gateway` for omitido, o pool usa o primeiro host utilizável do CIDR (`10.60.0.1` neste exemplo).

Exemplo: um CIDR que é muito pequeno

Solicitação:

```
{
  "name": "Pool quebrado",
  "cidr": "10.60.0.0/31"
}
```

Resposta (`422 Unprocessable Entity`):

```
{
  "error": "cidr muito pequeno: 10.60.0.0/31 não tem hosts
utilizáveis"
}
```

Pool padrão

Um pool pode ser a faixa padrão do DHCP. O servidor DHCP fornece um endereço a partir do pool padrão para qualquer cliente que não tenha locação pré-alocada e sem vinculação explícita de pool. Use o pool padrão para fornecer uma faixa definida pela API ao serviço DHCP base, não apenas para sessões NSWO.

Defina o padrão com o campo `default`, ou a caixa de seleção na página web. Apenas um pool pode ser o padrão ao mesmo tempo. Quando você marca um pool como padrão, o OmniTWAG limpa a marca no pool padrão anterior.

Cada locação ativa registra sua fonte. Veja a visualização Locações de IP: uma locação mostra `PGW (tunneled)`, um `pool NSWO` nomeado, ou o `pool padrão`.

Interface web

O painel de controle possui uma página de administração **Pools de IP**. A página lista cada pool com seu nome, CIDR, gateway e contagem alocada. A página possui controles para adicionar, editar e excluir um pool.

A antiga página **Pool DHCP** agora é rotulada como **Locações de IP**. A página de Locações de IP mostra as loca��ões ativas.

Solução de problemas

Um pool está esgotado

Sintomas: Novas sessões NSWO de um ponto de acesso não conseguem obter um endereço. OmniTWAG retorna `pool_exhausted`.

Possíveis causas:

- Todos os hosts utilizáveis no CIDR do pool estão alocados.
- O CIDR do pool é muito pequeno para o número de sessões ativas.

Resolução:

1. Verifique a contagem `allocated` em relação ao número de hosts utilizáveis no CIDR.
2. Amplie o CIDR do pool para adicionar mais hosts utilizáveis, ou libere alocações antigas encerrando sessões ociosas.
3. Confirme que o tempo de locação não é tão longo que os endereços liberados permaneçam alocados.

Um ponto de acesso não usa um pool

Sintomas: Sessões de um ponto de acesso não são endereçadas a partir de nenhum pool de IP.

Possíveis causas:

- O ponto de acesso usa um breakout local de propriedade do ponto de acesso, não um NSWO ancorado no TWAG. Um ponto de acesso com breakout local de propriedade do AP endereça seus próprios clientes e não usa um pool OmniTWAG.
- A credencial do AP não possui `ip_pool_id`, portanto, nenhum pool está vinculado.

Resolução:

1. Confirme que o ponto de acesso usa NSWO ancorado no TWAG. Apenas sessões NSWO usam um pool.
2. Se o ponto de acesso deve usar um pool, defina `ip_pool_id` em sua credencial RADIUS para o `id` do pool. Veja Segredos Compartilhados RADIUS.

O UE não obtém endereço

Sintomas: O UE se conecta, mas não recebe nenhum endereço IPv4.

Possíveis causas:

- A fonte de endereço da credencial `address_source` não corresponde a como o UE solicita um endereço. Com `dhcp`, o UE deve enviar um DHCP DISCOVER. Com `pool`, OmniTWAG pré-aloca o endereço.
- O pool vinculado está esgotado.
- O `ip_pool_id` na credencial aponta para um pool que não existe.

Resolução:

1. Confirme que o `address_source` na credencial corresponde ao comportamento do UE.
2. Confirme que o pool vinculado tem endereços livres. Verifique a contagem `allocated`.
3. Confirme que o `ip_pool_id` na credencial aponta para um pool válido.

Visão Geral

Referência de Métricas TWAG

Visão Geral

TWAG expõe métricas do Prometheus na porta **9568** (definida por `prometheus.port`). As métricas de gauge são atualizadas por um poller a cada 5 segundos. As métricas de contador são emitidas a partir do caminho de código que as aciona.

Índice

- Endpoint de Métricas
- Métricas de Autenticação RADIUS
- Métricas de Autorização e Limite de Taxa RADIUS
- Métricas de Contabilidade RADIUS
- Métricas EAP
- Gauges de Cliente, AP e Sessão
- Métricas de Throughput de Dados
- Métricas de Sessão e Caminho de Dados TWAG
- Métricas de Cobrança Online Gy
- Métricas de Diâmetro STa
- Métrica de Mensagem de Diâmetro
- Métricas NSWO
- Métricas DHCP
- Métricas DHCPv6 e RA IPv6
- Métricas WLCP
- Métricas da VM Erlang
- Consultas de Exemplo Grafana
- Regras de Alerta
- Páginas Relacionadas

Endpoint de Métricas

```
http://<twag-host>:9568/metrics
```

O Prometheus achata cada nome de evento em um nome de métrica substituindo pontos por sublinhados. Por exemplo, o evento `[:radius, :access, :request, :count]` é exposto como `radius_access_request_count`.

Métricas de Autenticação RADIUS

Esses contadores rastreiam fluxos de Access-Request/Response RADIUS por RFC 2865.

Métrica	Tipo	Rótulos	Descrição
`radius_access_request_count`	Contador	nenhum	Pacotes de RADIUS Access-Request recebidos.
`radius_access_accept_count`	Contador	nenhum	Pacotes de RADIUS Access-Accept enviados (autenticação bem-sucedida).
`radius_access_reject_count`	Contador	nenhum	Pacotes de RADIUS Access-Reject enviados (autenticação falhada).
`radius_access_challenge_count`	Contador	nenhum	Pacotes de RADIUS Access-Challenge enviados (troca EAP em andamento).

Consultas de exemplo:

```
# Taxa de requisições por segundo
rate(radius_access_request_count[5m])

# Taxa de desafios (negociações EAP ativas)
rate(radius_access_challenge_count[5m])

# Razão de sucesso da autenticação
sum(rate(radius_access_accept_count[5m])) /
  (sum(rate(radius_access_accept_count[5m])) +
sum(rate(radius_access_reject_count[5m])))
```

Métricas de Autorização e Limite de Taxa RADIUS

Esses contadores rastreiam pacotes que são descartados ou sinalizados antes ou durante a autenticação. Veja Rejeições RADIUS e Segredos Compartilhados RADIUS.

Métrica	Tipo	Rótulos	
`radius_auth_unauthorized_source_count`	Contador	`source`	Pa or co su de de
`radius_auth_bad_secret_count`	Contador	`source`, `credential_id`	Pa qu un fa M (s in
`radius_access_rate_limited_count`	Contador	`nas_ip`	Ac de ex `ra` na

Consultas de exemplo:

```
# APs desconhecidos aparecendo (adicione uma credencial para
estes)
increase(radius_auth_unauthorized_source_count[1h])

# APs com um segredo compartilhado incorreto
sum by (source) (increase(radius_auth_bad_secret_count[1h]))

# Dispositivos NAS com limite de taxa
sum by (nas_ip) (increase(radius_access_rate_limited_count[15m]))
```

Métricas de Contabilidade RADIUS

Rastreia mensagens de Contabilidade RADIUS por RFC 2866.

radius_accounting_request_count

Tipo: Contador

Rótulo	Descrição
`status_type`	Tipo de status de contabilidade (veja abaixo).

Valores do tipo de status: `start`, `stop`, `interim_update`, `accounting_on`, `accounting_off`.

Consultas de exemplo:

```
# Inícios de sessão por minuto
rate(radius_accounting_request_count{status_type="start"}[5m]) *
60

# Paradas de sessão por minuto
rate(radius_accounting_request_count{status_type="stop"}[5m]) * 60
```

Métricas EAP

Esses contadores rastreiam fases EAP-AKA por RFC 4187, além de EAP-SIM, EAP reauthenticação (ERP) e falhas genéricas de EAP.

Métrica	Tipo	Descri��ão
`eap_aka_identity_count`	Contador	Trocas de identidade EAP-AKA (Fase 1).
`eap_aka_challenge_count`	Contador	Trocas de desafio EAP-AKA (Fase 2).
`eap_aka_sync_failure_count`	Contador	Falhas de sincronização EAP-AKA (SQN fora de sincronia, aciona ressincronização).
`eap_aka_auth_success_count`	Contador	Autenticações EAP-AKA bem-sucedidas.
`eap_aka_auth_reject_count`	Contador	Autenticações EAP-AKA rejeitadas (UE falhou em autenticar a rede / falha AUTN).
`eap_aka_client_error_count`	Contador	Notificações de erro de cliente EAP-AKA recebidas do UE.
`eap_auth_rejected_count`	Contador	Negativas de autenticação esperadas (por exemplo, HSS USER_UNKNOWN) que rejeitam o UE de forma limpa.
`eap_process_crash_count`	Contador	Erros inesperados ao processar uma mensagem EAP (UE rejeitado, sessão preservada).
`eap_sim_start_count`	Contador	Trocas de início EAP-SIM.
`eap_sim_auth_success_count`	Contador	Autenticações EAP-SIM bem-sucedidas.

Métrica	Tipo	Descri��ão
`eap_sim_auth_reject_count`	Contador	Autenticações EAP-SIM rejeitadas.
`eap_erp_initiate_count`	Contador	Iniciações de reautenticação EAP ERP.
`eap_erp_success_count`	Contador	Reautenticações EAP ERP bem-sucedidas.
`eap_erp_mac_failure_count`	Contador	Falhas de verificação de MAC EAP ERP.
`eap_erp_fallback_count`	Contador	Recuos EAP ERP para autenticação completa. Rótulo `reason`.

Consultas de exemplo:

```
# Falhas de sincronização (devem ser raras - alerte se > 0)
increase(eap_aka_sync_failure_count[1h])

# Autenticações AKA bem-sucedidas por minuto
rate(eap_aka_auth_success_count[5m]) * 60

# Falhas de processamento EAP (devem permanecer constantes)
increase(eap_process_crash_count[1h])
```

Gauges de Cliente, AP e Sessão

Esses gauges são amostrados a cada 5 segundos pelo poller.

Métrica	Tipo	Rótulos	Descrição
`radius_active_clients_count`	Gauge	nenhum	Clientes atualmente autenticados (processos `Twag.Session.Client` ativos).
`radius_access_points_count`	Gauge	nenhum	Pontos de acesso registrados (entradas `APState`).
`radius_clients_per_ap_count`	Gauge	`nas_ip`, `ssid`	Clientes conectados por ponto de acesso.
`radius_active_sessions_count`	Gauge	`status`	Sessões agrupadas por status de contabilidade (`start`, `stop`, `interim_update`).
`radius_active_sessions_total`	Gauge	nenhum	Sessões em status `start` ou `interim_update`.

Consultas de exemplo:

```
# Top 5 APs mais movimentados
topk(5, radius_clients_per_ap_count)

# Clientes por SSID
sum by (ssid) (radius_clients_per_ap_count)

# Alerta sobre uma queda repentina em sessões ativas
delta(radius_active_sessions_total[5m]) < -10
```

Métricas de Throughput de Dados

Esses gauges são somados em todas as sessões rastreadas a partir dos contadores de contabilidade RADIUS.

Métrica	Tipo	Unidade	Descrição
`radius_data_bytes_in`	Gauge	bytes	Total de bytes recebidos em todas as sessões (Acct-Input-Octets).
`radius_data_bytes_out`	Gauge	bytes	Total de bytes enviados em todas as sessões (Acct-Output-Octets).
`radius_session_total_time`	Gauge	segundos	Tempo total de sessão em todas as sessões (Acct-Session-Time).

Consultas de exemplo:

```
# Throughput total em MB
(radius_data_bytes_in + radius_data_bytes_out) / 1024 / 1024

# Duração média da sessão
radius_session_total_time / radius_active_sessions_total
```

Métricas de Sessão e Caminho de Dados TWAG

Esses contadores rastreiam eventos do ciclo de vida por sessão emitidos por `Twag.Session.Client`.

Métrica	Tipo	Rótulos	Des
`twag_session_teardown_count`	Contador	`reason`	Desmont sessão. F `account` `admin_d` `network` `identity` `liveness`
`twag_bearer_modify_command_count`	Contador	nenhum	Eventos Bearer-C aplicados sessão.
`twag_bearer_update_count`	Contador	nenhum	Eventos Update-B aplicados sessão.
`twag_detach_cleanup_start_count`	Contador	nenhum	Início de desanexa
`twag_detach_cleanup_complete_count`	Contador	nenhum	Conclusõ limpeza desanexa
`twag_detach_pgw_delete_count`	Contador	nenhum	Solicitaç Delete-S durante desanexa
`twag_detach_liveness_timeout_count`	Contador	nenhum	Desanex acionada timeout

Métrica	Tipo	Rótulos	Des
`twag_liveness_arp_probe_success_count`	Contador	nenhum	Provas d liveness sucedida
`twag_liveness_arp_probe_failure_count`	Contador	nenhum	Provas d liveness
`twag_handover_in_count`	Contador	`source`	Eventos in.
`twag_handover_out_count`	Contador	`cause`	Eventos handove
`twag_handover_ap_change_count`	Contador	nenhum	Eventos de AP int

Consultas de exemplo:

```
# Razões de desmontagem na última hora
sum by (reason) (increase(twag_session_teardown_count[1h]))

# Sessões eliminadas por nunca completarem a autenticação
increase(twag_session_teardown_count{reason="identity_timeout"}
[1h])

# Sessões descartadas porque o UE ficou em silêncio
increase(twag_detach_liveness_timeout_count[1h])
```

> Nota: os eventos de cota Gy `twag_gy_quota_blocked` e `twag_gy_quota_unblocked` são emitidos por `Twag.Session.Client`, mas não estão declarados como métricas do Prometheus hoje, portanto não aparecem no endpoint `/metrics`. Eles são visíveis nos logs estruturados.

Métricas de Cobrança Online Gy

Esses contadores rastreiam o controle de crédito Diameter Ro/Gy, usado apenas para quebra local NSWO. Veja Cobrança Online.

Métrica	Tipo	Rótulos	Descrição
`twag_gy_ccr_sent_count`	Contador	`type`	Credit-Control-Requests enviados ao OCS. `type` é `initial`, `update` ou `terminate`.
`twag_gy_ccr_error_count`	Contador	`type`	Falhas de envio ou construção de CCR.
`twag_gy_cca_received_count`	Contador	`type`, `result`	Credit-Control-Answers recebidos do OCS.

Consultas de exemplo:

```
# Taxa de envio de CCR por tipo
sum by (type) (rate(twag_gy_ccr_sent_count[5m]))

# Falhas de CCR
sum by (type) (increase(twag_gy_ccr_error_count[1h]))

# Resultados de CCA
sum by (result) (rate(twag_gy_cca_received_count[5m]))
```

Métricas de Diâmetro STa

Esses contadores rastreiam solicitações de entrada STa tratadas pelo TWAG.

Métrica	Tipo	Rótulos	Descriçã
`sta_asr_received_count`	Contador	nenhum	Abort-Sessio Requests recebidos.
`sta_rar_received_count`	Contador	nenhum	Re-Auth- Requests recebidos.
`sta_str_received_count`	Contador	nenhum	Session- Termination- Requests recebidos.
`sta_session_disconnect_count`	Contador	`reason`	Desconexões sessão STa.
`sta_session_reauth_count`	Contador	`re_auth_type`	Reautenticaç de sessão ST

Métrica de Mensagem de Diâmetro

diameter_message_count

Tipo: Contador

Rótulo	Descrição
`application`	Aplicação Diameter (por exemplo, `swx`, `sta`).
`command`	Nome do comando (por exemplo, `MAR`, `MAA`).
`direction`	`inbound` ou `outbound`.

Consultas de exemplo:

```
# Taxa de mensagens Diameter por aplicação
sum by (application) (rate(diameter_message_count[5m]))
```

Métricas NSWO

Métrica	Tipo	Descrição
`nswo_allocate_count`	Contador	Alocações de IP local NSWO.
`nswo_release_count`	Contador	Liberações de IP local NSWO.

Métricas DHCP

Esses contadores rastreiam o servidor DHCPv4. O evento `discover.pool_allocated` cobre o caminho de distribuição NSWO, onde o servidor aloca um novo endereço de um pool vinculado no DISCOVER.

Métrica	Tipo	Descrição
`dhcp_lease_allocate_count`	Contador	Alocações de lease.
`dhcp_lease_deallocate_count`	Contador	Desalocações de lease.
`dhcp_discover_count`	Contador	Pacotes DISCOVER servidos a partir de um lease pré-alocado.
`dhcp_discover_no_allocation_count`	Contador	DISCOVERs sem lease e sem pool vinculado (sem endereço para oferecer).
`dhcp_discover_pool_allocated_count`	Contador	DISCOVERs servidos ao distribuir um novo endereço de um pool vinculado (NSWO).
`dhcp_ack_count`	Contador	Pacotes ACK enviados.
`dhcp_nak_count`	Contador	Pacotes NAK enviados.
`dhcp_release_count`	Contador	Pacotes RELEASE recebidos.
`dhcp_decline_count`	Contador	Pacotes DECLINE recebidos.
`dhcp_inform_count`	Contador	Pacotes INFORM recebidos.

Métrica	Tipo	Descrição
`dhcp_packet_malformed_count`	Contador	Pacotes DHCP malformados.

Consultas de exemplo:

```
# Taxa de distribuição de pool (endereçamento NSWO)
rate(dhcp_discover_pool_allocated_count[5m])

# DISCOVERs que não puderam ser atendidos
increase(dhcp_discover_no_allocation_count[15m])
```

> Nota: o servidor DHCP também emite `dhcp_lease_reap` quando ele reaproveita um lease expirado. Este evento não está declarado como uma métrica do Prometheus hoje e não aparece no endpoint `/metrics`.

Métricas DHCPv6 e RA IPv6

Métrica	Tipo	Descrição
`dhcpv6_info_request_received_count`	Contador	Pacotes DHCPv6 Information-Request recebidos.
`dhcpv6_solicit_received_count`	Contador	Pacotes DHCPv6 Solicit recebidos.
`dhcpv6_request_received_count`	Contador	Pacotes DHCPv6 Request recebidos.
`dhcpv6_info_reply_sent_count`	Contador	Pacotes DHCPv6 Information-Reply enviados.
`dhcpv6_advertise_sent_count`	Contador	Pacotes DHCPv6 Advertise enviados.
`dhcpv6_reply_sent_count`	Contador	Pacotes DHCPv6 Reply enviados.
`ipv6_ra_prefix_add_count`	Contador	Adições de prefixo RA IPv6.
`ipv6_ra_prefix_remove_count`	Contador	Remoções de prefixo RA IPv6.
`ipv6_ra_advertisement_sent_count`	Contador	Anúncios de Roteador IPv6 enviados.

Métricas WLCP

Esses contadores rastreiam o Protocolo de Controle WLAN usado para modo de múltiplas conexões.

Métrica	Tipo	Unidade	Descrição
`wlcp_packet_received_count`	Contador	nenhum	Pacotes WLCP recebidos.
`wlcp_packet_received_bytes`	Soma	bytes	Total de bytes de pacotes WLCP recebidos.
`wlcp_packet_malformed_count`	Contador	nenhum	Pacotes WLCP malformados
`wlcp_packet_unregistered_count`	Contador	nenhum	Pacotes WLCP de clientes não registrados.
`wlcp_pdn_request_count`	Contador	nenhum	Solicitações de conexão PDN.
`wlcp_pdn_accept_count`	Contador	nenhum	Aceites de conexão PDN.
`wlcp_pdn_reject_count`	Contador	nenhum	Rejeições de conexão PDN.
`wlcp_pdn_disconnect_request_count`	Contador	nenhum	Solicitações de desconexão PDN.

Métrica	Tipo	Unidade	Descrição
`wlcp_pdn_disconnect_accept_count`	Contador	nenhum	Aceites de desconexão PDN.

Métricas da VM Erlang

Essas métricas fornecem visibilidade sobre o tempo de execução BEAM.

Memória

Métrica	Descrição
`vm_memory_total`	Memória total alocada (bytes).
`vm_memory_processes`	Memória usada por processos Erlang.
`vm_memory_processes_used`	Memória ativamente usada por processos.
`vm_memory_system`	Memória usada pelo tempo de execução.
`vm_memory_atom`	Memória usada por átomos.
`vm_memory_atom_used`	Memória ativamente usada por átomos.
`vm_memory_binary`	Memória usada por binários.
`vm_memory_code`	Memória usada por código carregado.
`vm_memory_ets`	Memória usada por tabelas ETS (estado de sessão e AP).

Sistema

Métrica	Descrição
`vm_system_info_process_count`	Número atual de processos Erlang.
`vm_system_info_port_count`	Número atual de portas.
`vm_system_info_atom_count`	Número atual de átomos.
`vm_system_info_schedulers`	Total de threads de agendamento.
`vm_system_info_schedulers_online`	Threads de agendamento online.

Agendadores

Métrica	Descrição
`vm_statistics_run_queue`	Comprimento total da fila de execução.
`vm_total_run_queue_lengths_total`	Total da fila de execução do agendador.
`vm_total_run_queue_lengths_cpu`	Fila de execução do agendador de CPU.
`vm_total_run_queue_lengths_io`	Fila de execução do agendador de IO.

Consultas de exemplo:

```
# Memória total em MB
vm_memory_total / 1024 / 1024

# Saturação da fila de execução (deve ser baixa)
avg_over_time(vm_statistics_run_queue[5m])
```

Consultas de Exemplo Grafana

Visão Geral da Autenticação

```
# Aceites por segundo
sum(rate(radius_access_accept_count[5m]))

# Rejeições por segundo
sum(rate(radius_access_reject_count[5m]))

# Percentual da razão de sucesso
100 * sum(rate(radius_access_accept_count[5m])) /
  (sum(rate(radius_access_accept_count[5m])) +
sum(rate(radius_access_reject_count[5m])))
```

Estatísticas por AP

```
# Requisições por AP (Access-Request não tem rótulo nas_ip; use
contabilidade em vez disso)
sum by (nas_ip) (rate(radius_accounting_request_count[5m]))

# Clientes por AP
radius_clients_per_ap_count
```

Análise de Rejeição e Limite de Taxa

```
# Rejeições de autenticação por classe de razão
sum(increase(radius_access_reject_count[1h]))
sum(increase(radius_auth_unauthorized_source_count[1h]))
sum(increase(radius_auth_bad_secret_count[1h]))

# Razões de desmontagem
sum by (reason) (increase(twag_session_teardown_count[1h]))
```

Uso de Dados

```
# Throughput total em MB
(radius_data_bytes_in + radius_data_bytes_out) / 1024 / 1024
```

Regras de Alerta

Alertas Críticos

```
groups:
  - name: twag_critical
    rules:
      - alert: HighAuthenticationFailureRate
        expr: >
          sum(rate(radius_access_reject_count[5m])) /
          (sum(rate(radius_access_accept_count[5m])) +
sum(rate(radius_access_reject_count[5m]))) > 0.1
        for: 5m
        labels:
          severity: critical
        annotations:
          summary: "Alta taxa de falha de autenticação (> 10%)"

      - alert: EAPSyncFailures
        expr: increase(eap_aka_sync_failure_count[5m]) > 0
        for: 1m
        labels:
          severity: warning
        annotations:
          summary: "Falhas de sincronização EAP-AKA detectadas"

      - alert: UnknownRadiusSources
        expr: increase(radius_auth_unauthorized_source_count[15m])
> 0
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Pacotes RADIUS de fontes sem credenciais
correspondentes"

      - alert: BadRadiusSecret
        expr: increase(radius_auth_bad_secret_count[15m]) > 0
        for: 5m
        labels:
          severity: warning
        annotations:
```

```
          summary: "Pacotes RADIUS rejeitados por um segredo
compartilhado incorreto"

      - alert: NoActiveClients
        expr: radius_active_clients_count == 0
        for: 15m
        labels:
          severity: warning
        annotations:
          summary: "Sem clientes ativos por 15 minutos"
```

Alertas de Recursos

```
groups:
  - name: twag_resources
    rules:
      - alert: HighMemoryUsage
        expr: vm_memory_total > 1073741824
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: "Uso de memória TWAG excede 1GB"

      - alert: HighRunQueue
        expr: avg_over_time(vm_statistics_run_queue[5m]) > 10
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Alta fila de execução do agendador indica
saturação da CPU"
```

Páginas Relacionadas

- Arquitetura
- Rejeições RADIUS
- Segredos Compartilhados RADIUS

- Sessões
- Cobrança Online
- Contabilidade por AP

Visão Geral

Cobrança Online (Gy)

Visão Geral

OmniTWAG pode cobrar sessões de dados em tempo real contra um Sistema de Cobrança Online (OCS). OmniTWAG utiliza a interface Diameter Ro (Gy) para enviar Solicitações de Controle de Crédito (CCR) e receber Respostas de Controle de Crédito (CCA). Isso permite que um operador aplique controle de crédito pré-pago ao tráfego WLAN que o TWAG atende.

A cobrança online se aplica a sessões onde o TWAG possui o caminho de dados, ou seja, **NSWO (breakout local)** e tráfego ancorado no TWAG. Nesse caso, o TWAG é o ponto onde o tráfego sai para a internet, atuando como o ponto de aplicação de cobrança. No modo **home-routed (GTP)**, o PGW realiza a cobrança, então um operador normalmente deixa o modo de cobrança global `:disabled`. Consulte a referência de User Plane para os modos de plano de dados.

Dois controles decidem se o Gy é executado para um determinado UE:

- O **modo de cobrança global** (`online_charging.mode`). Isso determina quais Solicitações de Controle de Crédito OmniTWAG pode enviar.
- O **modo de cobrança por AP** na credencial do AP (`charging_mode`). Isso controla, por ponto de acesso, se o OmniTWAG abre uma sessão Gy para aquele UE. Veja Credenciais RADIUS.

Ambos os controles devem permitir a cobrança para que um CCR-Initial seja enviado. Veja Modo de Cobrança por AP.

Índice

- Conceitos
- Modos de Cobrança
- Modo de Cobrança por AP
- Fluxo de Controle de Crédito
- Cota, Uso e Reautorização

- Aplicação de Cota
- Grupo de Classificação e Unidades de Serviço
- Configuração
- Comportamento Operacional e Tratamento de Falhas
- Observabilidade
- Tabelas de Referência

Conceitos

A interface Gy é uma aplicação Diameter que transporta mensagens de controle de crédito entre um elemento de rede e um OCS. OmniTWAG utiliza o dicionário 3GPP Ro com Auth-Application-Id `4`, conforme definido em RFC 4006 e 3GPP TS 32.299. OmniTWAG utiliza o Service-Context-Id `32251@3gpp.org`, que é o contexto de cobrança PS da 3GPP.

OmniTWAG envia três tipos de Solicitação de Controle de Crédito durante uma sessão:

- **CCR-Initial (CCR-I)**: no início da sessão. OmniTWAG solicita ao OCS para iniciar o controle de crédito para o assinante.
- **CCR-Update (CCR-U)**: durante a sessão. OmniTWAG relata dados utilizados e solicita mais cota.
- **CCR-Termination (CCR-T)**: no final da sessão. OmniTWAG relata o uso final de dados e fecha a sessão de controle de crédito.

O OCS responde a cada solicitação com uma Resposta de Controle de Crédito (CCA). Uma CCA transporta um `Result-Code`, uma concessão opcional de cota (`Granted-Service-Unit`), e um `Validity-Time` opcional.

OmniTWAG mantém um estado Gy separado para cada sessão de assinante. O estado Gy contém o `Session-Id` Diameter, o `CC-Request-Number` atual, os contadores de uso cumulativo, e a última cota que o OCS concedeu. OmniTWAG cria esse estado quando envia o CCR-I e descarta o estado quando a sessão termina.

Modos de Cobrança

O parâmetro `online_charging.mode` é o modo de cobrança **global**. Ele seleciona um dos três modos e controla quais Solicitações de Controle de Crédito OmniTWAG pode enviar. O `charging_mode` por AP (veja Modo de Cobrança por AP) então decide, para cada UE, se o OmniTWAG abre uma sessão Gy.

Modo	CCR-I no início	CCR-U durante a sessão	CCR-T no final
`:disabled`	Não	Não	Não
`:authorize_only`	Sim	Não	Sim
`:online_charging`	Sim	Sim	Sim

`:disabled`

OmniTWAG não contata o OCS. Este é o padrão. Use este modo para operação Home-Routed (GTP), onde o PGW realiza a cobrança.

`:authorize_only`

OmniTWAG envia um CCR-I no início da sessão e um CCR-T no final da sessão. OmniTWAG não envia mensagens CCR-U intermediárias. Use este modo para verificar o saldo do assinante no início e relatar o uso total no final, sem gerenciamento de cota ao vivo.

`:online_charging`

OmniTWAG envia um CCR-I no início, um CCR-U em cada atualização intermediária de contabilidade, e um CCR-T no final. Use este modo para gerenciamento completo de cota, onde o OCS concede e renova a cota durante a sessão.

Modo de Cobrança por AP

Cada credencial de AP contém um campo `charging_mode`. Isso é separado do `online_charging.mode` global. OmniTWAG resolve a credencial da fonte RADIUS da sessão, então cada ponto de acesso pode ter seu próprio comportamento. Veja Credenciais RADIUS.

`charging_mode` por AP	Significado	Efeito no Gy
`:charged`	Cobra o tráfego deste AP. Este é o padrão.	OmniTWAG abre uma sessão Gy (CCR-Initial) para o UE quando o modo global também permite.
`:uncharged`	Atende este AP, mas não cobra.	OmniTWAG estabelece a sessão de dados, mas não abre nenhuma sessão Gy. Nenhum CCR é enviado.
`:auth_only`	Apenas autenticar, sem sessão de dados.	OmniTWAG autentica o assinante e para. Não constrói nenhuma sessão de dados e não abre nenhuma sessão Gy.

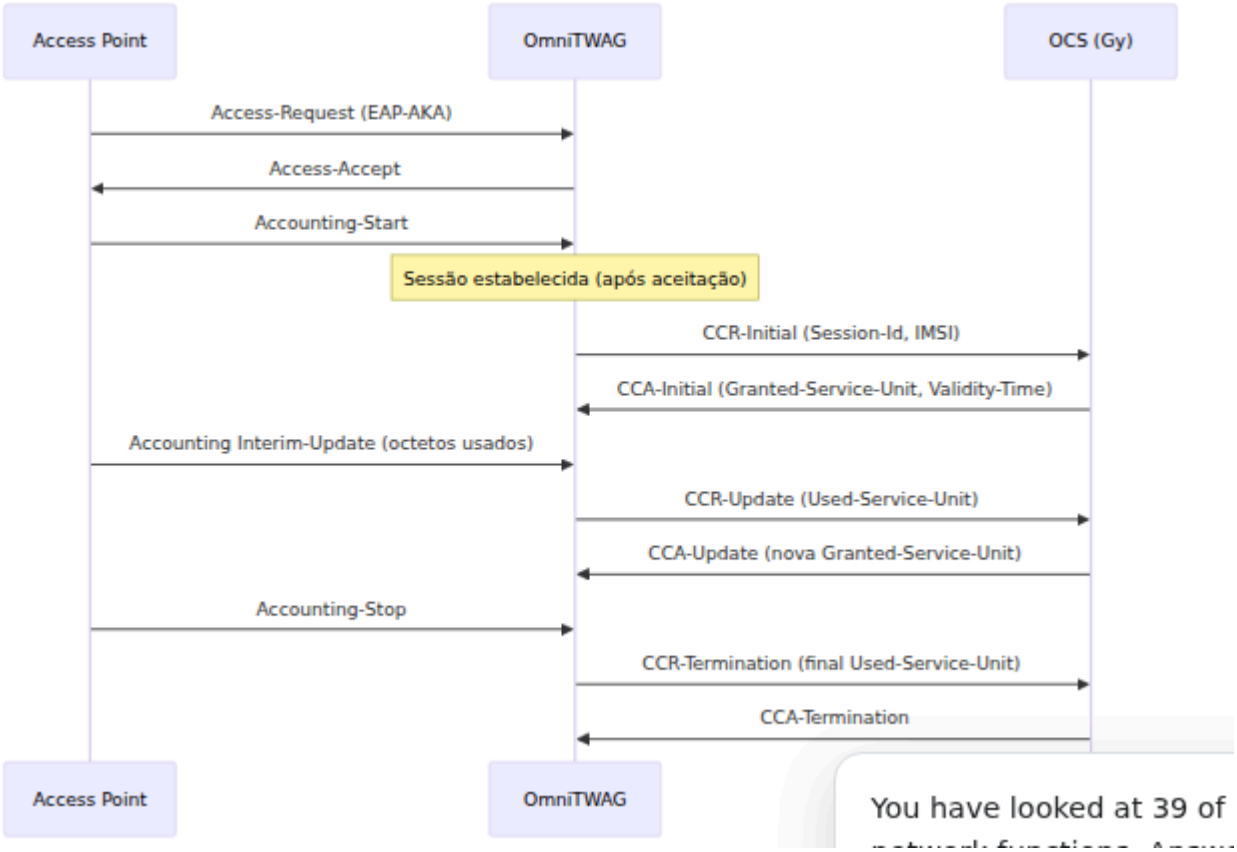
OmniTWAG envia um CCR-Initial somente quando **todas** essas condições são atendidas:

- O `online_charging.mode` global não é `:disabled` (o cliente Gy está habilitado).
- A sessão tem um IMSI.
- O `charging_mode` da credencial do AP é `:charged`.

Se qualquer condição falhar, OmniTWAG não abre nenhuma sessão Gy para o UE, e nenhum CCR-U ou CCR-T posterior é enviado. O modo por AP, portanto, controla o Gy independentemente do modo global: mesmo com o modo global definido como `:online_charging`, um AP marcado como `:uncharged` não gera tráfego Gy.

Fluxo de Controle de Crédito

A sessão de controle de crédito segue a sessão do assinante. OmniTWAG aciona cada Solicitação de Controle de Crédito a partir de um ponto específico no ciclo de vida da sessão RADIUS.



Início da sessão (CCR-Initial)

OmniTWAG envia o CCR-I após estabelecer a sessão. O estabelecimento da sessão ocorre na etapa pós-aceitação, que o servidor RADIUS aciona após o Access-Accept e o Accounting-Start. OmniTWAG envia o CCR-I somente quando o modo de cobrança global está habilitado, a sessão tem um IMSI, e o `charging_mode` da credencial do AP é `:charged`. Veja Modo de Cobrança por AP.

O CCR-I transporta o IMSI do assinante em um AVP `Subscription-ID` com tipo `1` (END_USER_IMSI). O CCR-I define `Multiple-Services-Indicator` como `1` e inclui um grupo `Multiple-Services-Credit-Control` (MSCC) vazio com um `Requested-Service-Unit` vazio e o `Rating-Group` configurado.

Atualizações intermediárias (CCR-Update)

No modo `:online_charging`, cada atualização intermediária de contabilidade RADIUS incorpora os octetos relatados e o tempo da sessão nos contadores de uso do Gy. OmniTWAG não envia um CCR-U em cada intermediário. Ele envia um CCR-U somente quando um desses gatilhos é acionado:

- O volume utilizado ultrapassou a fração configurada do volume concedido (o `quota.update_threshold`, padrão `0.8`).
- O `Validity-Time` da cota concedida expirou.

Um intermediário que não aciona um gatilho apenas registra o uso localmente, então o OCS não é consultado em cada intermediário. Quando um CCR-U é enviado, OmniTWAG lê os octetos usados e o tempo da sessão dos atributos de contabilidade e os relata em um AVP `Used-Service-Unit`. O CCR-U também inclui um `Requested-Service-Unit` vazio, para que o OCS possa conceder mais cota.

Um CCR-U também é enviado fora do caminho de contabilidade quando o temporizador de validade da cota é acionado. Veja Aplicação de Cota.

OmniTWAG lê esses atributos de contabilidade RADIUS para o CCR-U:

Atributo RADIUS	Campo Gy
`Acct-Input-Octets`	`CC-Input-Octets`
`Acct-Output-Octets`	`CC-Output-Octets`
`Acct-Session-Time`	`CC-Time`

OmniTWAG calcula `CC-Total-Octets` como a soma dos octetos de entrada e saída.

Fim da sessão (CCR-Termination)

OmniTWAG envia um CCR-T quando a sessão é encerrada, por exemplo, em um RADIUS Accounting-Stop. OmniTWAG relata o uso cumulativo final em um AVP

`Used-Service-Unit` e define o `Termination-Cause` como `1` (DIAMETER_LOGOUT). OmniTWAG envia o CCR-T tanto no modo `:authorize_only` quanto no modo `:online_charging`.

Cota, Uso e Reautorização

Cota concedida

Cada CCA pode transportar uma `Granted-Service-Unit` dentro do primeiro grupo MSCC. OmniTWAG armazena os valores concedidos no estado Gy:

- `CC-Total-Octets`, `CC-Input-Octets`, `CC-Output-Octets`: o volume de dados concedido.
- `CC-Time`: o tempo concedido.
- `Validity-Time`: o tempo em segundos que a concessão permanece válida. OmniTWAG transforma isso em um prazo absoluto e ativa um temporizador contra ele.

OmniTWAG também lê o `Final-Unit-Indication` do MSCC e o `Result-Code` a nível de MSCC de cada CCA. Uma `Final-Unit-Action` de TERMINATE, REDIRECT ou RESTRICT_ACCESS, ou um `Result-Code` de MSCC de `4010` (END_USER_SERVICE_DENIED) ou `4012` (CREDIT_LIMIT_REACHED), é armazenado como um sinal de exaustão de crédito. Veja Aplicação de Cota.

OmniTWAG atualiza a concessão armazenada em cada CCA-I e CCA-U bem-sucedidos.

Relato de uso

OmniTWAG mantém contadores de uso cumulativo no estado Gy. Os contadores de contabilidade RADIUS são cumulativos para a sessão, então cada atualização intermediária relata o total até o momento. OmniTWAG registra os contadores relatados e os envia como o `Used-Service-Unit`. O CCR-U define o `Reporting-Reason` como `4` (VALIDITY_TIME). O CCR-T define o `Reporting-Reason` como `2` (FINAL).

Reautorização

OmniTWAG reautentica em dois caminhos:

- **No tráfego.** Uma atualização intermediária de contabilidade RADIUS que ultrapassa o limite da cota ou encontra a validade expirada aciona um CCR-U. O intervalo intermediário é uma propriedade do ponto de acesso, então um intervalo intermediário mais curto fornece medição mais precisa.
- **No tempo.** OmniTWAG ativa um temporizador de validade da cota a partir do `Validity-Time` concedido. Se o temporizador expirar com pouco ou nenhum tráfego, OmniTWAG ainda envia um CCR-U para reautenticar. Se o OCS não enviar `Validity-Time`, OmniTWAG pode recorrer ao `quota.validity_time` configurado. Com nenhum dos dois presentes, nenhum temporizador de validade é ativado.

O `CC-Request-Number` aumenta em um após cada CCA bem-sucedido. Esse valor dá a cada solicitação na sessão de controle de crédito um número de sequência único, conforme exigido pela RFC 4006.

Aplicação de Cota

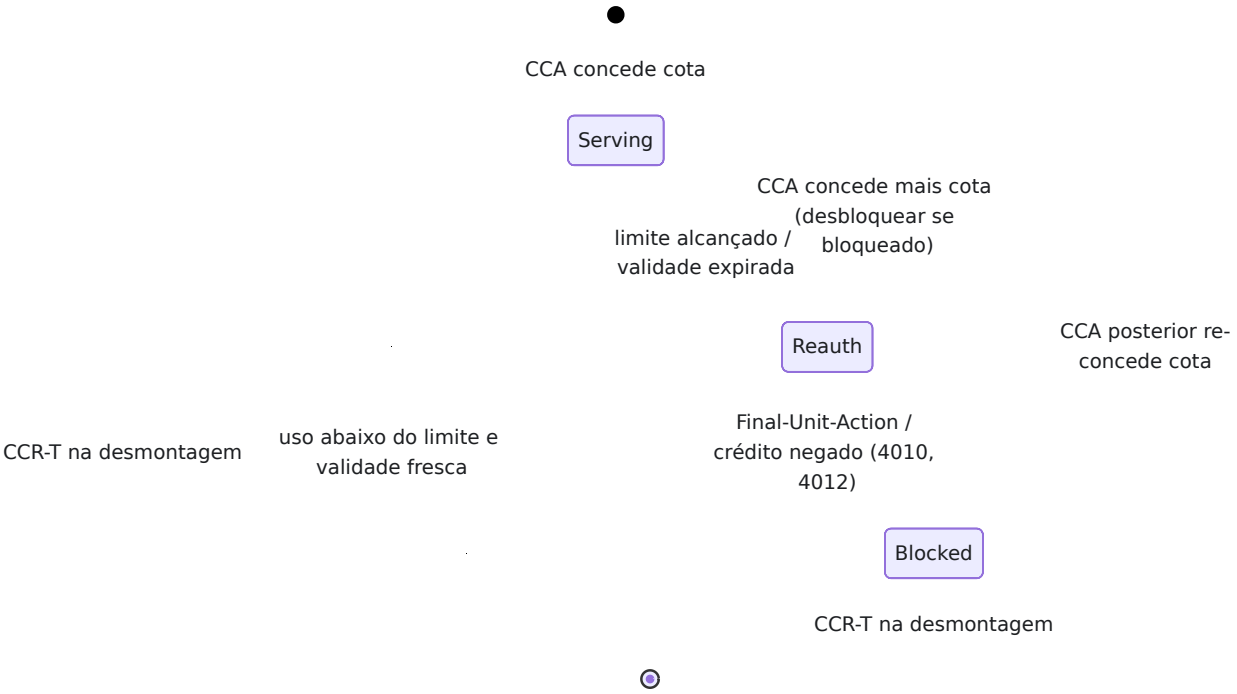
OmniTWAG aplica a exaustão de crédito no modo `:online_charging`. O TWAG não possui uma função de plano de usuário separada, então seu mecanismo de aplicação é o caminho de dados GRE: ele desregistra o UE do caminho de dados, o que interrompe o uplink e o downlink para aquele UE.

Uma resposta CCR-U aciona a aplicação quando o estado Gy mostra crédito esgotado, ou seja, quando o OCS:

- sinaliza qualquer `Final-Unit-Action` (TERMINATE, REDIRECT ou RESTRICT_ACCESS), ou
- retorna um `Result-Code` de MSCC de `4010` ou `4012`.

Como o TWAG não pode honrar REDIRECT ou RESTRICT_ACCESS em seu caminho de dados, qualquer `Final-Unit-Action` é tratada como uma parada. Quando a aplicação é acionada, OmniTWAG bloqueia o caminho de dados do UE, mas mantém a sessão, o aluguel de IP e a contabilidade ativos. Uma

reautorização posterior que concede cota utilizável restaura o caminho de dados.



A aplicação é de melhor esforço e falha aberta em erros de transporte: um CCR-U que falha na camada Diameter nunca bloqueia o UE. O bloqueio ocorre apenas em uma negação explícita de crédito do OCS ou sinal de exaustão. Veja Comportamento Operacional e Tratamento de Falhas.

Grupo de Classificação e Unidades de Serviço

OmniTWAG usa um único grupo de classificação para toda a sessão. O parâmetro `rating_group` define o valor de `Rating-Group` no MSCC do CCR-I. O padrão é `1`. OmniTWAG não divide o tráfego em múltiplos grupos de classificação ou identificadores de serviço. OmniTWAG solicita uma concessão de volume combinada com um `Requested-Service-Unit` vazio, para que o OCS decida a concessão.

Configuração

Defina os parâmetros de cobrança online sob a aplicação `:omnitwag`, no bloco `online_charging` de `config/runtime.exs`.

```
config :omnitwag,
  online_charging: %{
    # Modo de cobrança global: :disabled, :authorize_only, ou
:online_charging
    mode: :disabled,

    # Reino Diameter do OCS (Destination-Realm para CCR)
    ocs_realm: "epc.mnc001.mcc001.3gppnetwork.org",

    # Grupo de classificação para o grupo Multiple-Services-
Credit-Control
    rating_group: 1,

    # Gerenciamento de cota (usado no modo :online_charging)
    quota: %{
      # Fração do volume concedido em que um CCR-Update é acionado
      update_threshold: 0.8,

      # Tempo de validade de fallback (segundos) quando o OCS não
envia Validity-Time
      validity_time: 3600
    }
  }
```

Parâmetros

Parâmetro	Tipo	Requerido	Padrão	Descr
`mode`	Atom	Não	`:disabled`	Modo de co global. Um `:disabled` `:authoriz` ou `:online_c` Veja Modos Cobrança.
`ocs_realm`	String	Não	Reino Diameter	Reino Diam OCS. Omni usa isso co `Destinati` `Realm` em CCR. Se nã definido, OmniTWAG reino Diam local.
`rating_group`	Integer	Não	`1`	Valor de `Ra` `Group` que OmniTWAG no MSCC d Initial.
`quota.update_threshold`	Float	Não	`0.8`	Fração (0.0 do volume concedido uma atuali intermediá aciona um Update.

Parâmetro	Tipo	Requerido	Padrão	Descr
`quota.validity_time`	Integer	Não	(não definido)	Validade d de fallback segundos, para ativar temporizad validade qu OCS não en `Validity-` não definic nenhum temporizad fallback é a

O `charging_mode` por AP não faz parte deste bloco. Ele reside em cada credencial de AP. Veja Credenciais RADIUS.

O `Origin-Host` e o `Origin-Realm` Diameter para a sessão Gy vêm da identidade Diameter na configuração `:diameter_ex` (`host` e `realm`). O `Destination-Host` não é configurado. OmniTWAG aprende o `Destination-Host` a partir do `Origin-Host` do primeiro CCA, e adiciona este AVP a mensagens CCR-U e CCR-T posteriores.

Transporte Diameter e a aplicação Ro

O cliente Gy envia através da pilha Diameter compartilhada. Configure o OCS como um par Diameter e habilite a aplicação `:ro`, na configuração `:diameter_ex`.

```
config :diameter_ex,
  diameter: %{
    service_name: :omnitouch_twag,
    host: "amanaki",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    request_timeout: 5000,
    applications: [
      %{
        application_name: :ro,
        application_dictionary: :diameter_gen_3gpp_ro,
        vendor_specific_application_ids: [
          %{vendor_id: 0, auth_application_id: 4,
acct_application_id: nil}
        ]
      }
    ],
    peers: [
      %{
        host: "ocs.epc.mnc001.mcc001.3gppnetwork.org",
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        ip: "10.179.2.233",
        port: 3868,
        transport: :diameter_tcp,
        initiate_connection: true
      }
    ]
  }
```

Parâmetro	Tipo	Requerido	Padrão	
`service_name`	Atom	Sim	-	Nor Dia Om soli des
`host`	String	Sim	-	Ider loca con Gy `rea`
`realm`	String	Sim	-	Rei Om com e co `Rea` `ocs` defi
`request_timeout`	Integer	Não	5000	Tem mili esp ant falh
`application_name`	Atom	Sim	-	Nor Dia `:ro`
`application_dictionary`	Atom	Sim	-	Dici par com `:di` par

Parâmetro	Tipo	Requerido	Padrão	
`vendor_specific_application_ids`	Lista	Sim	-	ID d esp forr `ven` `aut` `4`.
`peers`	Lista	Sim	-	List Adi um Par

Parâmetros do Par

Parâmetro	Tipo	Requerido	Padrão	Desc
`host`	String	Sim	-	Identida Diamet par (FQ Deve corresp identida OCS.
`realm`	String	Sim	-	Reino Diamet par. Def como o do OCS
`ip`	String	Sim	-	Endere OCS pa conexã
`port`	Integer	Não	3868	Porta D do OCS porta p 3868.
`transport`	Atom	Não	`:diameter_tcp`	Protoco transpo `:diame` para TC
`initiate_connection`	Booleano	Não	false	Quando OmniTW inicia a conexã par.

> **Nota:** O bloco `online_charging` no `config/runtime.exs` enviado também mostra `ocs_host`, `service_identifier`, um `request_timeout` de nível superior, e as chaves `initial_request` e `subsequent_request` da `quota`. O código de cobrança atual não lê essas chaves. Ele lê `mode`, `ocs_realm`, `rating_group`, `quota.update_threshold`, e `quota.validity_time`. O tempo limite da solicitação Diameter vem do `request_timeout` do `:diameter_ex`. OmniTWAG solicita cota com um `Requested-Service-Unit` vazio, então `initial_request` e `subsequent_request` não moldam o CCR atualmente. Não dependa das chaves não lidas até que o código as suporte.

Comportamento Operacional e Tratamento de Falhas

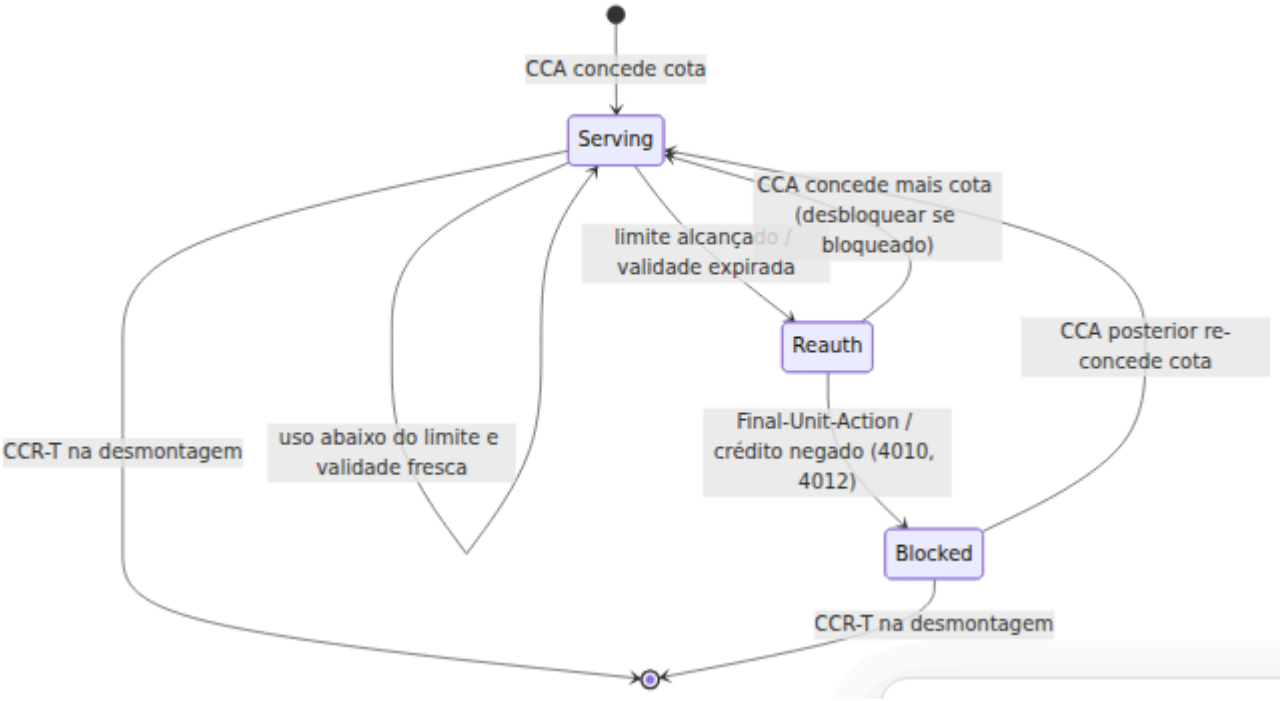
Comportamento de falha aberta

OmniTWAG trata uma falha de transporte Gy como não fatal. Se uma Solicitação de Controle de Crédito falhar na camada Diameter, a sessão do assinante continua e OmniTWAG não bloqueia o tráfego.

- **OCS inacessível ou tempo limite**: O CCR falha no transporte Diameter. OmniTWAG registra um aviso. Para um CCR-I, OmniTWAG ainda mantém o estado Gy, para que possa enviar um CCR-T mais tarde. Para um CCR-U, OmniTWAG mantém o estado Gy anterior e não bloqueia o UE. Para um CCR-T, OmniTWAG registra o aviso e completa a desmontagem.
- **Código de resultado de erro CCA**: O OCS retorna um `Result-Code` a nível de comando que não é `2001` (DIAMETER_SUCCESS). OmniTWAG registra o erro e trata a solicitação como falhada. OmniTWAG mantém a sessão ativa.

A aplicação é separada da falha de transporte. No modo `:online_charging`, OmniTWAG bloqueia o caminho de dados do UE quando o OCS nega explicitamente crédito ou sinaliza uma Final-Unit-Action em um CCA MSCC. Veja Aplicação de Cota. Ele nunca bloqueia em uma falha de transporte, e nunca nega acesso no modo `:authorize_only`. Planeje a integração do OCS com essa divisão em mente.

Resumo do ciclo de vida da sessão de controle de crédito



Tratamento de código de resultado

OmniTWAG lê o código de resultado em dois níveis.

- **`Result-Code` a nível de comando.** OmniTWAG trata `2001` como sucesso. Ele também trata um `Result-Code` ausente como sucesso, porque algumas respostas de pares chegam como um registro bruto sem um código de resultado decodificado. Qualquer outro valor a nível de comando é um erro: OmniTWAG o registra, não tenta novamente, e mantém a sessão ativa.
- **`Result-Code` a nível de MSCC.** Dentro do primeiro grupo MSCC, um `Result-Code` de `4010` (END_USER_SERVICE_DENIED) ou `4012` (CREDIT_LIMIT_REACHED) é lido como um sinal de negação de crédito. No modo `:online_charging`, isso aciona a aplicação e bloqueia o caminho de dados do UE. Veja Aplicação de Cota.

Observabilidade

OmniTWAG emite eventos de Telemetria para cada Solicitação e resposta de Controle de Crédito. Os eventos transportam um valor de metadados `type` de `:initial`, `:update`, ou `:terminate`.

Evento de Telemetria	Quando	Metadados
`[:twag, :gy, :ccr, :sent]`	OmniTWAG envia um CCR	`type`
`[:twag, :gy, :cca, :received]`	OmniTWAG recebe uma CCA	`type`, `result` (`:success` ou `:error`)
`[:twag, :gy, :ccr, :error]`	Um CCR falha no transporte	`type`
`[:twag, :gy, :quota, :blocked]`	O caminho de dados do UE foi bloqueado por exaustão de crédito	`identity`
`[:twag, :gy, :quota, :unblocked]`	O caminho de dados do UE foi restaurado após uma re-concessão	`identity`

> **Nota:** Esses eventos de Telemetria Gy não estão registrados como métricas Prometheus na versão atual. A Referência de Métricas ainda não os exporta. Para observar o comportamento do Gy hoje, use os logs da aplicação. O cliente Gy registra cada CCR-I, CCR-U e CCR-T no nível INFO, e cada falha no nível WARN ou ERROR. As linhas de log transportam o `Session-Id` Diameter e, para o CCR-I, o IMSI.

Tabelas de Referência

Valores de CC-Request-Type

Valor	Nome	Mensagem
1	INITIAL_REQUEST	CCR-Initial
2	UPDATE_REQUEST	CCR-Update
3	TERMINATION_REQUEST	CCR-Termination

Os valores são definidos em RFC 4006 §8.3.

Valores de Reporting-Reason usados

Valor	Nome	Onde OmniTWAG o usa
2	FINAL	CCR-Termination
4	VALIDITY_TIME	CCR-Update

Código de Resultado

Código	Nível	Nome	Comportamento do OmniTWAG
2001	Comando	DIAMETER_SUCCESS	Tratado como sucesso. O estado Gy avança.
(ausente)	Comando	-	Tratado como sucesso, para respostas de registro bruto.
Qualquer outro	Comando	(erro)	Registrado como um erro. A sessão continua.
4010	MSCC	END_USER_SERVICE_DENIED	Lido como crédito negado. Aplica (bloqueia o UE) no modo `:online_charging`.
4012	MSCC	CREDIT_LIMIT_REACHED	Lido como crédito negado. Aplica (bloqueia o UE) no modo `:online_charging`.

Valores de Final-Unit-Action

O MSCC `Final-Unit-Indication` transporta uma `Final-Unit-Action`. OmniTWAG não possui função de plano de usuário, então não pode honrar REDIRECT ou RESTRICT_ACCESS no caminho de dados. Ele trata qualquer Final-Unit-Action como uma parada e bloqueia o UE.

Valor	Nome	Comportamento do OmniTWAG
0	TERMINATE	Bloqueia o caminho de dados do UE (sessão retida).
1	REDIRECT	Bloqueia o caminho de dados do UE (redirecionamento não suportado).
2	RESTRICT_ACCESS	Bloqueia o caminho de dados do UE (restrição não suportada).

Final-Unit-Indication é definido em 3GPP TS 32.299 e RFC 4006 §5.6.

Aplicação Diameter

ID	Interface	Dicionário	Direção	Referência
4	Ro / Gy	`diameter_gen_3gpp_ro`	TWAG para OCS	RFC 4006, 3GPP TS 32.299

Referências de Especificação

Tópico	Especificação
Aplicação de Controle de Crédito Diameter	RFC 4006
Aplicações de cobrança Diameter (Ro)	3GPP TS 32.299
Arquitetura de cobrança	3GPP TS 32.240
TWAG / acesso não 3GPP (S2a, NSWO)	3GPP TS 23.402
Protocolo Base Diameter	RFC 6733

Páginas Relacionadas

- Visão Geral
- Credenciais RADIUS
- Pools de IP
- Gerenciamento de Endereço IP
- Sessões
- S2a GTP-C
- User Plane

Visão Geral

Contabilidade por AP

Visão Geral

OmniTWAG agrega dados de contabilidade RADIUS para cada ponto de acesso. O operador vê o tráfego total e a contagem de sessões para cada ponto de acesso. Esses dados são separados dos registros de contabilidade por sessão. OmniTWAG chaveia os totais por AP pelo `NAS-IP-Address` do ponto de acesso. Quando um ponto de acesso omite `NAS-IP-Address` (por exemplo, alguns modelos da UBNT), OmniTWAG recorre ao IP de origem UDP do pacote RADIUS, de modo que a contabilidade se anexe à mesma entrada do ponto de acesso que a autenticação criou.

Índice

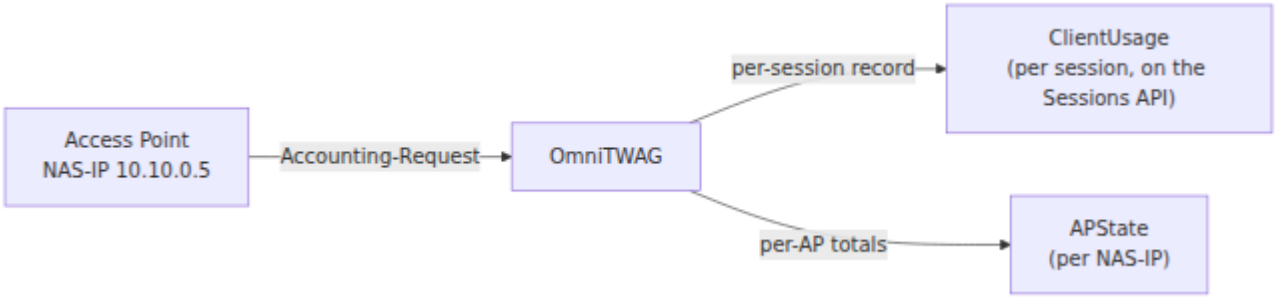
- Conceitos
- Como os totais são contados
- Campos registrados
- Contagem de clientes ativos
- Onde ver os dados
- Relação com dados por sessão
- Solução de Problemas
- Páginas Relacionadas

Conceitos

Um ponto de acesso envia pacotes RADIUS Accounting-Request para cada sessão de cliente. Um pacote tem um `Acct-Status-Type` que indica o evento:

- **Início** (`1`): uma sessão começou.
- **Atualização Interina** (`3`): uma sessão está ativa, com os contadores até agora.
- **Parada** (`2`): uma sessão parou, com os contadores finais.

OmniTWAG lê os contadores de octetos e pacotes e o `NAS-IP-Address` de cada pacote. OmniTWAG adiciona esses valores aos totais acumulados para aquele ponto de acesso. Isso dá ao operador uma visão por AP do tráfego e das sessões.



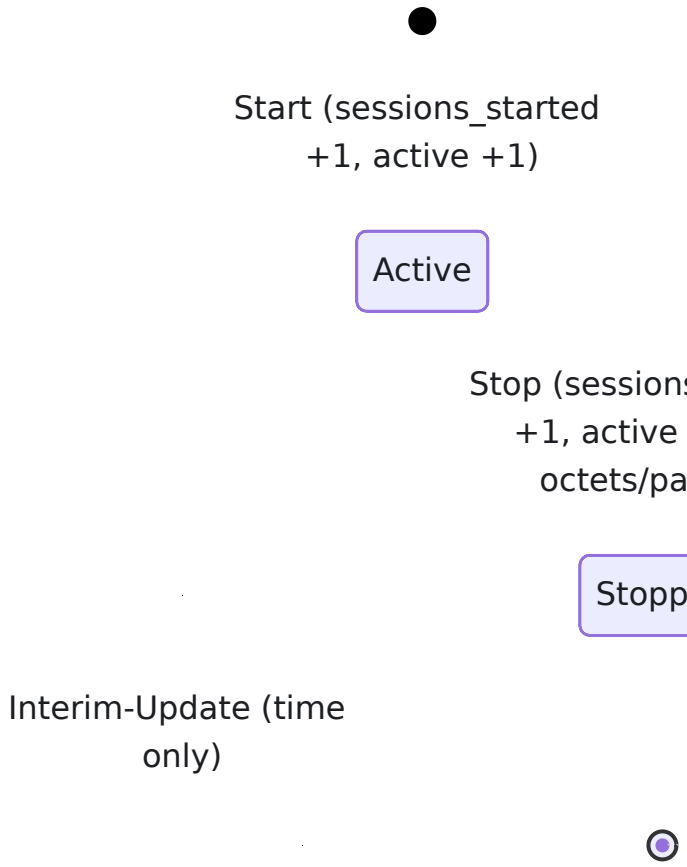
Como totais são contados

Os contadores de octetos e pacotes RADIUS são **cumulativos para cada sessão**. Uma Atualização Interina e uma Parada relatam o total da sessão até agora, não a mudança desde o último pacote. Se OmniTWAG adicionasse os contadores de cada pacote, ele contaria o mesmo tráfego mais de uma vez.

Para evitar contagem dupla, OmniTWAG adiciona os contadores de octetos e pacotes aos totais por AP **apenas em uma Parada**. Uma Parada relata o total final para uma sessão concluída. As regras são:

- Em **Início**: aumentar `acct_sessions_started` em um. Aumentar `acct_active_sessions` em um.
- Em **Parada**: aumentar `acct_sessions_stopped` em um. Diminuir `acct_active_sessions` em um, mas não abaixo de zero. Adicionar os contadores finais de octetos e pacotes aos totais por AP.
- Em **Atualização Interina**: atualizar apenas o tempo da última contabilidade. Não alterar os totais.

Se OmniTWAG receber uma Parada para uma sessão que não viu começar, a contagem de sessões ativas não vai abaixo de zero. OmniTWAG ainda adiciona os octetos e pacotes finais. Se OmniTWAG receber um Accounting-Request para um ponto de acesso que não viu antes, OmniTWAG cria a entrada por AP. Isso significa que um ponto de acesso apenas para contabilidade ainda é rastreado.



Campos registrados

OmniTWAG registra esses campos para cada ponto de acesso, além dos metadados do ponto de acesso, como SSID e `Called-Station-Id`.

Campo	Tipo	Descrição
`acct_input_octets`	Inteiro	Total de octetos de entrada, somados sobre sessões paradas.
`acct_output_octets`	Inteiro	Total de octetos de saída, somados sobre sessões paradas.
`acct_input_packets`	Inteiro	Total de pacotes de entrada, somados sobre sessões paradas.
`acct_output_packets`	Inteiro	Total de pacotes de saída, somados sobre sessões paradas.
`acct_sessions_started`	Inteiro	Número de sessões que começaram neste ponto de acesso.
`acct_sessions_stopped`	Inteiro	Número de sessões que pararam neste ponto de acesso.
`acct_active_sessions`	Inteiro	Medidor de sessões abertas derivado da contabilidade: `started` menos `stopped`, nunca menos que zero. Reflete apenas o que a contabilidade RADIUS relata, podendo contar a mais quando um ponto de acesso perde pacotes de Parada (por exemplo, após uma reinicialização) e permanece desatualizado até que as Paradas cheguem ou a entrada expire. Este **não** é o número de "Clientes Ativos" mostrado na página de Pontos de Acesso — veja Contagem de clientes ativos.
`last_accounting_at`	Inteiro	Hora do último pacote de contabilidade, em milissegundos desde

Campo	Tipo	Descrição
		a época Unix.

A direção dos contadores de octetos e pacotes segue a definição RADIUS em RFC 2866. Entrada é o tr��fego do cliente para a rede. Saída é o tráfego da rede para o cliente.

Contagem de clientes ativos

O número de **Clientes Ativos** na página de Pontos de Acesso, e o campo `client_count` na API do ponto de acesso, relatam quantos clientes estão **conectados agora**. Esse valor não é o contador de contabilidade `acct_active_sessions`. Ele é derivado dos processos de sessão em memória ao vivo que OmniTWAG executa para cada cliente autenticado, agrupados pelo `NAS-IP-Address` do ponto de acesso.

Cada processo de sessão monitora a vivacidade a partir das atualizações interinas de contabilidade. Quando um cliente para de enviar atualizações por um número configurado de intervalos, o processo se encerra, independentemente de um RADIUS Accounting-Stop chegar. Como a contagem de clientes ativos é um total desses processos ao vivo, ela se esvazia para zero por conta própria quando um ponto de acesso cai — não depende do ponto de acesso enviar uma Parada, e não aumenta após uma reinicialização do ponto de acesso.

Use `client_count` (o número ao vivo) como a contagem de clientes atualmente conectados. Use `acct_active_sessions` quando você quiser especificamente o medidor de sessões abertas relatado pela contabilidade, mantendo suas ressalvas acima em mente.

Onde ver os dados

Os campos por AP são parte do registro do ponto de acesso. O operador os lê em dois lugares:

- **REST API**: o endpoint `GET /access_points` retorna todos os pontos de acesso com seus campos de contabilidade por AP. O endpoint `GET /access_points/{id}` retorna um ponto de acesso pelo seu `NAS-IP-Address`.
- **Interface Web**: a página de **Pontos de Acesso** mostra todos os pontos de acesso. Expanda um ponto de acesso para ver seus dados completos, incluindo os campos de contabilidade por AP.

Relação com dados por sessão

Os totais por AP são um agregado. Para cada RADIUS Accounting-Request, OmniTWAG escreve dois registros:

- Os **totais por AP** descritos aqui, mantidos em `APState` e chaveados por NAS-IP.
- Um **registro por sessão**, mantido em `ClientUsage` e chaveado por `Acct-Session-Id`. Esta é a visão ao vivo de tráfego e contabilidade por assinante, e é exposta na API de Sessões, não no registro do ponto de acesso. Veja Sessões.

Os totais por AP contam octetos e pacotes apenas em uma Parada, então eles ficam atrasados em relação à visão ao vivo por sessão durante uma sessão ativa. Use a API de Sessões quando precisar do uso atual de um único assinante, e os totais por AP quando precisar do tráfego vitalício e contagens de sessões para um ponto de acesso.

A entrada do ponto de acesso é criada seja pela autenticação (o AP é registrado a partir do Access-Request, veja RADIUS Shared Secrets) ou, para pontos de acesso apenas para contabilidade, pelo primeiro Accounting-Request. Um ponto de acesso nunca aparece se seus pacotes forem descartados porque o IP de origem não corresponder a nenhuma sub-rede de credenciais.

Solução de Problemas

Os totais não mudam

Sintomas: Os totais de octetos e pacotes para um ponto de acesso permanecem em zero, mesmo que clientes usem a rede.

Causas possíveis:

- O ponto de acesso não envia pacotes de Parada. OmniTWAG adiciona octetos e pacotes apenas em uma Parada.
- O ponto de acesso não envia contabilidade de forma alguma.

Resolução:

1. Confirme que o ponto de acesso envia pacotes Accounting-Request para o OmniTWAG.
2. Confirme que o ponto de acesso envia uma Parada ao final de uma sessão.
3. Confirme que o segredo compartilhado de contabilidade no ponto de acesso corresponde ao credencial em arquivo. Veja RADIUS Shared Secrets.

A contagem de sessões ativas parece muito alta

Sintomas: O valor de `acct_active_sessions` é maior que o número real de clientes ativos.

Causas possíveis:

- O ponto de acesso enviou pacotes de In��cio, mas não enviou os pacotes de Parada correspondentes, por exemplo, após uma reinicialização do ponto de acesso.

Resolução:

1. Confirme que o ponto de acesso envia uma Parada para cada sessão.
2. Defina um intervalo de atualização interina no ponto de acesso, para que o OmniTWAG veja sessões ativas.

3. A contagem ativa se corrige à medida que novos pacotes de Parada chegam. A contagem não vai abaixo de zero.

Isso afeta apenas `acct_active_sessions`. O número de **Clientes Ativos** na página de Pontos de Acesso (e o campo `client_count` da API) não é afetado, porque é um total de processos de sessão ao vivo em vez do contador de contabilidade. Veja Contagem de clientes ativos.

Um ponto de acesso não aparece

Sintomas: Um ponto de acesso que envia contabilidade não aparece na lista.

Causas possíveis:

- OmniTWAG descartou os pacotes porque o IP de origem não correspondeu a nenhuma sub-rede de credenciais.

Resolução:

1. Verifique a métrica `radius_auth_unauthorized_source_count`. Veja Metrics Reference e RADIUS Rejections.
2. Adicione uma credencial para a sub-rede do ponto de acesso. Veja RADIUS Shared Secrets.

Páginas Relacionadas

- Visão Geral
- Sessões
- RADIUS Shared Secrets
- RADIUS Rejections
- Metrics Reference
- Arquitetura

Visão Geral

Benchmark de Performance do OmniTWAG

Taxa de autenticação e sinalização do plano de controle do Trusted WLAN Access Gateway.

Ambiente de teste

CPU	Intel(R) Core(TM) i5-5250U CPU @ 1.60GHz (2 núcleos)
Runtime	Elixir 1.17.0 / Erlang OTP 27 (JIT) (erts 15.0.1)
Medição	1 s de aquecimento + 3 s de medição + 1 s de memória por op
Commit	`main` @ `01ec4e8`

A taxa de transferência é relatada como **ips** (iterações por segundo: quanto maior, mais rápido). **Avg** e **Mediana** são a latência por chamada em microssegundos; a mediana é a melhor estimativa da latência típica em estado estacionário. **Mem/op** é a memória heap alocada por chamada.

Autenticação completa (de ponta a ponta)

A figura principal: cada cenário encadeia a sequência completa de anexação (EAP Identity decode, manipulação do vetor de autenticação, a hierarquia de chaves EAP-AKA', construção de desafio, MAC, decodificação de resposta e verificação de MAC), então sua taxa de transferência (**ips**) é **anexos completos, ou seja, sessões, por segundo por núcleo**. O *modo local*

calcula o vetor de autenticação no dispositivo (Milenage); o *modo STa* o recebe do HSS via SWx/STa.

Cenário	Taxa de transferência (ips)	Avg (µs)	Mediana (µs)	Mem/op
Anexo completo EAP-AKA', modo STa (vetor do HSS)	17,311	57.77	52.90	7,048 B
Anexo completo EAP-AKA', modo local (vetor calculado no dispositivo)	12,493	80.05	73.31	10,560 B

Criptografia

Operações criptográficas por autenticação: geração de vetor de autenticação SIM (Milenage), a hierarquia de chaves EAP-AKA' (CK'/IK', chaves de sessão, MAC), chaves de reautenticação rápida ERP, e a função genérica de derivação de chaves 3GPP.

Operação	Taxa de transferência (ips)	Avg (µs)	Mediana (µs)	Mem/op
Derivação de chave WLCP (EMSK -> chave WLCP)	262,975	3.80	3.54	72 B
EAP-AKA' construir pacote de desafio (codificar)	257,991	3.88	0.87	456 B
EAP-AKA' derivar CK'/IK'	225,191	4.44	3.93	328 B
EAP-AKA' calcular MAC (HMAC-SHA-256/16)	223,503	4.47	4.11	72 B
KDF genérico 3GPP (HMAC-SHA-256)	198,596	5.04	4.32	248 B
ERP derivar rMSK (por reautenticação)	113,269	8.83	8.01	376 B
ERP reautenticação rápida completa (rRK -> rIK -> rMSK)	61,496	16.26	14.97	520 B
Milenage compute_all (RES, CK, IK, AK, MAC-A)	52,451	19.07	16.11	3,560 B
EAP-AKA' derive_keys (MK -> todas as chaves de sessão)	31,696	31.55	28.93	1,552 B

Operação	Taxa de transferência (ips)	Avg (µs)	Mediana (µs)	Mem/op
EAP-AKA' criptografia de autenticação completa (CK'/IK' + todas as chaves)	28,380	35.24	32.59	1,808 B

Operação	O que faz
EAP-AKA' construir pacote de desafio	Codifica o EAP-Request/AKA'-Challenge (AT_RAND, AT_AUTN, AT_KDF, AT_MAC).
Derivação de chave WLCP	Deriva a chave de proteção WLCP do EMSK (modo de múltiplas conexões).
KDF genérico 3GPP	Função de derivação de chave HMAC-SHA-256 usada em toda a hierarquia de chaves.
EAP-AKA' calcular MAC	HMAC-SHA-256 AT_MAC sobre um pacote EAP, truncado para 16 bytes.
EAP-AKA' derivar CK'/IK'	Transforma CK/IK em CK'/IK' vinculado ao nome da rede de acesso.
ERP derivar rMSK	Chave de sessão por reautenticação para EAP Re-autenticação (RFC 6696).
ERP reautenticação rápida completa	Cadeia completa de derivação rRK → rIK → rMSK para uma reautenticação rápida.
Milenage compute_all	Um vetor de autenticação AKA: RES, CK, IK, AK, MAC-A.
EAP-AKA' derive_keys	Expande a chave mestre em K_encr, K_aut, K_re, MSK, EMSK.
EAP-AKA' criptografia de autenticação completa	Derivação de CK'/IK' mais a expansão completa da chave de sessão.

Codecs de pacotes

Operações de codificação/decodificação do plano de sinalização: análise de pacotes EAP, manipulação de mensagens WLCP (TS 24.244) para modo de

múltiplas conexões, e o atributo de chave MPPE construído em cada RADIUS Access-Accept.

Operação	Taxa de transferência (ips)	Avg (µs)	Mediana (µs)	Mem/op
WLCP decodificar (Solicitação de Conexão PDN)	4,229,558	0.24	0.20	448 B
WLCP codificar (Solicitação de Conexão PDN)	2,492,441	0.40	0.24	160 B
WLCP codificar + decodificar ida e volta	1,989,407	0.50	0.40	608 B
EAP decodificar pacote de identidade	336,816	2.97	2.66	1,136 B
MPPE construir VSA (Send-Key, RC4 + MD5)	135,026	7.41	5.32	1,296 B

Operação	O que faz
WLCP decodificar	Analisa uma Solicitação de Conexão PDN WLCP do cabeçalho.
WLCP codificar	Serializa uma Solicitação de Conexão PDN WLCP para o cabeçalho.
WLCP codificar + decodificar ida e volta	Codifica e depois decodifica uma mensagem WLCP.
EAP decodificar pacote de identidade	Analisa uma resposta de identidade EAP (extração de NAI).
MPPE construir VSA	Constrói o MS-MPPE-Send-Key RADIUS VSA (RC4 + MD5) carregando o MSK.

Segredos Compartilhados RADIUS (por grupo de IP de origem)

Visão Geral

OmniTWAG autentica cada cliente RADIUS (um ponto de acesso ou NAS) com um segredo compartilhado. Este recurso permite que um operador gerencie **muitos** segredos compartilhados, cada um vinculado a uma ou mais sub-redes de IP de origem. Um grupo de pontos de acesso que compartilha uma sub-rede também compartilha um segredo. O operador adiciona, altera e remove essas credenciais sob demanda através da API REST ou da interface web do painel de controle. O armazenamento é persistente, então as credenciais sobrevivem a uma reinicialização.

Índice

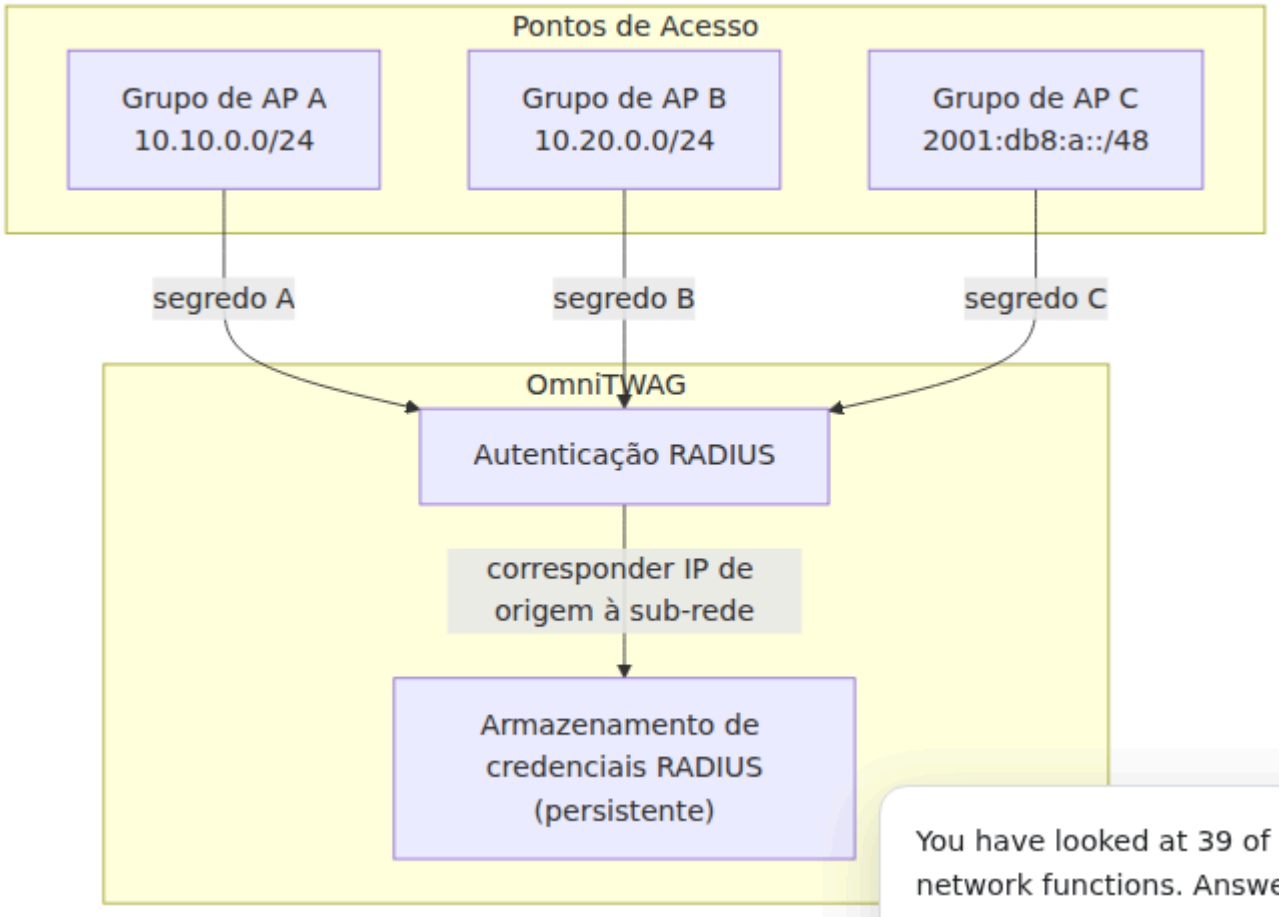
- Conceitos
- Como a autenticação usa uma credencial
- Modelo de credencial
- Quebra e cobrança por AP
- API REST
- Interface web
- Configuração
- Migração de um único segredo estático
- Métricas
- Solução de Problemas

Conceitos

Uma **credencial** vincula um segredo compartilhado a uma ou mais **sub-redes de origem**. Uma sub-rede de origem é um bloco CIDR. A sub-rede pode ser IPv4, IPv6 ou uma mistura de ambos.

O **IP de origem** é o endereço IP do ponto de acesso que envia o pacote RADIUS. OmniTWAG lê o IP de origem do pacote UDP recebido. OmniTWAG não usa o atributo `NAS-IP-Address` para seleção de segredo, porque um atacante pode definir esse atributo para qualquer valor.

Uma sub-rede de credencial também **autoriza** o ponto de acesso. Se o IP de origem corresponder a uma sub-rede de credencial, o ponto de acesso é autorizado. Se o IP de origem não corresponder a nenhuma sub-rede de credencial, OmniTWAG descarta o pacote. Isso significa que quando um operador adiciona uma credencial para um novo ponto de acesso, o ponto de acesso é autorizado ao mesmo tempo.

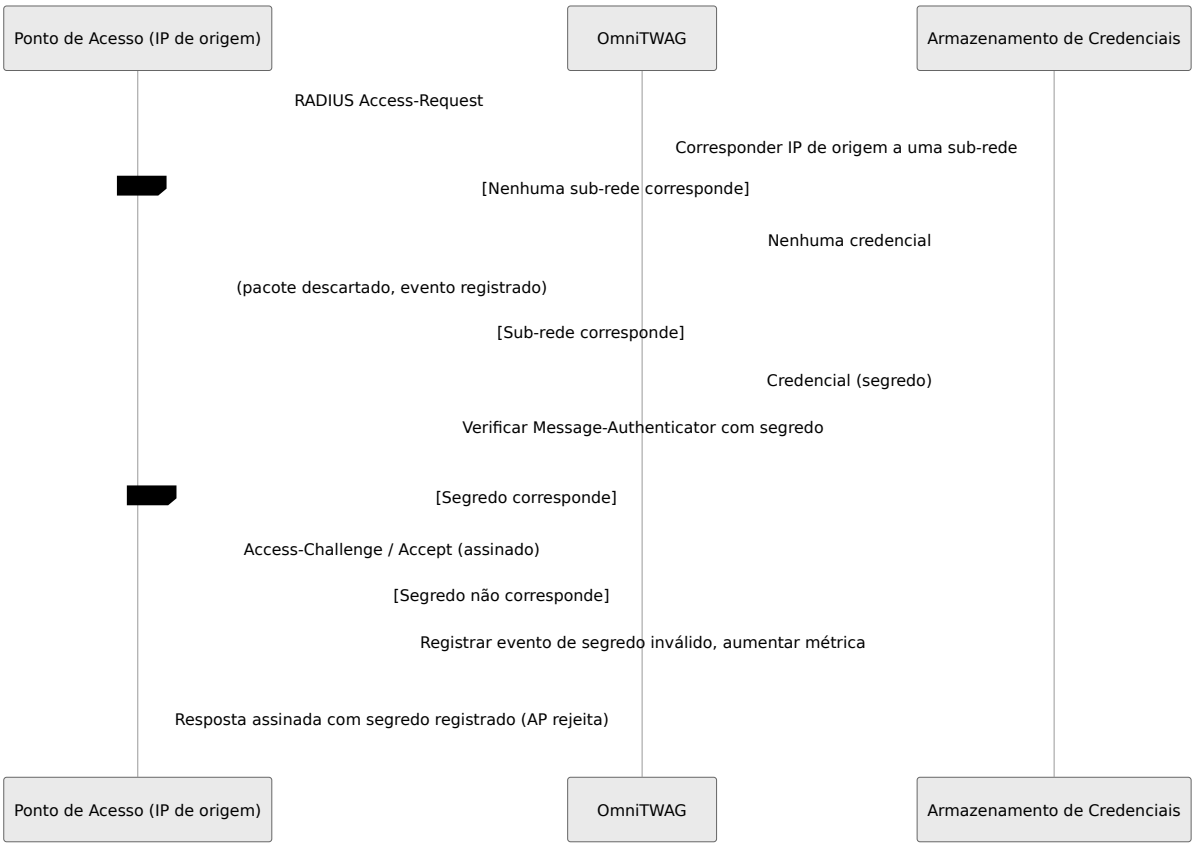


Como a autenticação usa uma credencial

OmniTWAG resolve o segredo compartilhado para cada pacote RADIUS recebido. Os passos são:

1. OmniTWAG lê o IP de origem do pacote.
2. OmniTWAG encontra a credencial cuja sub-rede corresponde ao IP de origem. Se mais de uma sub-rede corresponder, OmniTWAG usa a sub-rede com o **prefixo mais longo** (a correspondência mais específica).
3. Se nenhuma sub-rede corresponder, OmniTWAG descarta o pacote. A origem não está autorizada.
4. Se uma sub-rede corresponder, OmniTWAG usa o segredo dessa credencial para decodificar o pacote, verificar o `Message-Authenticator`, criptografar as chaves MPPE e assinar a resposta.

Se uma sub-rede corresponder, mas o `Message-Authenticator` não verificar, o ponto de acesso possui um segredo diferente do que está registrado. OmniTWAG registra esse evento e aumenta uma métrica. OmniTWAG ainda processa o pacote. A resposta é assinada com o segredo registrado, então o ponto de acesso rejeita a resposta e a autenticação falha. Esse comportamento torna fácil encontrar um segredo mal configurado. Ele não descarta o pacote, então um caso de borda válido nunca é descartado por engano.



Modelo de credencial

Cada credencial possui estes campos.

Campo	Tipo	Descrição
`id`	String	Identificador único (UUID). O servidor define este valor.
`name`	String	Rótulo legível por humanos, por exemplo `APs do CBD de Melbourne`.
`secret`	String	Segredo compartilhado RADIUS.
`subnets`	Lista de String	Uma ou mais sub-redes CIDR (IPv4, IPv6 ou uma mistura), por exemplo `["10.10.0.0/24", "2001:db8:a::/48"]`.
`breakout_type`	String	Manipulação do plano de dados para sessões deste ponto de acesso: `tunneled`, `nswo` ou `local`. Padrão `tunneled`. Veja Quebra e cobrança por AP.
`address_source`	String	Para quebra `nswo`, como o IP do UE é atribuído: `pool` ou `dhcp`. Padrão `pool`. Veja Quebra e cobrança por AP.
`charging_mode`	String	Comportamento de cobrança online Gy: `charged`, `uncharged` ou `auth_only`. Padrão `charged`. Veja Quebra e cobrança por AP.
`ip_pool_id`	String	ID opcional de um pool de IP que endereça sessões `nswo` deste ponto de acesso. `null` significa nenhum pool.
`inserted_at`	Integer	Hora em que a credencial foi criada, em milissegundos desde a época Unix. O servidor define este valor.
`updated_at`	Integer	Hora em que a credencial foi alterada pela última vez, em milissegundos desde a época

Campo	Tipo	Descrição
		Unix. O servidor define este valor.

O operador define `name`, `secret`, `subnets`, `breakout_type`, `address_source`, `charging_mode` e `ip_pool_id`. O servidor define `id`, `inserted_at` e `updated_at`.

Quebra e cobrança por AP

Cada credencial também controla o plano de dados e a cobrança para os pontos de acesso no grupo. Os padrões preservam o comportamento padrão: um túnel GTP para o PGW com cobrança online.

breakout_type

`breakout_type` define como o tráfego do assinante sai do TWAG.

Valor	Descrição
`tunneled`	OmniTWAG estabelece uma sessão GTP-C/GTP-U com o PGW e encaminha o tráfego do assinante pelo túnel. Este é o padrão. Veja Interface GTP-C S2a.
`nswo`	Offload WLAN Não-Suave. OmniTWAG ancla a sessão localmente e não faz túnel para o PGW. OmniTWAG atribui o IP do UE de um pool de IP. O tráfego do assinante transita pelo TWAG e sai para a rede local.
`local`	O ponto de acesso possui o plano de dados. OmniTWAG autentica o assinante e não atribui IP. O UE obtém seu endereço da rede local.

address_source

`address_source` aplica-se apenas quando `breakout_type` é `nswo`. Ele define como o IP do UE é atribuído. Veja Pools de IP.

Valor	Descrição
`pool`	OmniTWAG pré-aloca um endereço fixo do pool de IP associado para cada sessão. Este é o padrão.
`dhcp`	O servidor DHCP do OmniTWAG distribui um endereço do intervalo do pool quando o UE envia um DHCP DISCOVER.

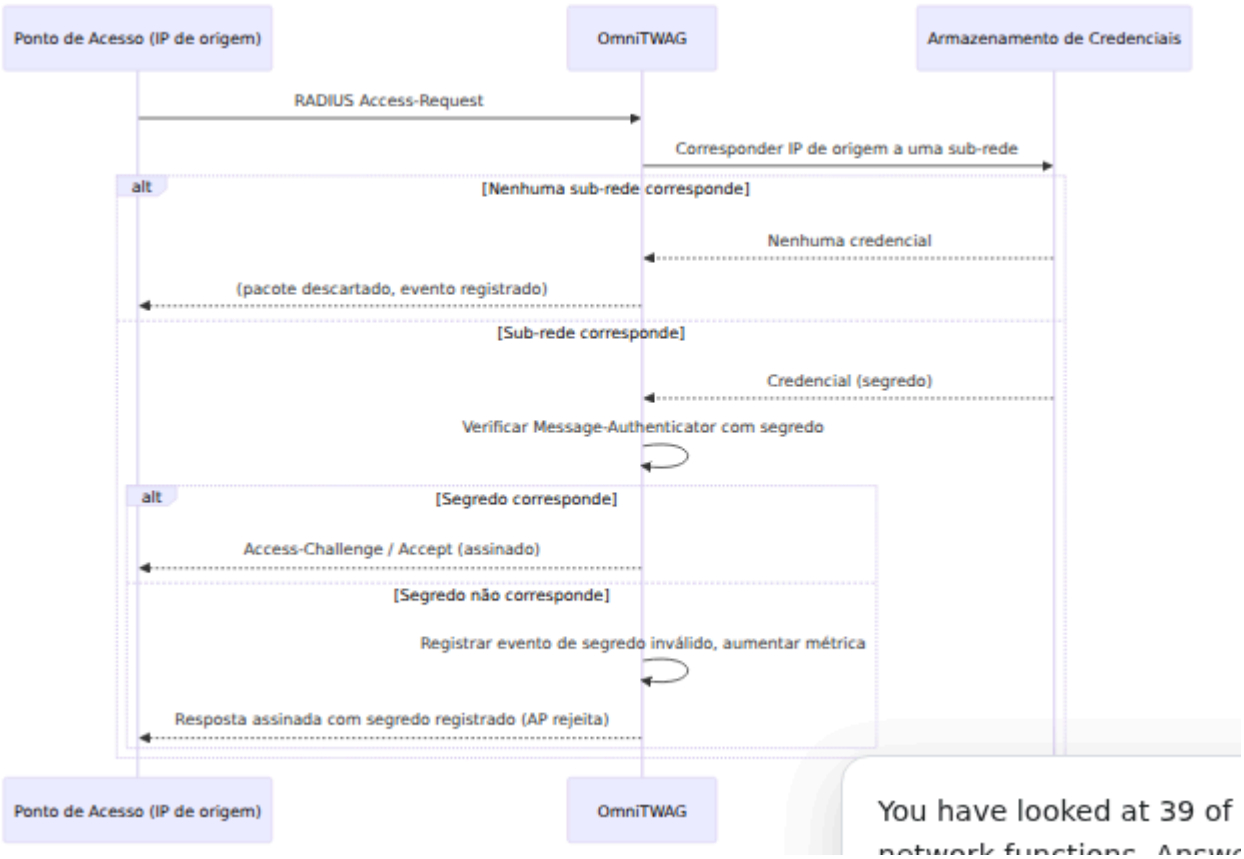
charging_mode

`charging_mode` define o comportamento de cobrança online Gy. Veja Cobrança Online.

Valor	Descrição
`charged`	OmniTWAG abre uma sessão de cobrança online Gy e relata o uso ao OCS. Este é o padrão.
`uncharged`	OmniTWAG estabelece a sessão de dados, mas não abre uma sessão Gy. Use este valor para acesso de tarifa fixa ou não cobrado.
`auth_only`	OmniTWAG autentica o assinante, mas não estabelece uma sessão de dados. Use este valor quando outro sistema fornecer conectividade.

Como os campos interagem

`charging_mode: auth_only` tem prioridade. Quando uma credencial usa `auth_only`, OmniTWAG não estabelece uma sessão de dados, então `breakout_type` e `address_source` não têm efeito.



API REST

A API REST do OmniTWAG serve estes endpoints sobre HTTPS na porta `api_ex` configurada (`8444` por padrão). O caminho base é `/radius_credentials`.

Método	Caminho	Ação	Descrição
GET	`/radius_credentials`	index	Listar todas as credenciais.
GET	`/radius_credentials/{id}`	show	Obter uma credencial por `id`.
POST	`/radius_credentials`	create	Criar uma credencial.
PUT	`/radius_credentials/{id}`	update	Alterar uma credencial.
DELETE	`/radius_credentials/{id}`	delete	Remover uma credencial.

Corpo da requisição

`POST` e `PUT` aceitam um corpo JSON com estes campos.

Campo	Tipo	Obrigatório	Descrição
`name`	String	Não	Rótulo legível por humanos.
`secret`	String	Sim	Segredo compartilhado RADIUS. Não deve estar vazio.
`subnets`	Lista de String	Sim	Uma ou mais sub-redes CIDR válidas. Não deve estar vazio.
`breakout_type`	String	Não	`tunneled`, `nswo` ou `local`. Padrão `tunneled`. Veja Quebra e cobrança por AP.
`address_source`	String	Não	`pool` ou `dhcp` (para `nswo`). Padrão `pool`.
`charging_mode`	String	Não	`charged`, `uncharged` ou `auth_only`. Padrão `charged`.
`ip_pool_id`	String	Não	`id` de um pool de IP para endereçamento `nswo`. Uma string vazia significa nenhum pool.

`PUT` altera apenas os campos no corpo. Campos que não estão no corpo mantêm seu valor atual.

Respostas

Status	Significado
`200 OK`	A requisição foi bem-sucedida (index, show, update, delete).
`201 Created`	A credencial foi criada.
`404 Not Found`	Nenhuma credencial possui o `id` fornecido.
`422 Unprocessable Entity`	O `secret` está vazio, uma sub-rede não é um bloco CIDR válido, ou `breakout_type` / `address_source` / `charging_mode` não é um valor válido.

Exemplo: listar credenciais

Requisição:

```
GET /radius_credentials
```

Resposta:

```
{
  "count": 1,
  "radius_credentials": [
    {
      "id": "0d1e2f3a-4b5c-6d7e-8f90-1a2b3c4d5e6f",
      "name": "APs do CBD de Melbourne",
      "secret": "s3cr3t",
      "subnets": ["10.10.0.0/24", "2001:db8:a::/48"],
      "inserted_at": 1734512400000,
      "updated_at": 1734512400000
    }
  ]
}
```

Exemplo: criar uma credencial

Requisição:

```
POST /radius_credentials
```

```
{
  "name": "APs do CBD de Melbourne",
  "secret": "s3cr3t",
  "subnets": ["10.10.0.0/24", "2001:db8:a::/48"]
}
```

Uma requisição bem-sucedida retorna `201 Created` com a nova credencial, incluindo seu `id`.

Exemplo: uma sub-rede inválida

Requisição:

```
{
  "secret": "s3cr3t",
  "subnets": ["10.10.0.0/24", "not-a-subnet"]
}
```

Resposta (`422 Unprocessable Entity`):

```
{
  "error": "sub-rede CIDR inválida(s): not-a-subnet"
}
```

Interface web

O painel de controle possui uma página **Segredos RADIUS** em `/radius_credentials`. A página lista cada credencial com seu nome, sub-redes, um segredo mascarado e a hora da última alteração. A página possui

um formulário para adicionar uma credencial e controles para editar ou excluir uma credencial. Insira uma ou mais sub-redes no campo de sub-redes. Separe as sub-redes com uma vírgula ou uma nova linha.

Configuração

O armazenamento de credenciais RADIUS mantém seus dados em um arquivo DETS. Defina o caminho do arquivo no bloco `radius_config`.

```
config :omnitwag,
  radius_config: %{
    # Caminho para o armazenamento persistente de credenciais
RADIUS.
    credentials_dets_path: "priv/radius_credentials.dets",
    # Campos de segredo único legados (veja "Migração" abaixo).
    allowed_source_subnets: [],
    secret: "123456"
  }
```

Parâmetro	Tipo	Obrigatório	Padrão
`credentials_dets_path`	String	Não	`priv/radius_credentia`
`secret`	String	Não	`123456`
`allowed_source_subnets`	Lista de String	Não	`[]`

Parâmetro	Tipo	Obrigatório	Padrão

Migração de um único segredo estático

Versões anteriores usavam um único `secret` estático e uma lista de permissões `allowed_source_subnets` opcional. Na primeira inicialização, se o armazenamento de credenciais estiver vazio, OmniTWAG inicializa uma credencial a partir desses campos legados. A inicialização ocorre apenas quando ambas as condições são verdadeiras:

- O armazenamento não possui credenciais.
- O `secret` está definido e `allowed_source_subnets` não está vazio.

A credencial inicializada usa o `secret` legado e as sub-redes legadas. Isso mantém uma implantação existente funcionando após uma atualização. OmniTWAG registra uma mensagem quando inicializa uma credencial.

Se `allowed_source_subnets` estiver vazio, OmniTWAG não inicializa uma credencial. Nesse caso, OmniTWAG rejeita todas as requisições RADIUS até que o operador adicione uma credencial. Adicione credenciais através da API REST ou da interface web.

A autenticação por pacote não lê `secret` ou `allowed_source_subnets`. Esses campos servem apenas como a inicialização da primeira inicialização. Gerencie credenciais ativas através da API ou da interface web.

Métricas

OmniTWAG expõe esses contadores no endpoint Prometheus. Veja a Referência de Métricas para o endpoint e exemplos de consulta.

radius_auth_unauthorized_source_count

Tipo: Contador
Descrição: Número de pacotes RADIUS descartados porque o IP de origem não corresponde a nenhuma sub-rede de credencial.

Rótulo	Descrição
`source`	IP de origem do pacote descartado.

radius_auth_bad_secret_count

Tipo: Contador
Descrição: Número de pacotes RADIUS cujo IP de origem corresponde a uma credencial, mas cujo `Message-Authenticator` n��o verificou. O ponto de acesso possui um segredo diferente do que está registrado.

Rótulo	Descrição
`source`	IP de origem do ponto de acesso.
`credential_id`	`id` da credencial correspondente.

Solução de Problemas

Um ponto de acesso não recebe resposta

Sintomas: O ponto de acesso envia requisições RADIUS, mas não recebe resposta. A métrica `radius_auth_unauthorized_source_count` aumenta.

Possíveis causas:

- Nenhuma sub-rede de credencial contém o IP de origem do ponto de acesso.

- O ponto de acesso envia de um IP diferente do esperado, por exemplo, um endereço NAT.

Resolução:

1. Encontre o IP de origem do ponto de acesso nos logs do OmniTWAG.
2. Confirme que uma sub-rede de credencial contém esse IP.
3. Se nenhuma credencial cobre o IP, adicione uma credencial para a sub-rede correta.

Um ponto de acesso está autorizado, mas a autenticação falha

Sintomas: A métrica `radius_auth_bad_secret_count` aumenta para o ponto de acesso. Os clientes não conseguem autenticar.

Possíveis causas:

- O ponto de acesso possui um segredo compartilhado diferente do que está registrado na credencial.

Resolução:

1. Leia o rótulo `credential_id` na métrica ou encontre a mensagem de segredo inválido nos logs.
2. Compare o segredo no ponto de acesso com o segredo na credencial.
3. Corrija o segredo no ponto de acesso ou atualize a credencial para corresponder.

Credenciais são perdidas após uma reinicialização

Sintomas: A lista de credenciais está vazia após uma reinicialização. OmniTWAG rejeita todas as requisições.

Possíveis causas:

- O `credentials_dets_path` aponta para um diretório que não é gravável ou para um caminho que é limpo na reinicialização.

Resolução:

1. Defina `credentials_dets_path` para um diretório de dados persistente e gravável.
2. Confirme que o processo OmniTWAG pode gravar nesse caminho.
3. Adicione as credenciais novamente através da API ou da interface web.

Rejeições RADIUS

Visão Geral

OmniTWAG rastreia fontes RADIUS cujos pacotes foram rejeitados. Isso permite que um operador identifique um ponto de acesso desconhecido ou mal configurado rapidamente. O recurso é a imagem espelhada de Segredos Compartilhados RADIUS: o armazenamento de credenciais registra as fontes que o OmniTWAG **aceita**, e o rastreador de rejeições registra as fontes que o OmniTWAG **rejeita**.

O rastreador mantém seus dados na memória. Os dados não são persistentes. O rastreador é redefinido na reinicialização. Isso é intencional. Veja Solução de Problemas.

Índice

- Conceitos
- Como um pacote chega ao rastreador
- Razões para rejeição
- Modelo de entrada
- Comportamento do armazenamento
- REST API
- Métricas
- Solução de Problemas

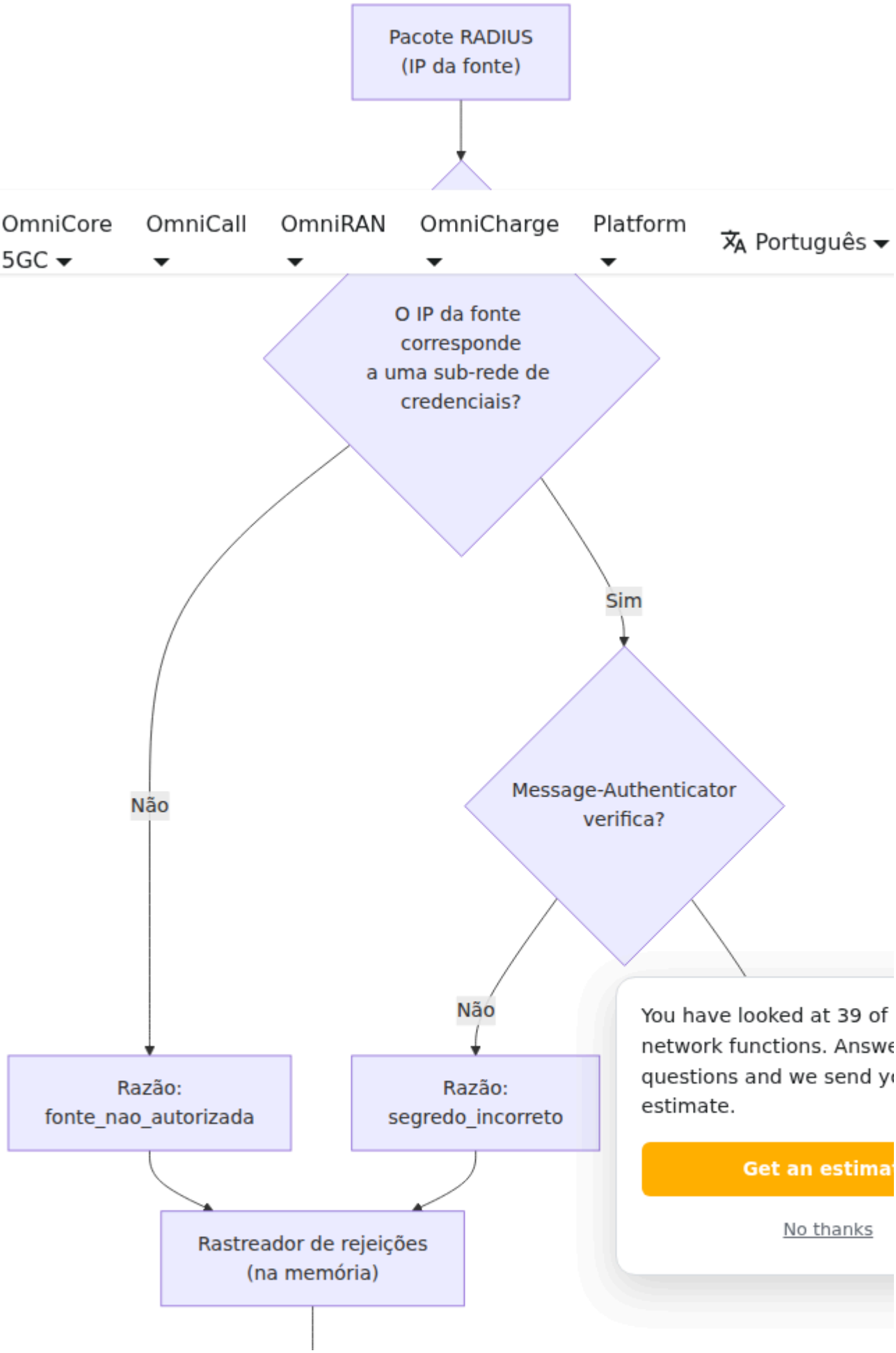
Conceitos

Uma **fonte rejeitada** é um endereço IP que enviou um pacote RADIUS que o OmniTWAG não aceitou. O rastreador registra uma entrada por par único de **fonte** e **razão**. Se a mesma fonte for rejeitada várias vezes pela mesma razão, o OmniTWAG aumenta um contador na entrada existente. O OmniTWAG não adiciona uma nova linha.

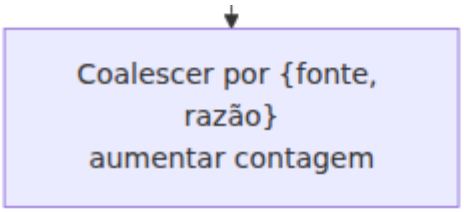
O **IP da fonte** é o endereço IP que enviou o pacote RADIUS. O OmniTWAG lê o IP da fonte do pacote UDP recebido. O OmniTWAG não usa o atributo `NAS-IP-Address`, porque um atacante pode definir esse atributo para qualquer valor.

Como um pacote chega ao rastreador

O OmniTWAG compara cada pacote RADIUS recebido com as sub-redes de credenciais. Um pacote é rejeitado quando falha em uma das duas verificações. O rastreador registra a fonte e a razão para cada pacote rejeitado.



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



Razões para rejeição

Uma fonte é rejeitada por uma das duas razões.

Razão	Significado
`fonte_nao_autorizada`	O IP da fonte não corresponde a nenhuma sub-rede de credenciais. O OmniTWAG descartou o pacote. Este é o caso de ponto de acesso desconhecido.
`segredo_incorreto`	Uma sub-rede de credenciais correspondeu, mas o `Message-Authenticator` do pacote não verificou. O ponto de acesso possui um segredo compartilhado diferente do que está registrado.

Para uma entrada `segredo_incorreto`, o rastreador também registra a credencial que correspondeu. Isso informa ao operador qual credencial verificar.

Modelo de entrada

Cada entrada possui estes campos.

Campo	Tipo	Descrição
`source`	String	IP da fonte do pacote rejeitado. Este valor é o identificador da entrada.
`reason`	String	Razão para a rejeição: `fonte_nao_autorizada` ou `segredo_incorreto`.
`count`	Integer	Número de vezes que o OmniTWAG rejeitou esta fonte por esta razão.
`first_seen`	Integer	Tempo em que a fonte foi rejeitada pela primeira vez, em milissegundos desde a época Unix.
`last_seen`	Integer	Tempo em que a fonte foi rejeitada pela última vez, em milissegundos desde a época Unix.
`credential_id`	String	`id` da credencial correspondente. Definido apenas quando a `reason` é `segredo_incorreto`.
`credential_name`	String	`name` da credencial correspondente. Definido apenas quando a `reason` é `segredo_incorreto`.

Os campos `credential_id` e `credential_name` estão vazios para uma entrada `fonte_nao_autorizada`, porque nenhuma credencial correspondeu.

Comportamento do armazenamento

O rastreador é um armazenamento na memória. Ele possui estas propriedades.

- **Não persistente.** O armazenamento é redefinido na reinicialização. Não sobrevive a uma reinicialização do processo.
- **Coalescido.** O OmniTWAG registra uma entrada por par de `source` e `reason`. Uma rejeição repetida aumenta o `count` e atualiza o `last_seen`. Não adiciona uma linha.
- **Limitado.** O armazenamento mantém um número máximo de fontes. O máximo padrão é 256.
- **Evicção pelo menos recentemente visto.** Quando o armazenamento está cheio, o OmniTWAG remove a entrada com o valor de `last_seen` mais antigo para abrir espaço para uma nova fonte. O armazenamento é limitado porque um IP de fonte pode ser falsificado. Uma inundação de IPs de fonte falsificados não pode crescer o armazenamento sem limites.

REST API

A API REST do OmniTWAG serve estes endpoints na porta API configurada (`8444` por padrão). O caminho base é `/radius_rejections`.

Método	Caminho	Ação	Descrição
GET	`/radius_rejections`	índice	Lista fontes rejeitadas. A lista é paginada. Pesquise por `source` ou `reason`.
DELETE	`/radius_rejections/{id}`	deletar	Esquecer uma fonte. `{id}` é o IP da fonte.

Use `DELETE` para esquecer uma fonte após resolvê-la, por exemplo, após adicionar uma credencial que cobre o IP da fonte.

Respostas

Status	Significado
`200 OK`	A solicitação foi bem-sucedida.
`404 Not Found`	Nenhuma entrada possui o IP da fonte fornecido.

Exemplo: listar fontes rejeitadas

Solicitação:

```
GET /radius_rejections
```

Resposta:

```
{
  "count": 2,
  "radius_rejections": [
    {
      "source": "203.0.113.7",
      "reason": "fonte_nao_autorizada",
      "count": 42,
      "first_seen": 1734512400000,
      "last_seen": 1734512999000,
      "credential_id": null,
      "credential_name": null
    },
    {
      "source": "10.10.0.15",
      "reason": "segredo_incorreto",
      "count": 3,
      "first_seen": 1734512500000,
      "last_seen": 1734512890000,
      "credential_id": "0d1e2f3a-4b5c-6d7e-8f90-1a2b3c4d5e6f",
      "credential_name": "APs do CBD de Melbourne"
    }
  ]
}
```

Exemplo: esquecer uma fonte

Solicitação:

```
DELETE /radius_rejections/203.0.113.7
```

Resposta:

```
{
  "deleted": "203.0.113.7"
}
```

Métricas

O rastreador de rejeições complementa dois contadores Prometheus que o recurso de credenciais documenta. Cada contador aumenta ao mesmo tempo que o OmniTWAG registra uma rejeição.

Contador	Razão da rejeição
`radius_auth_fonte_nao_autorizada_count`	`fonte_nao_autorizada`
`radius_auth_segredo_incorreto_count`	`segredo_incorreto`

Os contadores fornecem totais ao longo do tempo. O rastreador fornece uma visão ao vivo, por fonte. Veja Segredos Compartilhados RADIUS para as definições dos contadores, e a Referência de Métricas para o endpoint Prometheus e exemplos de consulta.

Solução de Problemas

Um ponto de acesso desconhecido aparece na lista

Sintomas: Uma fonte aparece com `reason` definida como `fonte_nao_autorizada`. O métrico `radius_auth_fonte_nao_autorizada_count` aumenta para essa fonte.

Possíveis causas:

- Nenhuma sub-rede de credenciais contém o IP da fonte.
- O ponto de acesso está atrás de NAT. O IP da fonte RADIUS que o OmniTWAG vê difere do endereço do próprio ponto de acesso.

Resolução:

1. Leia o campo `source` da entrada.
2. Confirme se este é um ponto de acesso conhecido.

3. Adicione uma sub-rede de credenciais que cubra o IP da fonte que o OmniTWAG realmente vê. Para um caso de NAT, use o endereço traduzido, não o endereço do ponto de acesso. Veja Segredos Compartilhados RADIUS.
4. Exclua a entrada com `DELETE /radius_rejections/{id}` para esquecer a fonte.

Um ponto de acesso conhecido aparece como segredo_incorreto

Sintomas: Uma fonte conhecida aparece com `reason` definida como `segredo_incorreto`. O métrico `radius_auth_segredo_incorreto_count` aumenta para essa fonte.

Possíveis causas:

- O ponto de acesso possui um segredo compartilhado diferente do que está registrado na credencial.

Resolução:

1. Leia os campos `credential_id` e `credential_name` da entrada. Estes nomeiam a credencial correspondente.
2. Compare o segredo no ponto de acesso com o segredo naquela credencial.
3. Corrija o segredo no ponto de acesso ou atualize a credencial para corresponder. Veja Segredos Compartilhados RADIUS.
4. Exclua a entrada com `DELETE /radius_rejections/{id}` para esquecer a fonte.

A lista está vazia após uma reinicialização

Sintomas: A lista de rejeições está vazia após uma reinicialização, mesmo que fontes tenham sido listadas antes da reinicialização.

Causa:

- Este é um comportamento esperado. O rastreador mantém seus dados na memória. Os dados não são persistentes. O armazenamento é redefinido

na reinicialização.

Resolução:

- Nenhuma ação é necessária. Novas rejeições repopulam a lista. Use os métricos `radius_auth_fonte_nao_autorizada_count` e `radius_auth_segredo_incorreto_count` para um histórico que sobrevive a uma reinicialização.

Visão Geral

S2a / Interface GTP-C para o PGW

Visão Geral

A interface S2a conecta o OmniTWAG ao Gateway de Rede de Dados de Pacote (PGW). Ela transporta conectividade PDN baseada em GTP para uma Rede de Acesso WLAN Confiável (TWAN). O OmniTWAG atua como o ponto final GTP-C do TWAN. Este papel é semelhante ao de um ePDG na interface S2b, mas utiliza tipos de interface S2a e um Identificador TWAN.

A interface S2a possui dois planos:

- **GTP-C (plano de controle)**: cria, modifica e exclui sessões PDN. Também realiza gerenciamento de caminho. Este documento cobre o GTP-C.
- **GTP-U (plano de usuário)**: transporta dados UE em túneis GTP. Consulte o guia de Plano de Usuário.

O GTP-C na S2a segue 3GPP TS 29.274 (GTPv2-C). O TWAG utiliza tipos de mensagem GTPv2-C e elementos de informação (IEs) com tipos de interface específicos da S2a.

Índice

- Onde S2a se Encaixa: Os Três Modos de Quebra
- Papel no Ciclo de Vida da Sessão
- Tipos de Interface S2a
- Estabelecimento de Sessão
  - Gatilho
  - Solicitação de Criação de Sessão
  - Bearer Padrão
  - Resposta de Criação de Sessão
  - Sessões de Emergência

- Gerenciamento de Bearer
- Desmontagem de Sessão
- Gerenciamento de Caminho
- Configuração
- Tabelas de Referência
- Observabilidade
- Solução de Problemas
- Referências 3GPP

Onde S2a se Encaixa: Os Três Modos de Quebra

O OmniTWAG resolve um modo de plano de dados por ponto de acesso (AP), a partir da credencial RADIUS que corresponde ao AP. O valor `breakout_type` na credencial seleciona um dos três modos. Consulte Credenciais RADIUS para as configurações de `breakout_type`, `address_source` e `ip_pool_id`.

Modo (`breakout_type`)	Quem atribui o IP do UE	Plano de dados	S2a / GTP utilizado
`tunneled`	PGW (no IE de Alocação de Endereço PDN)	GRE do AP para o TWAG, depois GTP-U na S2a para o PGW, depois para a internet pelo PGW	Sim. Este documento.
`nswo`	TWAG, de um pool de IP	GRE do AP para o TWAG, depois quebra local para a internet (NAT) no TWAG	Não. Sem sessão PGW.
`local`	A rede local no AP	O AP possui o plano de dados. O TWAG não transporta tráfego.	Não. Sem sessão PGW, sem IP.

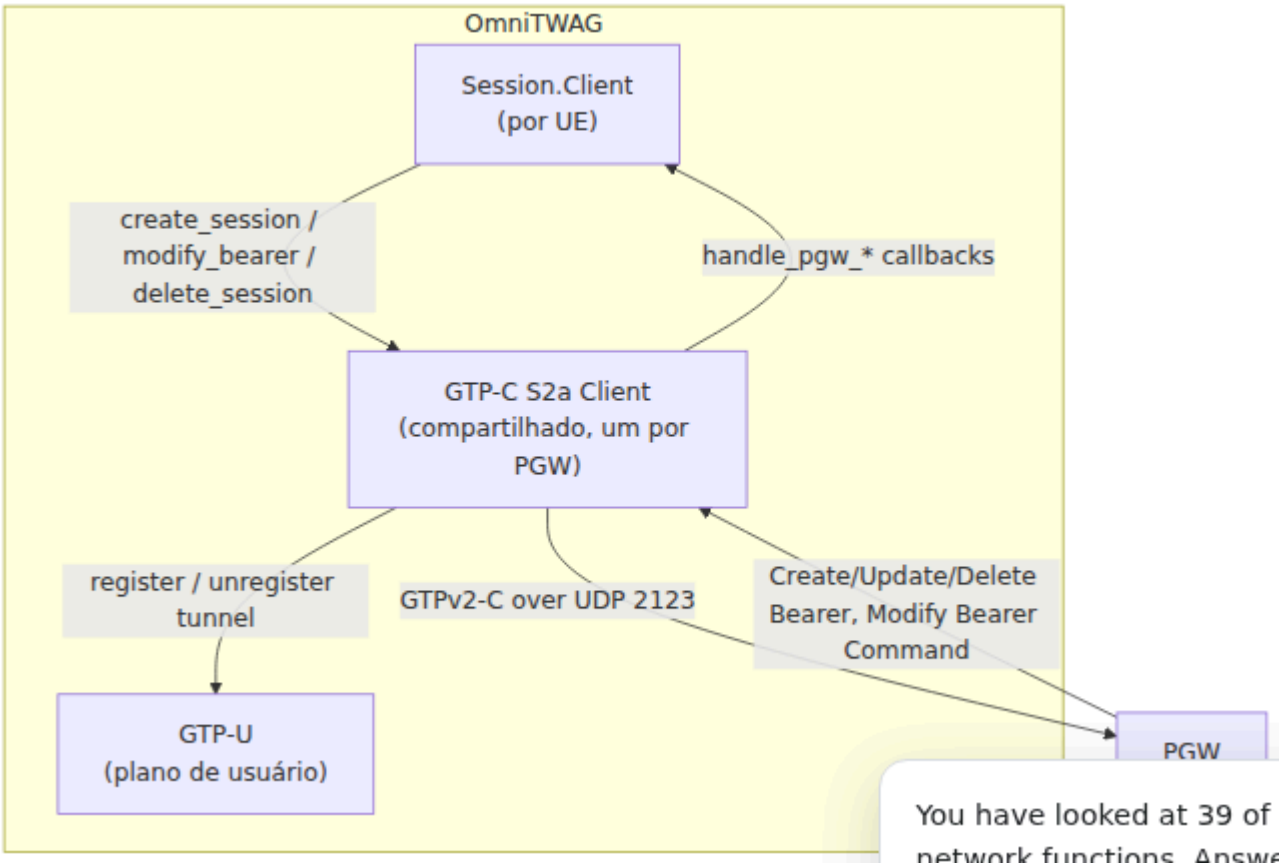
O GTP-C S2a se aplica apenas ao modo `tunneled`. No modo `tunneled`, o PGW aloca o endereço IP do UE e aplica políticas e cobrança. Nos modos `nswo` e `local`, o TWAG não cria uma sessão PDN, portanto, nenhuma mensagem S2a é enviada.

A cobrança no modo `tunneled` é feita pelo PGW, não pelo TWAG. A cobrança online do lado do TWAG em Cobrança Online se aplica aos modos quebrados localmente.

> Nota: Um manipulador global separado de Offload WLAN Não-Suave (NSWO) (`Twag.NSWO.Handler`, habilitado com `nswo.enabled: true`) fornece quebra local para cada sessão quando o GTP não é utilizado. Consulte a seção Quebra Local do guia do Plano de Usuário. O tipo de quebra `nswo` por AP descrito acima é o modelo baseado em credenciais e é preferido.

Papel no Ciclo de Vida da Sess��o

O cliente GTP-C é um único processo compartilhado (`Twag.Gtp.GtpcS2a`). Ele gerencia todas as sessões contra um PGW configurado. Cada sessão UE `tunneled` autenticada aciona uma troca de Criação de Sessão. O processo `Session.Client` por sessão chama o cliente GTP-C. Ele não envia mensagens GTPv2-C por si só.



Tipos de Interface S2a

O TWAG utiliza tipos de interface F-TEID S2a. Estes são diferentes dos S2b (ePDG), que utilizam os tipos 30 e 31. Os valores seguem TS 29.274 Tabela 8.22-1.

Tipo de Interface	Valor	Direção	Uso
`s2a_twan_gtp_c`	35	Controle TWAG	F-TEID TWAG na Solicitação de Criação de Sessão
`s2a_twan_gtp_u`	34	Usuário TWAG	F-TEID bearer TWAG (GTP-U)
`s2a_pgw_gtp_c`	36	Controle PGW	F-TEID PGW na Resposta de Criação de Sessão
`s2a_pgw_gtp_u`	37	Usuário PGW	F-TEID bearer PGW (GTP-U)

O Tipo RAT é sempre WLAN (valor 3).

Estabelecimento de Sessão

Gatilho

Uma autenticação EAP bem-sucedida inicia a sessão. O AP envia um Accounting-Start, que aciona o processamento pós-aceitação. O `Session.Client` resolve o modo de quebra do AP. Para o modo `tunneled`, ele chama o cliente GTP-C para criar a sessão PDN. Se o parâmetro `gtp.enabled` estiver definido como `false`, o TWAG ignora a S2a. Se a sessão não tiver IMSI, o TWAG não cria a sessão. Para os modos `nswo` e `local`, o TWAG não envia nenhuma mensagem S2a. Consulte Sessões para o ciclo de vida completo da sessão.

Solicitação de Criação de Sessão

O TWAG envia uma Solicitação de Criação de Sessão (tipo de mensagem conforme TS 29.274 §7.2.1) para o PGW. O TWAG gera um novo TEID de plano de controle não previsível e um TEID de plano de usuário para a solicitação. Ele constrói a solicitação com estes IEs:

Elemento de Informação	Fonte / Valor	Notas
IMSI	Assinatura UE	Omitido se não estiver presente.
Rede Servidora (MCC/MNC)	`gtp.mcc` / `gtp.mnc`, ou primeiros dígitos do IMSI	Omitido se algum valor estiver faltando.
Tipo RAT	WLAN (3)	Fixo para S2a.
F-TEID (controle)	Tipo de interface 35, TEID de controle TWAG, IP local TWAG	Identifica o ponto final GTP-C do TWAG.
Nome do Ponto de Acesso (APN)	`gtp.apn`, padrão `internet`. APN de emergência se solicitado.	Consulte Sessões de Emergência.
Modo de Seleção	2 (fornecido pela rede, assinatura não verificada)	O APN é sempre fornecido pela rede no TWAN.
Alocação de Endereço PDN (PAA)	Tipo solicitado com um endereço zero	O TWAG solicita um endereço. O PGW o atribui.
Taxa Máxima Agregada de Bits (APN-AMBR)	Padrão 256000 uplink / 256000 downlink	Substituído se uma taxa for fornecida.
Contexto de Bearer a Criar	EBI padrão 5, F-TEID bearer (tipo 34), QoS	Consulte Bearer Padrão.

Elemento de Informação	Fonte / Valor	Notas
Restrição de APN	2 (normal), ou 0 para emergência	
Identidade do ME (IMEISV)	Identidade do equipamento UE	Omitido se não estiver presente.
Recuperação	Contador de recuperação do TWAG	Usado para detecção de reinício.
Características de Cobrança	`0x0800`	Valor fixo.
Identificador TWAN	SSID de `twan_ssid`, BSSID opcional de `twan_bssid`	Identifica a WLAN.
Fuso Horário do UE	Derivado do deslocamento de hora local do host	
Indicação	Sinal de transferência	O sinal de transferência é definido para uma transferência de entrada.

O tipo de PDN padrão é IPv4. Ele pode ser definido como `:ipv4`, `:ipv6` ou `:ipv4v6` através do parâmetro `gtp.pdn_type`. O endereço PAA solicitado é sempre zero, então o PGW atribui o endereço real.

> Nota: O IE de Opções de Configuração de Protocolo Adicionais (APCO) não está incluído. O TWAG não solicita endereços DNS ou P-CSCF através do APCO neste momento. Ele lê DNS do PCO ou IE PCO Adicional na resposta se o PGW fornecer um.

Bearer Padrão

O TWAG cria um bearer padrão por sessão. O EBI padrão é 5. O bearer não possui filtros TFT, portanto, transporta todo o tráfego. Os valores de QoS de nível bearer na solicitação utilizam estes valores fixos:

Parâmetro QoS	Valor
QCI	9
Nível de Prioridade	10
Capacidade de Pré-emissão	desativada
Vulnerabilidade à Pré-emissão	ativada
GBR / MBR (uplink e downlink)	0

O TWAG não suporta bearers dedicados. Consulte Gerenciamento de Bearer.

Resposta de Criação de Sessão

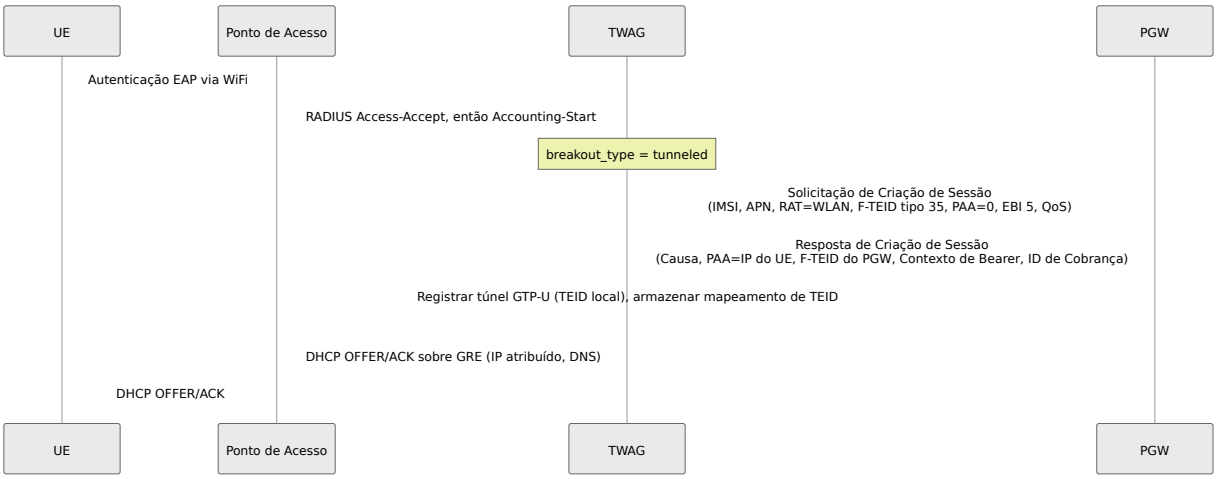
O TWAG combina a resposta com a solicitação pendente pelo número de sequência. Ele verifica o IE de Causa. A solicitação é um sucesso se a causa for `request_accepted`, `new_pdp_type_due_to_network_preference`, ou `new_pdp_type_due_to_single_address_bearer_only`. Qualquer outra causa retorna um `gtpc_error` com esse valor de causa.

Em caso de sucesso, o TWAG lê estes valores da resposta:

Elemento de Resposta	Valor Extraído	Uso
Alocação de Endereço PDN (PAA)	Endereço IP do UE atribuído	Entregue ao UE por DHCP sobre GRE. Consulte Gerenciamento de Endereço IP.
F-TEID (controle PGW)	TEID de controle PGW e IP PGW	Usado como destino para mensagens de controle posteriores.
Contexto de Bearer	F-TEID bearer (TEID e IP GTP-U do PGW), EBI, ID de Cobrança	Usado para registrar o túnel GTP-U.
APN-AMBR	Limites de taxa da sessão	Armazenado na sessão.
PCO / PCO Adicional	Endereços de servidores DNS	Entregues ao UE por DHCP.

O TWAG armazena um mapeamento de seu TEID de controle local para o TEID de controle do PGW e a identidade do UE. Esse mapeamento roteia mensagens iniciadas pelo PGW posteriormente para a sessão correta. Se nenhum F-TEID bearer estiver presente, o TWAG recorre ao F-TEID do PGW para o túnel de plano de usuário.

Após um sucesso, o `Session.Client` registra o túnel GTP-U (que gera o TEID de plano de usuário local do lado do TWAG), pré-aloca o lease DHCP com o IP e DNS atribuídos, e registra o UE para roteamento de downlink GRE. Consulte Plano de Usuário para o túnel e o caminho de downlink.



Sessões de Emergência

Se a sessão solicitar serviço de emergência, o TWAG utiliza o APN de emergência. O parâmetro `emergency_apn` define este APN. O valor padrão é `sos`. Para uma sessão de emergência, a Restrição de APN é definida como 0 (sem restrição).

Gerenciamento de Bearer

Apenas Bearer Padrão

O OmniTWAG suporta apenas o bearer padrão. Ele não suporta bearers dedicados. Isso segue o perfil de Modo de Conexão Única Transparente (TSCM) do TWAN. O gerenciador de bearer pode rastrear uma tabela de bearers indexada por EBI, mas a S2a estabelece apenas o bearer padrão.

Modificar Bearer (Iniciado pelo TWAG)

O TWAG pode enviar uma Solicitação de Modificação de Bearer (TS 29.274 §7.2.7). Esta solicitação transporta um contexto de bearer com o EBI padrão, o F-TEID de plano de usuário do TWAG (tipo 34) e o IP local do TWAG. O TEID de destino é o TEID de controle do PGW. O TWAG combina a Resposta de Modificação de Bearer pelo número de sequência e verifica o IE de Causa para sucesso. Uma transferência de entrada reutiliza o caminho de Criação de

Sessão com o sinal de transferência definido, não uma Modificação de Bearer separada.

Procedimentos Iniciados pelo PGW

O PGW pode iniciar procedimentos de bearer. O TWAG recebe esses em seu socket GTP-C. O TEID de destino na mensagem é o TEID de controle local do TWAG. O TWAG usa o mapeamento de TEID armazenado para encontrar o TEID de controle do PGW e a identidade do UE. Em seguida, notifica o processo `Session.Client` correspondente.

Mensagem PGW	Resposta TWAG	Comportamento
Solicitação de Criação de Bearer	Resposta de Criação de Bearer, causa `request_rejected`	O TWAG rejeita todos os bearers dedicados (TSCM). Nenhum bearer é criado.
Comando de Modificação de Bearer	Solicitação de Modificação de Bearer de volta para o PGW	O TWAG reconhece os novos parâmetros (APN-AMBR, F-TEID). Se o envio falhar, ele envia uma Indicação de Falha de Modificação de Bearer com causa `system_failure`.
Solicitação de Atualização de Bearer	Resposta de Atualização de Bearer, causa `request_accepted` por bearer	O TWAG aceita a atualização de QoS e notifica a sessão.
Solicitação de Exclusão de Bearer	Resposta de Exclusão de Bearer, causa `request_accepted`	Consulte Desmontagem de Sessão.

Para um Comando de Modificação de Bearer, o TWAG lê o APN-AMBR e o contexto de bearer (EBI, novo F-TEID, QoS). Ele notifica o processo da sessão,

então envia uma Solicitação de Modificação de Bearer para o PGW para reconhecimento. Se o contexto de bearer carregar um novo F-TEID GTP-U, a sessão registra novamente o túnel GTP-U com os novos valores remotos. Para uma Solicitação de Atualização de Bearer, o TWAG lê o APN-AMBR e cada contexto de bearer (EBI, QoS), notifica a sessão, então aceita cada bearer na resposta.

Desmontagem de Sessão

Exclusão de Sessão (Iniciada pelo TWAG)

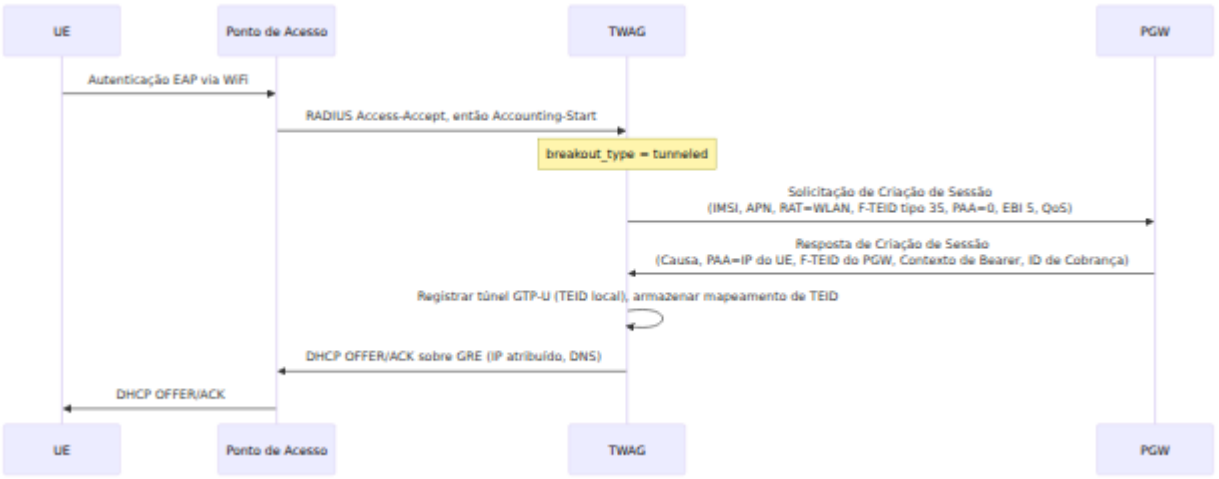
O TWAG exclui a sessão PDN quando o UE se desconecta. A desmontagem do `Session.Client` chama o cliente GTP-C. O TWAG envia uma Solicita��ão de Exclusão de Sessão (TS 29.274 §7.2.9) para o TEID de controle do PGW. A solicitação transporta:

- ID de Bearer EPS (o EBI padrão, padrão 5)
- Causa (padrão `local_detach`)

O TWAG combina a Resposta de Exclusão de Sessão pelo número de sequência e verifica o IE de Causa. Ele então remove as entradas do mapeamento de TEID para aquele TEID de controle do PGW. Se a sessão não tiver TEID de controle do PGW (por exemplo, uma sessão `nswo` ou `local`), o TWAG ignora a exclusão GTP-C. Após a exclusão, o `Session.Client` desregistra o túnel GTP-U, libera o lease DHCP e qualquer endereço de pool de IP, desregistra o UE do GRE e limpa a sessão.

Exclusão de Bearer (Iniciada pelo PGW)

O PGW pode iniciar a desmontagem com uma Solicitação de Exclusão de Bearer. O TWAG lê o EBI vinculado, a lista de EBI e a causa. Ele notifica o processo `Session.Client` correspondente, que faz uma limpeza apenas local. O TWAG não envia uma Solicitação de Exclusão de Sessão neste caso, porque o PGW já excluiu o bearer. O TWAG então envia uma Resposta de Exclusão de Bearer com causa `request_accepted`. Se o EBI vinculado for o bearer padrão, toda a conexão PDN é desmontada.



Gerenciamento de Caminho

Echo

O TWAG executa o gerenciamento de caminho GTPv2-C com Solicitação de Echo e Resposta de Echo (TS 29.274 §6). O parâmetro `echo_enabled` ativa isso. Ele está ativado por padrão. O TWAG envia uma Solicitação de Echo para o PGW a cada 60 segundos. Cada Solicitação de Echo transporta o número de sequência do TWAG. Cada Resposta de Echo transporta um contador de Recuperação.

Quando o PGW envia uma Solicitação de Echo, o TWAG responde com uma Resposta de Echo que transporta o contador de Recuperação do TWAG.

Estado do Caminho

O TWAG rastreia o estado do caminho para o PGW. Uma Resposta de Echo bem-sucedida marca o caminho como UP e redefine a contagem de falhas. Falhas consecutivas de Echo incrementam um contador. Após 3 falhas consecutivas, o TWAG marca o caminho como DOWN.

Detecção de Reinício do PGW

O TWAG armazena o contador de Recuperação do PGW de cada Resposta de Echo. Se o contador mudar, o TWAG detecta um reinício do PGW. Ele então

limpa todas as sessões afetadas. Notifica cada processo de sessão com uma causa `pgw_restart` e limpa o mapeamento de TEID.

Timeout de Solicitação e Retransmissão

O TWAG usa um temporizador T3-RESPONSE e N3-REQUESTS para tentativas de solicitações de controle.

Temporizador / Contador	Valor	Significado
T3-RESPONSE	3000 ms	Tempo para esperar por uma resposta antes de uma retransmissão.
N3-REQUESTS	3	Máximo de retransmissões antes que a solicitação falhe.

Se uma solicitação expirar, o TWAG retransmite a mesma mensagem com o mesmo número de sequência. Ele reinicia o temporizador. Após 3 tentativas sem resposta, o TWAG falha a solicitação. Uma falha de Criação de Sessão retorna `timeout` para o chamador. Retransmissões de Echo seguem os mesmos limites de temporizador e tentativas. Uma falha de Echo conta para o limite de queda do caminho.

O TWAG abre o socket UDP GTP-C na inicialização. Se o socket não puder ser aberto, o TWAG tenta novamente a cada 5 segundos.

Configuração

A S2a é configurada sob o mapa `gtp`. Este mapa é parte da configuração da aplicação `omnitwag` em `runtime.exs`.

```
config :omnitwag,
  gtp: %{
    # Habilitar a interface S2a (necessária para quebra tunelada)
    enabled: true,

    # Ponto final GTP-C do PGW
    pgw_ip: "10.5.198.1",
    pgw_port: 2123,

    # IP local do TWAG para F-TEIDs GTP-C e GTP-U
    local_ip: "10.5.198.200",

    # Porta do plano de usuário GTP-U
    gtpu_port: 2152,

    # Gerenciamento de caminho (Echo)
    echo_enabled: true,

    # Padrões de Solicitação de Criação de Sessão
    apn: "internet",
    pdn_type: :ipv4
  }
```

Parâmetros GTP

Parâmetro	Tipo	Necessário	Padrão	Descriçã
`enabled`	Booleano	Não	`false`	Habilitar a interface S2a Defina como `true` para quebra tune Quando `fal` TWAG não in o cliente GTF ignora a S2a para cada sessão.
`pgw_ip`	String	Sim (se habilitado)	`127.0.0.1`	Endereço IP C do PGW. O TWAG envia todas as mensagens GTPv2-C par este endereç
`pgw_port`	Inteiro	Não	`2123`	Porta UDP GT do PGW. A p GTP-C padrã 2123.
`local_ip`	String	Não	Detectado automaticamente	IP local do TW Usado em F- TEIDs e com fonte para G e GTP-U. Se definido, o T detecta

Parâmetro	Tipo	Necessário	Padrão	Descriçã
				automatican o IP de saída
`gtpu_port`	Inteiro	Não	`2152`	Porta UDP G para o plano usuário. Con Plano de Usu
`echo_enabled`	Booleano	Não	`true`	Habilitar gerenciamer de caminho GTPv2-C par PGW.
`apn`	String	Não	`internet`	Nome do Por de Acesso pa sessão PDN.
`pdn_type`	Átomo	Não	`:ipv4`	Tipo de PDN `:ipv4`, `:ipv` `:ipv4v6`.
`mcc`	String	Não	Primeiros 3 dígitos do IMSI	Código do Pa Móvel para c da Rede Servidora.
`mnc`	String	Não	Dígitos 4-5 do IMSI	Código da Re Móvel para c da Rede Servidora.

Parâmetros de Emergência e TWAN

Os seguintes parâmetros são lidos do ambiente da aplicação `omnitwag`, não do mapa `gtp`.

Parâmetro	Tipo	Padrão	Descrição
`emergency_apn`	String	`sos`	APN usado para uma sessão de emergência. A Restrição de APN é definida como 0 para essas sessões.
`twan_ssid`	String	`TWAG`	SSID colocado no IE do Identificador TWAN.
`twan_bssid`	Binário	(não definido)	BSSID opcional de 6 bytes para o IE do Identificador TWAN. Omitido se não for um valor válido de 6 bytes.

> Nota: O cliente GTP-C conecta-se a um único PGW. Não h`` seleção por APN ou múltiplos PGWs neste momento. Para mudar o PGW, atualize `pgw_ip`.

Tabelas de Referência

Tipos de Mensagem GTPv2-C (S2a)

Mensagem	Direção	Propósito
Solicitação / Resposta de Echo	Ambos	Gerenciamento de caminho.
Solicitação / Resposta de Criação de Sessão	TWAG para PGW	Estabelecer a sessão PDN.
Solicitação / Resposta de Modificação de Bearer	TWAG para PGW	Atualizar o F-TEID do bearer ou QoS.
Solicitação / Resposta de Exclusão de Sessão	TWAG para PGW	Desmontar a sessão PDN.
Solicitação / Resposta de Criação de Bearer	PGW para TWAG	Bearer dedicado. Rejeitado pelo TWAG.
Solicitação / Resposta de Atualização de Bearer	PGW para TWAG	Atualizar QoS. Aceito pelo TWAG.
Comando de Modificação de Bearer	PGW para TWAG	PGW pede ao TWAG para atualizar os parâmetros do bearer.
Indicação de Falha de Modificação de Bearer	TWAG para PGW	Enviado quando um Comando de Modificação de Bearer não pode ser honrado.
Solicitação / Resposta de Exclusão de Bearer	PGW para TWAG	Desmontagem iniciada pelo PGW.

Códigos de Causa GTPv2-C

O TWAG utiliza estes valores de causa. Os códigos seguem TS 29.274 §8.4.

Código	Nome	Significado
2	Desconexão Local	Causa padrão para uma Exclusão de Sessão iniciada pelo TWAG.
13	Falha de Rede	Falha de rede relatada pelo par.
16	Solicitação Aceita	Sucesso.
64	Contexto Não Encontrado	O par não possui contexto correspondente.
72	Falha de Sistema	Usado em uma Indicação de Falha de Modificação de Bearer.
92	Autenticação do Usuário Falhou	Autenticação rejeitada.
93	Acesso ao APN Negado	O APN não é permitido.
94	Solicitação Rejeitada	Usado quando o TWAG rejeita uma Solicitação de Criação de Bearer.
96	IMSI/IMEI Não Conhecido	O assinante não é conhecido.

Observabilidade

O TWAG emite eventos de telemetria para atividade de bearer e desmontagem. Esses eventos impulsionam as métricas na Referência de Métricas.

Evento	Quando
`[:twag, :bearer, :modify_command]`	O TWAG recebe um Comando de Modificação de Bearer do PGW.
`[:twag, :bearer, :update]`	O TWAG recebe uma Solicitação de Atualização de Bearer do PGW.
`[:twag, :detach, :pgw_delete]`	O TWAG recebe uma Solicitação de Exclusão de Bearer do PGW.
`[:twag, :session, :teardown]`	A sessão é desmontada.
`[:twag, :detach, :cleanup_start]`	A desmontagem da sessão começa.
`[:twag, :detach, :cleanup_complete]`	A desmontagem da sessão é concluída.

Status do Cliente

O cliente GTP-C relata uma captura de status. Este status mostra a saúde do caminho S2a.

Campo	Significado
`connected`	O socket UDP GTP-C está aberto.
`pgw_ip` / `pgw_port`	O ponto final PGW configurado.
`pending_requests`	Contagem de solicitações GTP-C em andamento.
`path_up`	O estado do caminho Echo para o PGW.
`echo_failures`	Contagem de falhas consecutivas de Echo.
`recovery_counter`	O contador de recuperação do TWAG.

Registro

O cliente GTP-C registra cada procedimento. As principais linhas de registro incluem:

- Criação de Sessão, Exclusão de Sessão e envios e respostas de Modificação de Bearer.
- Mudanças de estado do caminho: `Caminho para PGW ... está UP` e `Caminho para PGW ... está DOWN`.
- Detecção de reinício do PGW com os contadores de Recuperação antigos e novos.
- Mensagens de Criação, Atualização, Modificação e Exclusão de Bearer iniciadas pelo PGW.

Solução de Problemas

Falhas na Criação de Sessão

Sintomas: Sessões não são estabelecidas. O registro mostra `GTP Create Session failed`.

Possíveis causas:

- O PGW rejeita a solicitação com uma causa não de sucesso.
- O caminho para o PGW está down.
- O APN não está provisionado no PGW.
- O IMSI não é conhecido pelo núcleo.

Resolução:

1. Verifique o código de causa retornado na tabela de Códigos de Causa.
2. Verifique o estado do caminho com o status do cliente (`path_up`).
3. Confirme que o APN está provisionado no PGW.
4. Confirme que o assinante está provisionado no HSS.

Caminho para o PGW está Down

Sintomas: O registro mostra `Caminho para PGW ... está DOWN`. Novas sessões falham.

Possíveis causas:

- O PGW é inalcançável na porta UDP 2123.
- Um firewall bloqueia o GTP-C.
- O PGW não está em execução.

Resolução:

1. Confirme que os valores `pgw_ip` e `pgw_port` estão corretos.
2. Confirme que o caminho de rede permite UDP 2123 em ambas as direções.
3. Confirme que o serviço GTP-C do PGW está em execução.
4. Aguarde o próximo Echo. Uma Resposta de Echo bem-sucedida marca o caminho como UP.

Sessões Caem Após um Reinício do PGW

Sintomas: Todas as sessões são desmontadas juntas. O registro mostra `PGW restart detected`.

Possíveis causas:

- O PGW reiniciou e alterou seu contador de Recuperação.

Resolução:

1. Este é um comportamento esperado. O TWAG limpa sessões obsoletas após um reinício do PGW.
2. Confirme que o PGW está estável. UEs reautenticam e restabelecem sessões.

Sessões Estabelecidas, mas Sem Fluxos de Dados

Sintomas: O UE recebe um endereço IP, mas o tráfego não passa.

Possíveis causas:

- O túnel GTP-U ou o caminho GRE não estão ativos.

Resolução:

1. Confirme que `gtp.enabled` e `gre.enabled` estão ambos definidos como `true` para quebra tunelada.
2. Verifique as estatísticas GTP-U e GRE no guia do Plano de Usuário.

Referências 3GPP

Especificação	Título
TS 29.274	Protocolo de Tunelamento GPRS Evoluído para o plano de Controle (GTPv2-C).
TS 23.402	Melhorias de arquitetura para acessos não-3GPP (TWAN, S2a).
TS 24.302	Acesso ao EPC via redes de acesso não-3GPP.

Visão Geral

Sessões

Visão geral

Uma **sessão** representa um assinante autenticado no OmniTWAG. O OmniTWAG identifica uma sessão pela identidade do assinante (o IMSI ou nome de usuário). O OmniTWAG mantém as sessões na memória. A API de Sessões fornece o estado da sessão ao vivo.

A visualização de Sessões também transporta o tráfego do assinante. O OmniTWAG mescla o registro de contabilidade RADIUS na sessão pelo `Acct-Session-Id`. Portanto, uma sessão é a única fonte tanto para o estado de um assinante quanto para o tráfego do assinante. Não há um endpoint de contabilidade separado. A contabilidade é parte da sessão.

Índice

- Conceitos
- Ciclo de vida do status de autenticação
- Modelo de sessão
- Referência de status de autenticação
- API REST
- Ações por sessão
- Exemplos de JSON
- Interface web
- Solução de problemas

Conceitos

Cada assinante autenticado tem uma sessão. O OmniTWAG cria a sessão quando um Access-Request RADIUS inicia uma autenticação EAP. O OmniTWAG atualiza a sessão à medida que a autenticação avança e à medida que o caminho de dados é estabelecido.

Uma sessão rastreia a identidade do assinante, o ponto de acesso, o MAC do cliente, o endereço IP do UE alocado, o método e fase EAP, e o estado GTP e DNS. Quando a contabilidade RADIUS chega para o assinante, o OmniTWAG mescla os contadores de tráfego na sessão.

A forma da sessão é aberta. As chaves que estão presentes dependem do método de autenticação e de quão longe a sessão progrediu. Uma sessão que apenas começou o EAP tem menos chaves do que uma sessão que alcançou o RADIUS Accept e estabeleceu um caminho de dados GTP e DHCP.

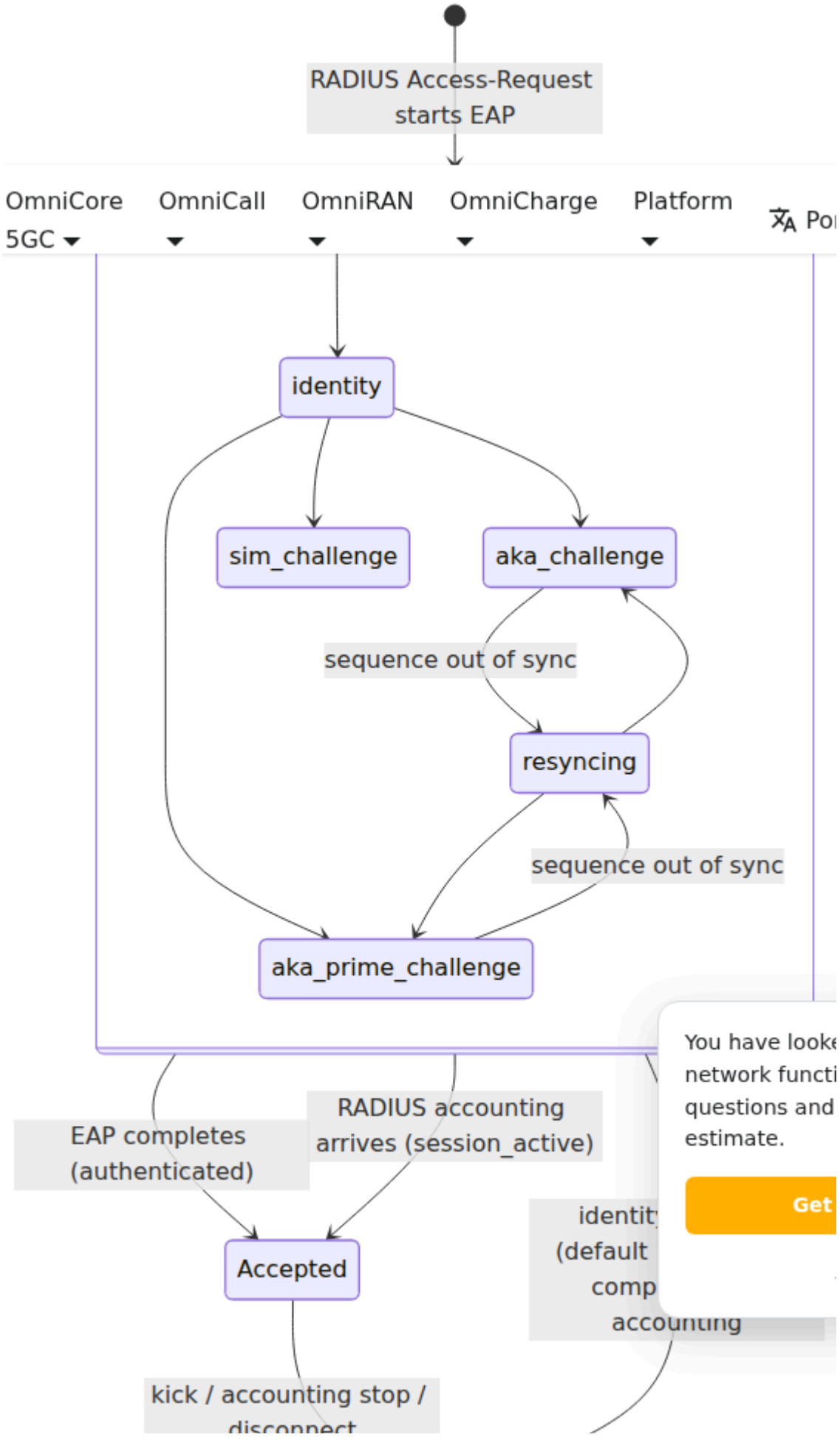
Ciclo de vida do status de autenticação

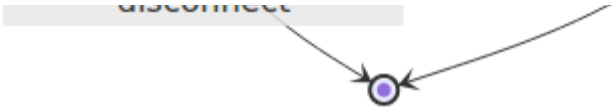
Uma sessão tem um status de autenticação. O status é derivado da fase EAP da sessão (`eap_phase`).

- Uma sessão está **Autenticando** enquanto `eap_phase` é uma fase de autenticação em progresso. As fases em progresso são `identity`, `aka_challenge`, `aka_prime_challenge`, `sim_challenge` e `resyncing`.
- Uma sessão é **Aceita** uma vez que está autenticada, ou uma vez que está ativa (`session_active`).

Uma sessão que recebe contabilidade RADIUS é promovida para ativa. A contabilidade flui apenas para um assinante que já está autenticado. A chegada da contabilidade é portanto a prova de que o assinante está ativo, então o OmniTWAG marca a sessão como Aceita.

Uma sessão que permanece em uma fase de autenticação em progresso é encerrada. Se uma sessão começou a autenticação, nunca completou e nunca recebeu contabilidade, o OmniTWAG encerra a sessão após um tempo limite. Isso impede que uma tentativa de autenticação inacabada fique como Autenticando para sempre. O tempo limite padrão é de 10 minutos.


RADIUS Access-Request starts EAP
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
identity
sim_challenge
aka_challenge
sequence out of sync
resyncing
sequence out of sync
aka_prime_challenge
EAP completes (authenticated)
RADIUS accounting arrives (session_active)
Accepted
kick / accounting stop /



Modelo de sessão

Uma sessão tem os campos abaixo. A forma é aberta, então uma sessão pode também transportar outras chaves GTP ou de cobrança. Cada campo abaixo é renderizado como uma string imprimível, um inteiro ou uma lista. Valores binários são renderizados como strings imprimíveis ou como hexadecimais. Tuplas IP são renderizadas como strings de ponto-quádruplo.

Campo	Tipo	Descrição
`identity`	String	Identidade da sessão. Este é o IMSI ou o nome de usuário. O OmniTWAG identifica a sessão por este valor.
`imsi`	String	IMSI do assinante.
`mac_address`	String	Endereço MAC do UE.
`ue_ip`	String	Endereço IP do UE alocado, como uma string de ponto-quádruplo.
`nas_ip`	String	IP de origem do AP (RADIUS `NAS-IP-Address`).
`called_station_id`	String	Identificador do ponto de acesso (RADIUS `Called-Station-Id`).
`calling_station_id`	String	MAC do cliente (RADIUS `Calling-Station-Id`).
`acct_session_id`	String	ID da sessão de contabilidade RADIUS. O OmniTWAG mescla o registro de contabilidade por esta chave.
`eap_phase`	String	Fase EAP atual. Este valor determina o status de autenticação. Veja Referência de status de autenticação.
`eap_type`	String	Método EAP em uso, por exemplo `aka` ou `aka_prime`.
`pgw_ip`	String	Endereço IP do PGW, como uma string de ponto-quádruplo.
`bearer_id`	Integer	ID do portador EPS.

Campo	Tipo	Descrição
`dns_servers`	Lista de String	Servidores DNS atribuídos ao UE.
`created_at`	Integer	Hora de criação da sessão, em milissegundos desde a época Unix.
`last_activity`	Integer	Hora da última atividade, em milissegundos desde a época Unix.
`established_at`	Integer	Hora de estabelecimento da sessão, em milissegundos desde a época Unix.
`acct_input_octets`	Integer	Bytes recebidos do UE. Mesclados da contabilidade RADIUS.
`acct_output_octets`	Integer	Bytes enviados ao UE. Mesclados da contabilidade RADIUS.
`acct_input_packets`	Integer	Pacotes recebidos do UE. Mesclados da contabilidade RADIUS.
`acct_output_packets`	Integer	Pacotes enviados ao UE. Mesclados da contabilidade RADIUS.
`acct_session_time`	Integer	Duração da sessão de contabilidade, em segundos. Mesclados da contabilidade RADIUS.
`acct_status_type`	Integer	Último status de contabilidade. `1` é Início, `2` é Parada, `3` é Interino.
`acct_last_update`	Integer	Hora da última atualização de contabilidade, em milissegundos desde a época Unix.

Os campos `acct_*` estão presentes apenas após o registro de contabilidade do assinante ter sido mesclado na sessão.

Referência de status de autenticação

O OmniTWAG mapeia o valor `eap_phase` para um status exibido. A tabela abaixo lista o mapeamento.

`eap_phase`	Status exibido	Estado de autenticação
`identity`	Identidade	Autenticando
`aka_challenge`	Desafio AKA	Autenticando
`aka_prime_challenge`	Desafio AKA'	Autenticando
`sim_challenge`	Desafio SIM	Autenticando
`resyncing`	Re-sincronizando	Autenticando
`authenticated`	Aceito	Aceito
`session_active`	Aceito	Aceito
`accepted`	Aceito	Aceito
`rejected`	Rejeitado	Rejeitado

Qualquer outro valor `eap_phase` conta como Autenticando.

API REST

A API REST do OmniTWAG serve esses endpoints sobre HTTPS na porta API configurada (`8444` por padrão). O caminho base é `/sessions`.

Método	Caminho	Ação	Descrição
GET	`/sessions`	index	Lista sessões autenticadas. A resposta é paginada e pesquisável.
GET	`/sessions/{id}`	show	Obtém uma sessão por `identity`.
DELETE	`/sessions/{id}`	delete	Remove a sessão e desconecta o assinante.
PATCH	`/sessions/{id}`	action	Realiza uma ação na sessão. A ação está no corpo JSON.
PUT	`/sessions/{id}`	action	Realiza uma ação na sessão. A ação está no corpo JSON.

O parâmetro de caminho `{id}` é a `identity` da sessão (o IMSI ou nome de usuário).

Pesquisa e paginação

O endpoint de índice aceita parâmetros de paginação e pesquisa.

Parâmetro	Tipo	Descrição
`offset`	Integer	Número de sessões a serem ignoradas antes da página.
`limit`	Integer	Número máximo de sessões na página.
`q`	String	Termo de pesquisa. O OmniTWAG compara o termo com os campos de pesquisa abaixo.

O termo `q` é comparado com esses campos de sessão.

Campo de pesquisa	Combinações
`identity`	Identidade da sessão (IMSI ou nome de usuário).
`imsi`	IMSI do assinante.
`nas_ip`	IP de origem do AP (`NAS-IP-Address`).
`called_station_id`	Identificador do ponto de acesso.
`calling_station_id`	MAC do cliente.
`mac_address`	Endereço MAC do UE.
`ue_ip`	Endereço IP do UE alocado.

Corpo da solicitação de ação

`PATCH` e `PUT` aceitam um corpo JSON com um campo.

Campo	Tipo	Obrigatório	Descrição
`action`	String	Sim	A ação a ser realizada. Deve ser `kick`, `start_accounting` ou `stop_accounting`.

Veja Ações por sessão para o que cada ação faz.

Respostas

Status	Significado
`200 OK`	A solicitação foi bem-sucedida (index, show, delete, action).
`404 Not Found`	Nenhuma sessão tem a `identity` dada.
`422 Unprocessable Entity`	A ação não é uma das `kick`, `start_accounting` ou `stop_accounting`.

Uma ação bem-sucedida retorna um corpo com um campo `status` e a `identity`. Uma exclusão retorna `{"status": "deleted", "identity": ...}`.

Ações por sessão

Uma ação executa uma operação administrativa contra uma sessão ao vivo. Envie a ação no corpo JSON de uma solicitação `PATCH` ou `PUT` para `/sessions/{id}`.

Ação	Efeito	Resultado `status`
`kick`	Envia um RADIUS CoA-Disconnect para o ponto de acesso e encerra a sessão.	`kicked`
`start_accounting`	Marca a sessão como ativa e inicia a contabilidade. Isso promove a sessão para Aceita.	`accounting_started`
`stop_accounting`	Encerra a contabilidade, como se o ponto de acesso enviasse Accounting-Stop. Isso encerra a sessão.	`accounting_stopped`

Tanto `kick` quanto `stop_accounting` encerram a sessão. Use `kick` para desconectar ativamente o assinante no ponto de acesso. Use `stop_accounting` para encerrar a sessão como a rede a encerraria.

Exemplos de JSON

Uma sessão com tráfego

Solicitação:

```
GET /sessions/001010000000001
```

Resposta:

```
{
  "identity": "001010000000001",
  "imsi": "001010000000001",
  "mac_address": "AA:BB:CC:DD:EE:FF",
  "ue_ip": "10.45.0.2",
  "nas_ip": "192.0.2.10",
  "called_station_id": "AA-BB-CC-DD-EE-FF:WiFi-Calling",
  "calling_station_id": "11-22-33-44-55-66",
  "acct_session_id": "5A2E1F0C",
  "eap_phase": "session_active",
  "eap_type": "aka_prime",
  "pgw_ip": "192.0.2.20",
  "bearer_id": 5,
  "dns_servers": ["8.8.8.8", "8.8.4.4"],
  "created_at": 1712000000000,
  "last_activity": 1712000600000,
  "established_at": 1712000004000,
  "acct_input_octets": 1048576,
  "acct_output_octets": 5242880,
  "acct_input_packets": 1200,
  "acct_output_packets": 3400,
  "acct_session_time": 600,
  "acct_status_type": 3,
  "acct_last_update": 1712000600000
}
```


Uma ação de kick

Solicitação:

```
PATCH /sessions/001010000000001
```

```
{
  "action": "kick"
}
```

Resposta (`200 OK`):

```
{
  "status": "kicked",
  "identity": "001010000000001"
}
```

Uma ação desconhecida

Solicitação:

```
{
  "action": "reboot"
}
```

Resposta (`422 Unprocessable Entity`):

```
{
  "error": "Unknown action \"reboot\" (expected
kick|start_accounting|stop_accounting)",
  "identity": "001010000000001"
}
```

Interface web

O painel de controle tem uma página de **Clientes RADIUS**. A página lista cada sessão com seu status de autenticação. A página mostra a contagem de sessões Aceitas, Autenticando, e Rejeitadas. Use a página para revisar sessões ao vivo de relance.

Solução de problemas

Uma sessão está presa em "Autenticando"

Sintomas: Uma sessão mostra o status Autenticando. O status não muda para Aceito. A sessão permanece em uma fase EAP como `identity`, `aka_challenge`,

`aka_prime_challenge`, `sim_challenge` ou `resyncing`.

Possíveis causas:

- O assinante iniciou a autenticação EAP, mas nunca a completou.
- O ponto de acesso parou de enviar respostas EAP.
- O assinante falhou na autenticação e não tentou novamente.

Como é resolvido:

1. O OmniTWAG ativa um tempo limite quando a sessão entra em uma fase de autenticação em progresso.
2. Se a sessão nunca completar a autenticação e nunca receber contabilidade, o OmniTWAG encerra a sessão quando o tempo limite expira. O tempo limite padrão é de 10 minutos.
3. Uma sessão que completa a autenticação move-se para Aceita. Uma sessão que recebe contabilidade RADIUS é promovida para ativa e também se move para Aceita.

Para remover uma sessão presa de uma vez, envie uma ação `kick` ou uma solicitação `DELETE` para `/sessions/{id}`.

Uma sessão não mostra tráfego

Sintomas: Uma sessão está Aceita, mas os campos `acct_*` estão ausentes ou zerados.

Possíveis causas:

- Nenhum registro de contabilidade RADIUS foi recebido para o assinante ainda.
- O registro de contabilidade tem um `Acct-Session-Id` diferente do da sessão.

Resolução:

1. Confirme que o ponto de acesso envia contabilidade RADIUS para o OmniTWAG.

2. Confirme que o `Acct-Session-Id` da contabilidade corresponde ao `acct_session_id` da sessão. O OmniTWAG mescla o tráfego por esta chave.
3. Aguarde a próxima atualização interina de contabilidade, depois leia a sessão novamente.

Visão geral

Plano do Usuário e Caminho de Dados

Visão Geral

OmniTWAG separa o plano de controle do plano do usuário. O plano de controle executa a autenticação e a sinalização da sessão. O plano do usuário transporta o tráfego real do UE. Esta página descreve o plano do usuário. Ela explica como o OmniTWAG move pacotes do UE, como constrói e desmonta o túnel GTP-U e como encaminha pacotes de downlink de volta para o UE.

Para a sinalização de sessão GTP-C que configura o caminho de dados encapsulado, veja S2a / GTP-C. Para a visão completa do sistema, veja Arquitetura.

Índice

- Plano de Controle e Plano do Usuário
- Modos de Caminho de Dados
- Caminho de Dados GRE
- Encapsulamento GTP-U
- Encaminhamento de Downlink
- Portadoras e Modelos de Fluxo de Tráfego
- Tratamento de Erros
- MTU e Fragmentação
- Configuração
- Observabilidade

Plano de Controle e Plano do Usuário

O plano de controle e o plano do usuário funcionam como processos separados.

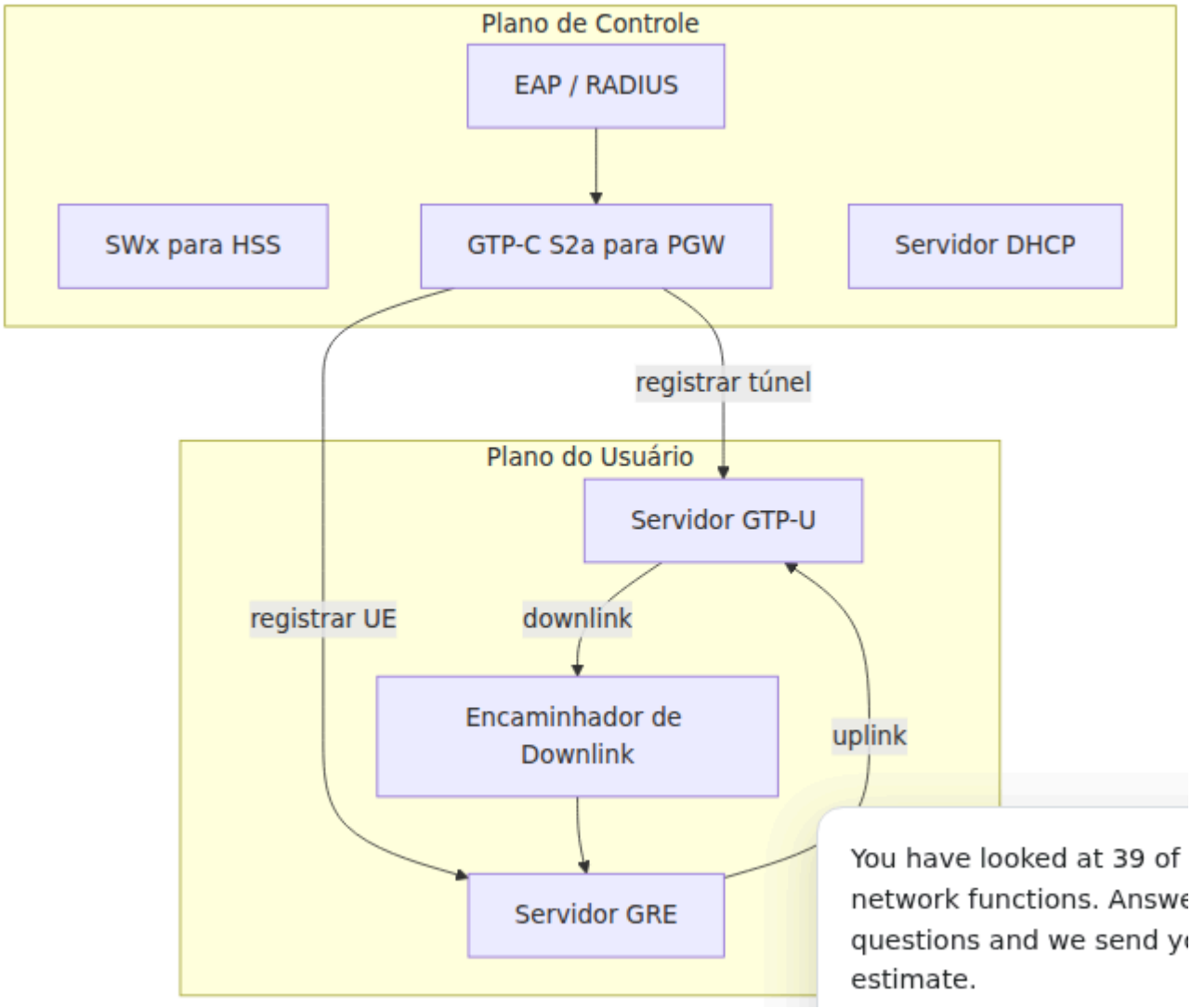
O **plano de controle** realiza o seguinte trabalho:

- Autenticação EAP sobre RADIUS com o ponto de acesso.
- Busca de vetor de autenticação do HSS via Diameter SWx.
- Sinalização de sessão GTP-C para o PGW na interface S2a. Isso cria e exclui a conexão PDN e as portadoras. Veja S2a / GTP-C.
- Entrega de endereço DHCP para o UE. Veja Gerenciamento de Endereço IP.

O **plano do usuário** realiza o seguinte trabalho:

- Encapsulamento GRE para o ponto de acesso.
- Encapsulamento GTP-U para o PGW na interface S2a-U, conforme 3GPP TS 29.281 (modo encapsulado apenas).
- Encaminhamento de pacotes de downlink em direção ao UE.
- Classificação de pacotes de uplink em portadoras.

O plano de controle configura o estado do plano do usuário. Após a autenticação, o processo de sessão cria a sessão GTP-C (em modo encapsulado), registra o túnel GTP-U e registra o UE com o servidor GRE. Então, o plano do usuário transporta o tráfego.



Modos de Caminho de Dados

OmniTWAG resolve um modo de plano de dados por ponto de acesso (AP) a partir da credencial RADIUS que corresponde ao AP. O valor `breakout_type` na credencial seleciona um dos três modos. Veja Credenciais RADIUS para as configurações `breakout_type`, `address_source` e `ip_pool_id`.

Modo (`breakout_type`)	IP do UE atribuído por	TWAG transporta tráfego	Caminho
`tunneled`	PGW	Sim	GRE do AP para TWAG, então GTP-U no S2a para o PGW.
`nswo`	TWAG, de um pool de IP	Sim	GRE do AP para TWAG, então saída local para a internet (NAT) no TWAG.
`local`	Rede local no AP	Não	O AP possui o plano de dados. O TWAG não transporta pacotes e não atribui IP.

O caminho de dados GRE se aplica tanto ao `tunneled` quanto ao `nswo` breakout. Em ambos os modos, o AP envia quadros do UE para o TWAG em GRE, e o TWAG envia pacotes de downlink de volta para o AP em GRE. A diferença está no uplink após GRE:

- **`tunneled`**: o TWAG encapsula o pacote de uplink em GTP-U e o envia para o PGW. O PGW atribui o IP do UE e aplica políticas e cobrança.
- **`nswo`**: o TWAG quebra o uplink localmente para a internet com uma regra de mascaramento NAT. O TWAG atribui o IP do UE de um pool de IP. Não há PGW ou sessão GTP-C. O Gy do lado do TWAG Cobrança Online se aplica.

No modo `local`, o TWAG não atribui um IP nem transporta tráfego, então nem o caminho GRE nem o GTP-U são usados para esse AP.

Modo Encapsulado (GTP-U para PGW)

No modo encapsulado, o OmniTWAG encapsula o tráfego do UE para o PGW. O PGW aloca o endereço IP do UE, aplica a política e a cobrança, e fornece acesso

à internet. Esta é a interface S2a conforme 3GPP TS 23.402 Seção 16.

O caminho de uplink é:

```
UE --WiFi--> AP --GRE--> TWAG --GTP-U S2a--> PGW --> Internet
```

O caminho de downlink é o inverso:

```
Internet --> PGW --GTP-U S2a--> TWAG --GRE--> AP --WiFi--> UE
```

Para usar este modo, defina `gtp.enabled` como `true` e `gre.enabled` como `true`, e defina a credencial do AP `breakout_type` como `tunneled`. Veja Configuração.

Saída Local (NSWO)

A Saída de WLAN Não-Suave (NSWO) é a saída local. O tráfego do UE sai diretamente do TWAG para a internet. O tráfego não vai para o PGW. O TWAG atribui o IP do UE e aplica mascaramento NAT. Isso é conforme 3GPP TS 23.402 Seção 16.2.

OmniTWAG tem duas maneiras de alcançar NSWO:

1. **Saída por AP (preferida)**: defina a credencial do AP `breakout_type` como `nswo`. O endereço do UE vem do pool de IP nomeado por `ip_pool_id`. O `address_source` da credencial seleciona como:
    - `pool` pré-aloca um endereço fixo do pool e registra o UE para downlink GRE de uma vez.
    - `dhcp` vincula o MAC do UE ao pool e permite que o servidor DHCP do TWAG distribua um endereço no DHCP DISCOVER. O downlink GRE é registrado quando o endereço é conhecido.
2. **Manipulador NSWO Global**: o `Twag.NSWO.Handler` fornece saída local para cada sessão quando o GTP não é usado. Ele é selecionado com `nswo.enabled: true` (e se aplica quando `gtp.enabled: false`). O manipulador gerencia um único pool de IP local e configura uma regra de

mascaramento NAT `iptables`. Na criação da sessão, o TWAG aloca um endereço do pool local, entrega-o ao UE via DHCP e registra o UE para downlink GRE, sem PGW / sessão GTP-C. A alocação é liberada na desmontagem da sessão.

Para ambos os caminhos, o uplink é mascarado para a internet e o downlink retorna pelo caminho GRE. Veja Gerenciamento de Endereço IP e Pools de IP para o endereçamento.

O manipulador NSWO global lê esses parâmetros do mapa `nswo`:

Parâmetro	Tipo	Padrão	Descrição
`enabled`	Booleano	`false`	Iniciar o manipulador NSWO global e rotear sessões via saída local.
`local_pool`	String (CIDR)	`10.100.0.0/16`	Pool do qual o endereço do UE é alocado.
`nat_interface`	String	`eth0`	Interface de saída para a regra de mascaramento NAT.
`dns_servers`	Lista	`["8.8.8.8", "8.8.4.4"]`	Servidores DNS anunciados ao UE. A saída local não tem PCO do PGW para transportá-los.

Caminho de Dados GRE

O servidor GRE (`Twag.Gre.Server`) é o ponto final do plano do usuário em direção ao ponto de acesso. O ponto de acesso envia os quadros do UE para o TWAG em GRE, conforme RFC 2784 e RFC 2890. O servidor GRE também envia os pacotes de downlink e as respostas DHCP de volta para o ponto de acesso.

GRE é o caminho voltado para o AP tanto para o breakout `tunneled` quanto para o `nswo`.

O servidor GRE usa um socket bruto para o protocolo IP 47 (GRE), aberto com `:socket.open/3`. O host deve conceder a capacidade `CAP_NET_RAW`. Se o socket não abrir, o servidor registra um erro e tenta novamente a cada 10 segundos. GRE está desativado por padrão. Para ativá-lo, defina `gre.enabled` como `true`.

Classificação GRE de Entrada

O servidor GRE lê o cabeçalho IP externo, o remove e decodifica o cabeçalho GRE. O servidor suporta dois tipos de carga útil GRE:

Protocolo GRE	Significado	Manipulação
`0x6558`	Ponte Ethernet Transparente	Extrai o quadro Ethernet interno, depois classifica por ethertype.
`0x0800`	IPv4	Trata como um pacote IPv4 de uplink.

Para um quadro de Ponte Ethernet Transparente, o servidor lê o ethertype interno:

Ethertype	Significado	Manipulação
`0x0800`	IPv4	Se o pacote for DHCP, passe-o para o servidor DHCP. Se não, encaminhe-o para o caminho de uplink.
`0x0806`	ARP	Manipule localmente. Não encaminhe.
`0x86DD`	IPv6	Encaminhe para o caminho de uplink.

O servidor mantém um mapa do endereço MAC do UE para o IP do ponto de acesso. O servidor aprende esse mapa a partir do MAC de origem de cada quadro de entrada. Esse mapa permite que o servidor roteie os quadros de downlink de volta para o ponto de acesso correto.

Manipulação de Uplink

Para um pacote IP de uplink, o servidor GRE lê o IP de origem. Este é o IP do UE. O servidor procura o IP do UE na tabela de UEs registrados para encontrar a identidade da sessão. Ele encontra o processo de sessão no registro e lê o TEID local da sessão. Em seguida, ele envia o pacote de forma assíncrona para o servidor GTP-U com aquele TEID local (em modo encapsulado). O envio assíncrono evita um deadlock entre os processos GRE e GTP-U.

Se não houver sessão para o IP do UE, ou se a sessão ainda não tiver túnel GTP-U, o servidor descarta o pacote. Um pacote IP pequeno (menos de 20 bytes) também é descartado.

> Nota: Para a saída `nswo`, o pool de IP local e o mascaramento NAT transportam o uplink para a internet. O servidor GRE registra o UE para roteamento de downlink da mesma forma que no modo encapsulado.

Registro do UE para Downlink

Quando a sessão se torna ativa, o processo de sessão registra o UE com o servidor GRE. O registro contém o IP do UE, o MAC do UE, o IP do ponto de acesso e a identidade da sessão. O servidor GRE usa esses dados para roteamento dos pacotes de downlink. Na desmontagem da sessão, o processo de sessão desregistra o UE.

Encapsulamento GTP-U

O servidor GTP-U (`Twag.Gtp.GtpU`) é o ponto final do plano do usuário em direção ao PGW em modo encapsulado. Ele executa um único socket UDP na porta `2152` e mantém uma tabela de túneis. O servidor implementa GTPv1-U conforme 3GPP TS 29.281.

Manipulação de TEID

Cada túnel tem um TEID local e um TEID remoto. O TEID remoto é o TEID do PGW. O TWAG usa o TEID remoto no cabeçalho dos pacotes que envia para o PGW. O PGW usa o TEID local no cabeçalho dos pacotes que envia para o TWAG. O TWAG procura o TEID local na tabela de túneis para encontrar o túnel.

Quando o processo de sessão recebe a Resposta de Criação de Sessão do PGW, ele registra o túnel. O servidor GTP-U gera o TEID local. O TWAG usa um valor aleatório de 32 bits para o TEID local e nunca usa zero. Isso segue o requisito de TEID não previsível em 3GPP TS 29.274.

Um túnel registrado contém os seguintes campos:

Campo	Descrição
`local_teid`	O TEID do lado do TWAG. O PGW envia pacotes de downlink com este valor.
`remote_teid`	O TEID do lado do PGW. O TWAG envia pacotes de uplink com este valor.
`remote_ip`	O endereço IP GTP-U do PGW.
`remote_port`	A porta UDP GTP-U do PGW. Padrão `2152`.
`session_id`	A identidade da sessão para o túnel.

Encapsulamento de Uplink

Para cada pacote de UE de uplink, o servidor GTP-U constrói um cabeçalho GTPv1-U G-PDU e coloca o cabeçalho na frente do pacote IP interno. O cabeçalho carrega a versão GTP (1), o tipo de protocolo (GTP), o tipo de mensagem (G-PDU, 255), o comprimento da carga útil, e o TEID remoto. O servidor envia o pacote para o IP e porta do PGW através do socket UDP. O servidor GRE chama o servidor GTP-U no caminho de envio assíncrono, o que evita um deadlock entre os dois processos.

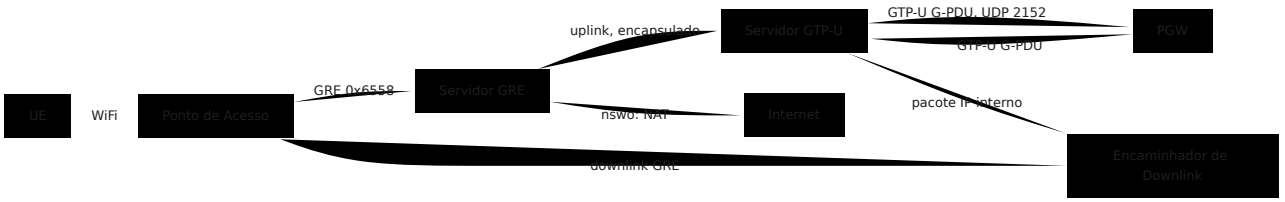
Desencapsulamento de Downlink

Para cada pacote que chega no socket GTP-U, o servidor lê o cabeçalho GTP-U. O servidor age de acordo com o tipo de mensagem:

Mensagem	Tipo	Ação
G-PDU	255	Procura o túnel pelo TEID local, depois encaminha o pacote IP interno para o encaminhador de downlink.
Echo Request	1	Responde com um Echo Response.
Echo Response	2	Aceita e ignora.
Error Indication	26	Aceita e ignora.
End Marker	254	Aceita e ignora.

Para um G-PDU, o servidor remove o cabeçalho GTP-U e quaisquer campos de cabeçalho opcionais (número de sequência, número N-PDU e tipo de cabeçalho de extensão seguinte). Em seguida, o servidor passa o pacote IP interno para `Twag.Downlink`. Veja Encaminhamento de Downlink.

Se um G-PDU chegar para um TEID que não tem túnel, o servidor descarta o pacote e envia uma Indicação de Erro GTP-U de volta para a origem. Veja Tratamento de Erros.



Encaminhamento de Downlink

O encaminhador de downlink (`Twag.Downlink`) recebe os pacotes IP internos do desencapsulamento GTP-U. O encaminhador envia cada pacote em direção ao UE. O encaminhador tem dois caminhos e prefere GRE.

Caminho GRE (Preferido)

Se o servidor GRE estiver em execução, o encaminhador envia o pacote através do GRE. O encaminhador lê o IP de destino do pacote interno. Este é o IP do UE. O encaminhador chama o servidor GRE. O servidor GRE procura o IP do UE na tabela de UEs registrados, envolve o pacote em um quadro GRE de Ponte Ethernet Transparente e o envia para o IP do ponto de acesso. O ponto de acesso entrega o quadro ao UE via WiFi.

Para o quadro Ethernet de downlink, o servidor GRE usa o MAC do UE como o MAC de destino. O servidor usa um MAC administrado localmente fixo (`02:00:00:00:00:01`) como o MAC de origem.

Caminho de Socket Bruto (Fallback)

Se o servidor GRE não estiver em execução, ou se o envio GRE falhar, o encaminhador usa um socket bruto. O encaminhador injeta o pacote IP bruto. O kernel então roteia o pacote para o UE a partir da tabela de roteamento local.

O encaminhador abre dois sockets brutos: um para IPv4 e um para IPv6. O encaminhador define a opção `IP_HDRINCL` (e `IPV6_HDRINCL` para IPv6). Esta opção significa que o encaminhador fornece o cabeçalho IP completo. Em alguns kernels, o socket ainda funciona sem esta opção, então uma falha ao definir a opção não é fatal.

O host deve conceder a capacidade `CAP_NET_RAW` para os sockets brutos. Se um socket não abrir, o encaminhador registra um aviso e desativa essa família de endereços. Se nem GRE nem um socket bruto estiverem disponíveis, o encaminhador conta e descarta os pacotes, e registra um aviso uma vez.

Seleção de Caminho

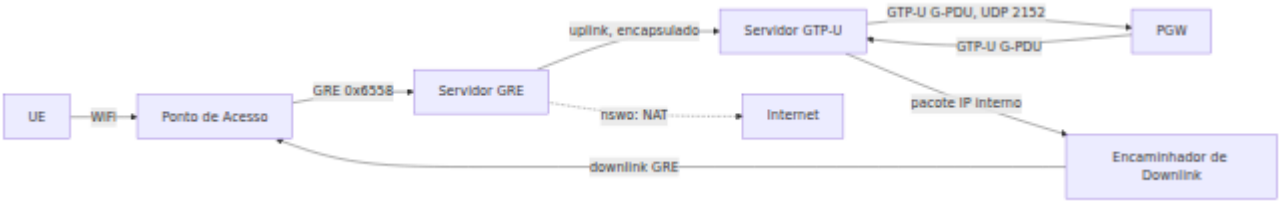
O parâmetro `prefer_gre` define a ordem de seleção do caminho.

- `prefer_gre: true` (padrão): o encaminhador tenta o túnel GRE primeiro e recua para o socket bruto se o GRE estiver indisponível ou se o envio GRE falhar.
- `prefer_gre: false`: o encaminhador tenta o socket bruto para a família de endereços do pacote primeiro e usa GRE apenas quando esse socket bruto não estiver disponível.

Verificações de Pacote

O encaminhador descarta um pacote nos seguintes casos:

- O encaminhador está desativado pela configuração.
- Nenhum socket está disponível, e o GRE não está disponível.
- O pacote é pequeno (menos de 20 bytes).
- O encaminhador não consegue ler o IP de destino.



Portadoras e Modelos de Fluxo de Tráfego

O gerenciador de portadoras (`Twag.Bearer.Manager`) mantém as portadoras para uma sessão. Cada portadora mapeia para uma Identidade de Portadora EPS (EBI). Uma portadora contém os TEIDs GTP-U, os filtros de pacotes do Modelo de Fluxo de Tráfego (TFT) e a classe QoS (QCI). Uma portadora é a portadora padrão. Isso segue 3GPP TS 23.402 Seção 16.1.6.1.

Portadoras Padrão e Dedicadas

Tipo de portadora	Filtros TFT	Propósito
Padrão	Nenhum	Transporta todo o tráfego que nenhuma portadora dedicada corresponde.
Dedicada	Uma ou mais	Transporta o tráfego que corresponde aos seus filtros de pacotes TFT.

Se a portadora padrão for removida, o gerenciador de portadoras sinaliza uma desmontagem completa da sessão. A remoção de uma portadora dedicada remove apenas essa portadora.

Filtros de Pacotes TFT

Um TFT é um Modelo de Fluxo de Tráfego conforme 3GPP TS 24.008 Seção 10.5.6.12. Um TFT contém uma operação e uma lista de filtros de pacotes. Cada filtro de pacote contém uma direção, uma precedência e uma lista de componentes. O analisador TFT suporta esses tipos de componentes:

Componente	Correspondência
Endereço remoto IPv4 e máscara	O endereço de destino do pacote.
Endereço local IPv4 e máscara	O endereço de origem do pacote.
Endereço remoto/local IPv6 e prefixo	O endereço IPv6.
Identificador de protocolo	O número do protocolo IP.
Porta local única / intervalo de portas locais	A porta de origem do UE.
Porta remota única / intervalo de portas remotas	A porta de destino.
Tipo de serviço / classe de tráfego	O byte TOS, com uma máscara.
Índice de parâmetro de segurança	Não correspondido no uplink TWAN.
Rótulo de fluxo	Não correspondido (apenas IPv6).

O analisador suporta as operações TFT: criar, excluir, adicionar filtros, substituir filtros, excluir filtros e nenhuma operação. O gerenciador de portadoras aplica essas operações quando o PGW envia uma Solicitação de Atualização de Portadora.

Classificação de Uplink

O classificador de uplink vincula cada pacote de uplink a uma portadora. As regras são:

1. Ler o IP de origem, o IP de destino, o protocolo, as portas e o TOS do pacote.
2. Testar o pacote contra os filtros TFT das portadoras dedicadas.
3. Um filtro corresponde apenas se todos os seus componentes corresponderem. Esta é a lógica AND.
4. Se mais de um filtro corresponder, selecione o filtro com o menor número de precedência. Um número de precedência menor é uma prioridade maior.
5. Se nenhuma portadora dedicada corresponder, selecione a portadora padrão.

Um filtro vazio (um filtro sem componentes) corresponde a todos os pacotes. Este é um curinga.

> **Nota:** O gerenciador de portadoras e o classificador TFT estão presentes para suporte a portadoras dedicadas. O caminho GRE de uplink encaminha o tráfego no TEID local da sessão. Uma divisão completa de uplink multi-portadora depende do PGW para criar portadoras dedicadas para a sessão, que o TWAG rejeita sob TSCM (veja S2a / GTP-C).

Tratamento de Erros

Echo GTP-U

O PGW e o TWAG usam a Solicitação de Echo GTP-U e a Resposta de Echo para testar o caminho do túnel. Quando o servidor GTP-U recebe uma Solicitação de Echo, ele responde com uma Resposta de Echo. O processo GTP-C S2a executa seu próprio echo quando `gtp.echo_enabled` é `true`. Veja S2a / GTP-C.

Indicação de Erro GTP-U

Se um G-PDU chegar para um TEID que não tem túnel, o servidor GTP-U envia uma Indicação de Erro GTP-U de volta para a origem. A Indicação de Erro carrega o TEID desconhecido e o endereço do par GTP-U do TWAG. Isso informa ao PGW para parar de enviar tráfego no túnel morto. O servidor também descarta o pacote e aumenta a contagem de erros.

Marcador Final GTP-U

O servidor GTP-U pode enviar um Marcador Final em um túnel antes de desmontar o túnel. Isso é para uma transferência. O Marcador Final informa ao par que não mais pacotes seguirão no caminho antigo.

Erros de Decodificação

Se o servidor GTP-U não conseguir decodificar um pacote, ele registra um aviso e aumenta a contagem de erros. Se o servidor GRE não conseguir decodificar um pacote, ou receber um protocolo ou ethertype desconhecido, ele conta o evento e descarta o pacote.

MTU e Fragmentação

O servidor GRE constrói o cabeçalho IPv4 externo para cada pacote GRE de downlink. O número do protocolo GRE é 47.

A encapsulação GTP-U e GRE adiciona bytes de cabeçalho ao pacote do UE. Para evitar fragmentação, o ponto de acesso deve reduzir o MSS TCP e o MTU do túnel. A configuração de referência do ponto de acesso Cisco define `mss 1410` e `mtu 1500` no túnel GRE. Veja Arquitetura para a configuração do ponto de acesso.

Configuração

Os processos do plano do usuário leem a configuração da aplicação `:omnitwag`. Os blocos relevantes são `gtp`, `gre` e `downlink`. O modo de saída do AP vem da credencial RADIUS, não destes blocos. Veja Credenciais RADIUS.

```
config :omnitwag,
  # GTP-U e GTP-C em direção ao PGW (S2a) - modo encapsulado
  gtp: %{
    enabled: false,
    pgw_ip: "10.5.198.1",
    pgw_port: 2123,
    local_ip: "10.5.198.200",
    gtpu_port: 2152,
    echo_enabled: true
  },

  # GRE em direção ao ponto de acesso
  gre: %{
    enabled: false,
    local_ip: "10.5.198.200"
  },

  # Encaminhamento de downlink
  downlink: %{
    enabled: true,
    ip_hdr_incl: true,
    prefer_gre: true
  }
```

Parâmetros GTP

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Iniciar os servidores GTP-C e GTP-U. Defina `true` para modo encapsulado
`pgw_ip`	String	Não	`"10.5.198.1"`	O endereço IP do PGW para GTP-C e GTP-U.
`pgw_port`	Inteiro	Não	`2123`	A porta UDP GTP-C do PGW.
`local_ip`	String	Não	Detectado automaticamente	O IP local do TWAG para GTP. Usado como origem para pacotes GTP-U e para o endereço do par da Indicação de Erro.
`gtpu_port`	Inteiro	Não	`2152`	A porta UDP local para o servidor GTP-U. Esta é a porta

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
				padrão GTP-U.
`echo_enabled`	Booleano	Não	`true`	Executar o echo GTP-C em direção ao PGW. Vej S2a / GTP-C

Parâmetros GRE

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Iniciar o servidor GRE. Defina `true` para receber quadros do UE do ponto de acesso via GRE. Necessário para ambos os breakouts `tunneled` e `nswo`.
`local_ip`	String	Não	`"0.0.0.0"`	O IP local para vincular o socket bruto GRE. Este é o IP de destino que o ponto de acesso envia o tráfego GRE.

Parâmetros de Downlink

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`enabled`	Booleano	Não	`true`	Iniciar o encaminhador de downlink. Quando `false`, o encaminhador descarta todos os pacotes de downlink.
`ip_hdr_incl`	Booleano	Não	`true`	Defina as opções `IP_HDRINCL` e `IPV6_HDRINCL` nos sockets brutos para que o encaminhador forneça o cabeçalho IP completo. Quando `false`, a opção não é definida. Deixe `true` a menos que o kernel rejeite `IP_HDRINCL`.
`prefer_gre`	Booleano	Não	`true`	Ordem de seleção de caminho. Quando `true`, o encaminhador tenta GRE primeiro e recua para o socket bruto. Quando `false`, ele tenta o socket bruto primeiro e usa GRE apenas quando o

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
				socket bruto não estiver disponível.

Capacidade do Host

O servidor GRE e o caminho de socket bruto de downlink usam sockets brutos. O host deve conceder a capacidade `CAP_NET_RAW`. Sem essa capacidade, os sockets brutos não abrem. O servidor GRE tenta o socket a cada 10 segundos. O encaminhador de downlink recua ou descarta pacotes, conforme descrito em Encaminhamento de Downlink.

Observabilidade

Os processos do plano do usuário mantêm estatísticas internas. Cada processo relata seu próprio status, cobrindo o ponto final do túnel GTP-U, o servidor GRE e o encaminhador de downlink. Para as métricas do Prometheus que o OmniTWAG exporta, veja a Referência de Métricas.

Status GTP-U

O status GTP-U relata `socket_open`, `local_ip`, `tunnel_count` e um mapa aninhado `stats`:

Estatística	Descrição
`packets_in`	A contagem de pacotes recebidos no socket GTP-U.
`packets_out`	A contagem de pacotes enviados no socket GTP-U.
`bytes_in`	O total de bytes recebidos.
`bytes_out`	O total de bytes enviados.
`errors`	A contagem de erros de decodificação e eventos de túnel desconhecido.
`tunnel_count`	O número de túneis registrados (campo de nível superior).

Status GRE

O status GRE relata `running`, `socket_open`, `local_ip`, `ap_tunnels`, `registered_ues` e um mapa aninhado `stats`:

Estatística	Descrição
`packets_in`	A contagem de pacotes GRE recebidos.
`packets_out`	A contagem de pacotes GRE enviados.
`bytes_in` / `bytes_out`	O total de bytes recebidos e enviados.
`dhcp_in`	A contagem de pacotes DHCP recebidos do ponto de acesso.
`uplink_forwarded`	A contagem de pacotes de uplink encaminhados para GTP-U.
`downlink_forwarded`	A contagem de pacotes de downlink enviados para o ponto de acesso.
`unknown_protocol`	A contagem de pacotes com um protocolo ou ethertype desconhecido.
`errors`	A contagem de erros de decodificação e envio.
`registered_ues`	O número de UEs registrados (campo de nível superior).

Estatísticas de Downlink

Estatística	Descrição
`forwarded`	A contagem de pacotes IPv4 encaminhados (GRE e socket bruto IPv4).
`forwarded_v6`	A contagem de pacotes IPv6 encaminhados.
`dropped`	A contagem de pacotes descartados.
`errors`	A contagem de erros de envio.
`bytes`	O total de bytes encaminhados.
`enabled`	O estado de ativação do encaminhador.
`socket_open`	O estado do socket bruto IPv4.

Visão Geral