

Media Gateway e SIP Proxy

Pesquisador: Müller Fernandes da Silva

Media Gateway

Media Gateway é um hardware ou serviço que faz o mapeamento e/ou transcoding de arquivos para serem utilizados em redes diferentes quando estas precisam se comunicarem (PSTN,SS7,NGN,PBX). Isto significa que mesmo se o usuário final usar recursos, hardware e aplicações diferentes do usuário que originou a chamada, será possível fechar uma comunicação entre eles.

Exemplo

Um MG pode pegar os dados vindo de uma PSTN, empacotá-los em streams de mídia caso ainda não estejam e entregar os pacotes para redes baseadas em pacotes. O caminho inverso também é verdadeiro.

MGWs não se resumem apenas como tradutores de arquivos. Também trabalham como pontes para o tráfego dos pacotes, como interfaces de reconhecimento de voz e fontes de áudio.

Detalhamento:

- **Media Gateway Unit (MG-unit):** Entidade física que executa as funções de um media gateway.
- **Media Gateway Controller (MGC):** Controla o funcionamento de um MG.
- **Transcoding:** Forma direta de conversão digital-para-digital de um arquivo. De forma mais técnica, conversão entre diferentes encoders.

SIP Proxy

Session Initiation Protocol (SIP) é um protocolo de controle da camada de aplicação usado para criar, modificar e terminar sessões com um ou mais participantes. Essas sessões podem ser de chamadas de telefone via internet, distribuição e conferências multimídia.

Um SIP proxy ou Servidor Proxy SIP é um elemento da rede que roteia uma chamada SIP até o usuário de destino.

Um pedido pode passar por vários SIP Proxies antes de chegar ao destinatário, onde cada SIP Proxy modificará o pedido de forma a tomar decisões de rotas a serem seguidas. A resposta do pedido seguirá exatamente a mesma rota do pedido, com sentido reverso.

Detalhamento

Um SIP Proxy pode operar em dois estados:

- **SIP Proxy stateless:** funciona apenas como um encaminhador de pedidos. Ele simplesmente envia os pedidos, descartando as informações da mensagem uma vez que tenham sido enviadas.
- **SIP Proxy statefull:** retém as informações de um pedido após o envio do mesmo, como também as informações de processamento do pedido. Ele usa essas

informações para afetar processos futuros e decidir se é necessário "to fork" o pedido, roteando-o para múltiplos destinos. Qualquer pedido feito para múltiplos locais deve ser operado em modo statefull.

Sequência de Funcionamento

1. **Validar o pedido**
 - Nesta etapa checa se a URI está bem formatada, se a sintaxe do pedido está correta, entre outras coisas.
2. **processar informações de roteamento**
 - checa se a URI especificada possui um endereço ou domínio ao qual é permitido uma resposta.
3. **determinar o destino para o pedido**
 - nesta etapa calcula os destinos do pedido, este depende do tipo do pedido, ou então é obtido de um abstract location service. Cada destino definido é representado por uma URI.
4. **encaminhar o pedido para cada destino**
5. **processar todas as respostas**

Exemplo:

This scenario is the basic SIP trapezoid, U1 -> P1 -> P2 -> U2, with both proxies record-routing. Here is the flow.

U1 sends:

```
INVITE sip:callee@domain.com SIP/2.0
Contact: sip:caller@u1.example.com
```

to P1. P1 is an outbound proxy. P1 is not responsible for domain.com, so it looks it up in DNS and sends it there. It also adds a Record-Route header field value:

```
INVITE sip:callee@domain.com SIP/2.0
Contact: sip:caller@u1.example.com
Record-Route: <sip:p1.example.com;lr>
```

P2 gets this. It is responsible for domain.com so it runs a location service and rewrites the Request-URI. It also adds a Record-Route header field value. There is no Route header field, so it resolves the new Request-URI to determine where to send the request:

```
INVITE sip:callee@u2.domain.com SIP/2.0
Contact: sip:caller@u1.example.com
Record-Route: <sip:p2.domain.com;lr>
Record-Route: <sip:p1.example.com;lr>
```

The callee at u2.domain.com gets this and responds with a 200 OK:

```
SIP/2.0 200 OK
Contact: sip:callee@u2.domain.com
Record-Route: <sip:p2.domain.com;lr>
Record-Route: <sip:p1.example.com;lr>
```

The callee at u2 also sets its dialog state's remote target URI to `sip:caller@u1.example.com` and its route set to:

```
(<sip:p2.domain.com;lr>,<sip:p1.example.com;lr>)
```

This is forwarded by P2 to P1 to U1 as normal. Now, U1 sets its dialog state's remote target URI to `sip:callee@u2.domain.com` and its route set to:

```
(<sip:p1.example.com;lr>,<sip:p2.domain.com;lr>)
```

Since all the route set elements contain the `lr` parameter, U1 constructs the following `BYE` request:

```
BYE sip:callee@u2.domain.com SIP/2.0
Route: <sip:p1.example.com;lr>,<sip:p2.domain.com;lr>
```

Fontes

<http://www.ietf.org/rfc/rfc3261.txt>
<http://datatracker.ietf.org/doc/rfc2805/>