

**Especialização em
Telecomunicações**

**TP 318 – Introdução às
Redes Multimídia**

© Antônio M. Alberti

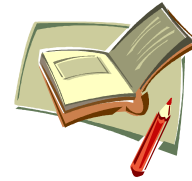
Voz sobre IP

**Prof. Antônio M. Alberti
Prof. José Marcos C. Brito**

© Antônio M. Alberti

Tópicos

- ✓ Introdução
- ✓ RTP
- ✓ RSTP
- ✓ RTCP
- ✓ Arquitetura SIP
- ✓ Arquitetura OPT
- ✓ Referências Bibliográficas



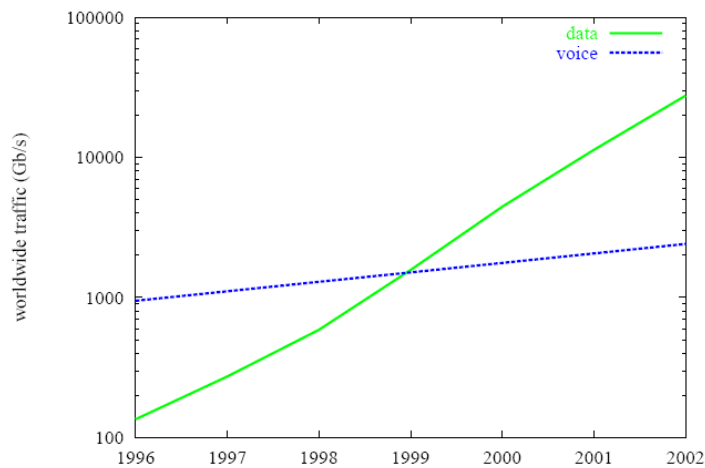
Introdução

- ✓ Motivação
- ✓ Pilha de Protocolos

Motivação

- ✓ No início da **década de 90**, boa parte do tráfego de dados **acessava** a Internet via **rede telefônica**.
- ✓ A entrada do tráfego de dados causou um grande impacto na rede telefônica, uma vez que as **chamadas telefônicas** têm uma **duração média** de **3 minutos** enquanto as **chamadas de Internet** têm uma duração de **30 minutos**.
- ✓ Além disso, o tráfego de dados possui uma ordem de grandeza em termos de taxa muito maior que o de voz.
- ✓ Assim, em 1999 o volume de **tráfego de dados** se igualou ao volume de **tráfego de voz**.

Motivação



Fonte: Schulzrinne

Motivação

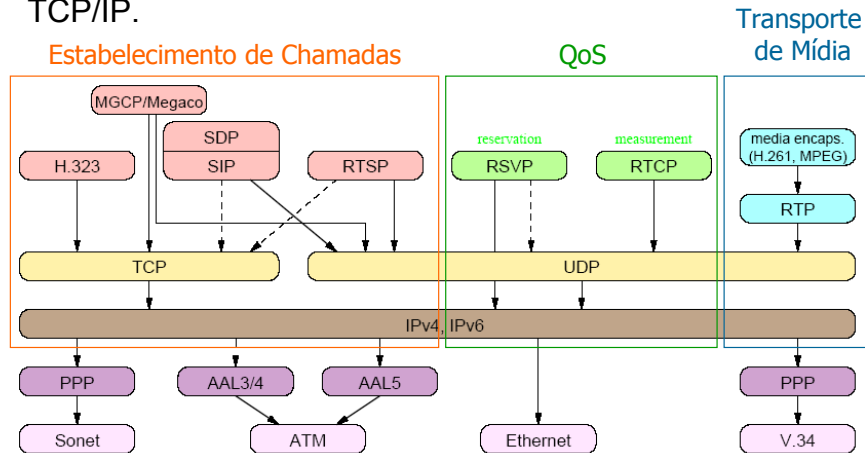
- ✓ Em 2002, o volume de **tráfego de dados** tornou-se **10 vezes maior** que o volume do **tráfego de voz**, e a tendência é que continue a crescer exponencialmente, enquanto o tráfego de voz permanece quase do mesmo tamanho, crescendo a uma taxa de 5% ao ano.
- ✓ Para atender a **demanda crescente** de Internet, as operadoras implantaram soluções como o **ADSL**, que aproveitam as redes metálicas existentes e provêm acesso a taxa mais altas.

Motivação

- ✓ Entretanto, uma infra-estrutura de **rede de dados** de longa distância teve que ser **implementada** para este fim.
- ✓ Esta infra-estrutura tipicamente utilizou as tecnologias **TCP/IP, ATM e SDH**.
- ✓ Como consequência, muitas operadoras ficaram interessadas em transportar voz sobre suas redes de dados.
- ✓ Assim surgiu a **telefonia IP**, também chamada de **telefonia via Internet**.
- ✓ A telefonia IP é uma tecnologia que torna possível manter uma **conversa de voz** através da **Internet TCP/IP**.

Pilha de Protocolos

- ✓ A figura apresenta a **pilha de protocolos** multimídia do TCP/IP.



Fonte: Schulzrinne

© Antonio M. Alberti

Pilha de Protocolos

- ✓ Primeiramente, estudaremos os protocolos que tem a ver com o **transporte de mídia e seu controle** após uma chamada já ter sido estabelecida:
 - RTP – *Real-time Transport Protocol*
 - RSTP – *Real-time Streaming Protocol*
 - RTCP – *RTP Control Protocol*
- ✓ Posteriormente, estudaremos as arquiteturas de protocolos utilizados no plano de controle:
 1. **Arquitetura SIP** – *Session Initiation Protocol*.
 2. **Arquitetura H.323**.
 3. **Arquitetura OPT** – *Open Packet Telephony*.

© Antônio M. Alberti

RTP

- ✓ O RTP é um protocolo de transporte interativo de mídia em tempo real.
- ✓ Foi definido na RFC 3550 de 2003.
- ✓ Define um formato de pacotes padrão para o transporte de áudio e vídeo através da Internet.
- ✓ Permite transmissões *unicast* e *multicast*.
- ✓ Usualmente utilizado em conjunto com UDP, mas pode também ser utilizado com o TCP.

RTP

- ✓ Não possui nenhum mecanismo que assegure a entrega de mídia a tempo nem outras garantias de QoS.
- ✓ Não garante a entrega de pacotes.
- ✓ Evita a entrega de pacotes fora de ordem.
- ✓ Suporta a transmissão em *multicast*.
- ✓ O RTP permite atribuir a cada fonte (por exemplo, uma câmara ou um microfone) o seu próprio fluxo de pacotes RTP independente.

RTP

- ✓ Por exemplo, para uma **videoconferência** entre dois participantes, quatro fluxos RTP poderiam ser abertos:
 - **Dois fluxos** para transmitir o **áudio** (um em cada direção);
 - **Dois fluxos** para o **vídeo** (novamente, um em cada direção).
- ✓ Contudo, algumas técnicas de codificação populares, incluindo **MPEG1** e **MPEG2** -- reúnem o áudio e o vídeo num único fluxo durante o processo de codificação.
- ✓ Quando o **áudio** e o **vídeo** são **reunidos** pelo **codificador**, então apenas um fluxo RTP é gerado em cada direção.

RTP

- ✓ **Formato do Pacote RTP**

V=2	P	E	CC	M	PT	Sequence Number
Timestamp						
Synchronization source (SSRC) Identifier						
Contributing source (CSRC) Identifiers (Variable)						
Data (Variable)						

Onde:

CC Contributor count
 E Extension
 M Marker
 P Padding
 PT Payload Type
 V Version

RTCP

- ✓ O *RTP Control Protocol* têm por principal função prover **realimentação** da **qualidade da distribuição** das mídias para o transmissor, de forma que este possa tomar ações de melhoria.
- ✓ Para tanto, o RTCP opera em conjunto com o RTP.
- ✓ O RTCP é definido na mesma RFC que o RTP (**RFC 3550**).
- ✓ Cada participante de uma sessão RTP transmite **periodicamente** pacotes de controle RTCP para todos os outros participantes.

RTCP

- ✓ Cada pacote RTCP contém **relatórios** do transmissor e/ou receptor.
- ✓ As estatísticas contidas nos relatórios são úteis para a aplicação, podendo ser utilizadas para **controlar** o **desempenho das transmissões**.
- ✓ As estatísticas incluem o número de pacotes enviados, número de pacotes perdidos, estimativas de *jitter*, *timestamps*, etc.

RTCP

- ✓ Para cada pacote RTP que um receptor recebe ele gera um relatório de recepção.
- ✓ O receptor agrega seus relatórios de recepção a um único pacote RTCP.
- ✓ O pacote RTCP é então enviado para o transmissor ou para a árvore multicast que congrega todos os participantes de uma determinada sessão.
- ✓ O transmissor também envia pacotes com relatórios de transmissão.

RTSP

- ✓ O *Real Time Streaming Protocol* foi desenvolvido pela IETF em 1998 na RFC 2326.
- ✓ O RTSP é um protocolo que permite que os usuários controlem remotamente um fluxo de mídia contínua.
- ✓ Ações de controle incluem: voltar ao início, avançar, pausa, continuar, seleção de trilha, etc...
- ✓ O RTSP é um protocolo de aplicação do tipo cliente/servidor.

Arquitetura SIP

- ✓ Introdução
- ✓ Elementos da Arquitetura
- ✓ Endereçamento
- ✓ Mensagens
- ✓ Registrando um Terminal
- ✓ Estabelecendo uma Sessão
- ✓ Encerrando uma Sessão

Introdução

- ✓ O *Session Initiation Protocol* é um protocolo de sinalização definido pelo IETF (RFC 3261) para a **criação**, **modificação** e **estabelecimento** de sessões com um ou mais participantes na Internet.
- ✓ O protocolo SIP foi idealizado para ser utilizado **especificamente** na Internet.
- ✓ Portanto, a interconexão com as redes telefônicas tradicionais deve ser feita através de *gateways*.
- ✓ O SIP é baseado na **arquitetura cliente/servidor** do HTTP – *Hipertext Transfer Protocol*.

Elementos da Arquitetura

- ✓ A **arquitetura SIP** é composta dos seguintes **elementos**:
 - Terminal SIP
 - Servidor *Proxy*
 - Servidor de Registro
 - Servidor de Localização

Terminal SIP

- ✓ O **terminal SIP** é um equipamento a partir do qual um usuário estabelece uma sessão SIP.
- ✓ O terminal SIP pode ser um **computador** com um **software** VoIP ou um **telefone** SIP.



Terminal SIP

- ✓ Um **terminal SIP** é composto de **dois agentes (User Agents)**:
 - **Agente Cliente do Usuário (UAC – User Agent Client)**
 - É responsável por gerar requisições e receber as respostas.
 - **Agente Servidor do Usuário (UAS – User Agent Server)**
 - É responsável por receber as requisições e respondê-las.

Servidor Proxy

- ✓ É responsável pelo **roteamento** das **requisições de novas chamadas**.
- ✓ Existem dois tipos de servidores *proxy*:
 - **Sem Estado (Stateless)**
 - Simplesmente repassa as requisições, não mantendo nenhuma informação de estado a respeito do estabelecimento da sessão.
 - **Com Estado (Forking)**
 - Atua como um procurador para o terminal, mantendo informações de estado a respeito do estabelecimento da sessão.

Servidor de Registro

- ✓ É responsável por **registrar** os **endereços SIP** e os **endereços reais** dos usuários SIP.
- ✓ Um *proxy* pode **solicitar** uma **pesquisa** para determinar se um determinado contato está registrado em um servidor de registros.
- ✓ O servidor de registros fará uma **busca** em seu **arquivo de registros** e responderá ao *proxy*.

Servidor de Localização

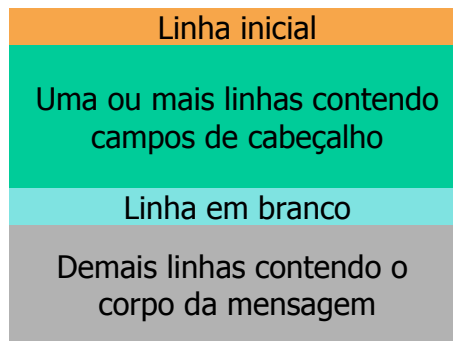
- ✓ **Servidores de localização** são **bancos de dados** que armazenam as informações de registro dos **servidores de registro**, e respondem a consultas enviadas por **servidores de redirecionamento** e *proxy*.
- ✓ Podem ser **implementados junto** aos outros três tipos de servidores.
- ✓ Porém, o SIP **não especifica** nenhum método de consulta a um servidor de localização.

Endereçamento

- ✓ O SIP usa um **formato de endereço** semelhante ao formato do e-mail para identificar os seus usuários:
 - **sip:** **usuário@domínio**
 - **sip:** o endereço SIP é escrito como um URL (*Uniform Resource Locator*).
 - **usuário** pode ser o nome do usuário ou o número do seu telefone.
 - **domínio** pode ser o domínio ao qual ele pertence ou o próprio endereço IP.
- ✓ Geralmente, o endereço SIP acaba sendo o próprio e-mail do usuário: **sip:** alberti@inatel.br.
- ✓ Outros URLs também podem ser utilizados:
 - **tel:** +553534719245
 - **fax:** +358.555.1234567

Mensagens

- ✓ O protocolo SIP usa **mensagens textuais**, muito semelhantes as mensagens HTTP.
- ✓ A figura abaixo mostra o **Formato Genérico das Mensagens SIP**:

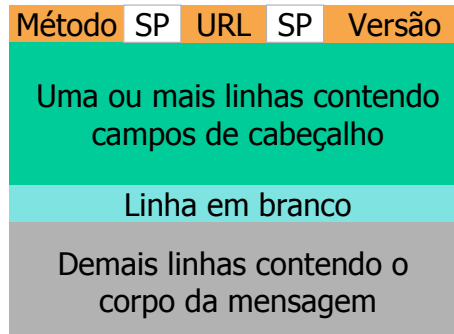


Mensagens

- ✓ Existem **dois tipos** de mensagens:
 - **Mensagem de Requisição**
 - Requistam uma ação do destinatário.
 - A ação requisitada é chamada de método.
 - **Mensagem de Resposta**
 - Indicam o resultado do processamento de uma requisição.
 - Indicam a ação tomada pelo destinatário da requisição.

Mensagens de Requisição

- ✓ **Formato**



- A linha inicial é chamada de linha de requisição:
 - **Método**: o método que deverá ser acionado no destinatário.
 - **URL**: endereço do destinatário.
 - **Versão**: versão do protocolo SIP; atualmente: SIP/2.0.

Mensagens de Requisição

✓ Métodos

- INVITE
 - Convida um usuário a participar de uma sessão.
- ACK
 - Confirma que o cliente recebeu uma mensagem resposta após enviar um INVITE.
- BYE
 - Indica ao servidor que um terminal deseja finalizar a sessão.
- CANCEL
 - Cancela um pedido que ainda não tenha sido atendido.

Mensagens de Requisição

✓ Métodos (cont.)

- OPTIONS
 - Solicita informações sobre capacidade, mas não estabelece sessão.
- REGISTER
 - Registra um usuário, enviando informações de localização do usuário para um servidor SIP.

✓ Exemplos

- INVITE sip:marconi@radio.org SIP/2.0
- INVITE sip:john@192.190.132.31 SIP/2.0
- REGISTER sip:registrar.munich.de SIP/2.0
- ACK sip:marconi@radio.org SIP/2.0
- BYE sip:tesla@high-voltage.org SIP/2.0

Mensagens de Resposta

✓ Formato

Versão	SP	Código	SP	Descrição
--------	----	--------	----	-----------

Uma ou mais linhas contendo campos de cabeçalho

Linha em branco

Demais linhas contendo o corpo da mensagem

- A linha inicial é chamada de linha de *status*:
 - **Versão**: versão do protocolo SIP; atualmente: SIP/2.0.
 - **Código**: contém o código de *status* do resultado de uma requisição.
 - **Descrição**: contém uma descrição curta do código de *status*.

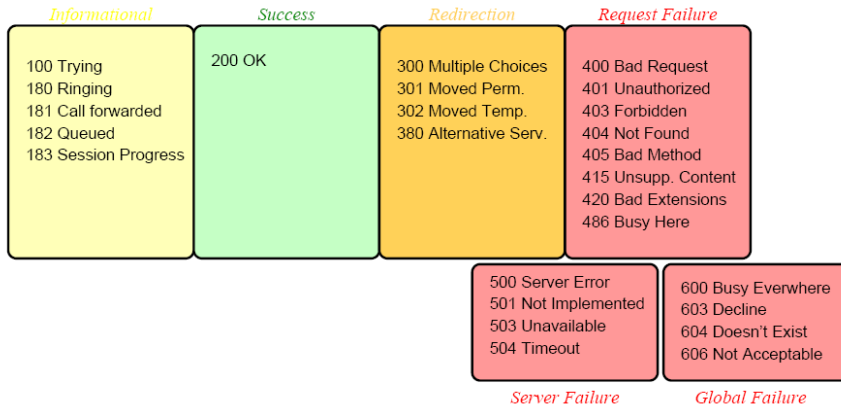
Mensagens de Resposta

✓ Formato do Código de Status

- Possui 3 dígitos que indicam o resultado de uma requisição:
 - **1º Dígito**: Define a classe da resposta.
 - **2º e 3º Dígito**: Definem a resposta.

Mensagens de Resposta

✓ Exemplos



Campos de Cabeçalho

✓ Existem **quatro** tipos de **campo de cabeçalho**:

General	Entity	Request	Response
Call-ID	Content-Encoding	Accept	Allow
Contact	Content-Length	Accept-Encoding	Proxy-Authenticate
Cseq	Content-Type	Accept-Language	Retry-After
Date		Authorization	Server
Enryption		Contact	Unsupported
Expires		Hide	Warning
From		Max-Forwards	WWW-Authenticate
Record-Route		Organization	
Timestamp		Priority	
To		Proxy-Authorization	
Via		Proxy-Require	
		Route	
		Require	
		Response-Key	
		Subject	
		User-Agent	

Campos de Cabeçalho

- ✓ O formato geral de cada campo de cabeçalho é:
 - **nome-do-campo**: **valor-do-campo**; **parâmetro** = **valor-parâmetro**

- ✓ Dentre os principais campos estão:
 - **To**:
 - Indica para que usuário e endereço se destina uma requisição.
 - Exemplo → **To: Bill Gates <sip:billgates@microsoft.com>**

 - **From**:
 - Indica o originador de uma requisição.
 - Exemplo → **From: Albert Einstein <sip:einstein@relativity.com>**

Campos de Cabeçalho

- **Via**:
 - Em requisições recém geradas por um terminal, informa para que endereço e porta a mensagem de resposta deve ser enviada e qual é o protocolo que deve ser usado na camada de transporte.
 - Exemplo → **Via: SIP/2.0/UDP 192.9.4.3:5060**
 - Quando uma requisição passa por um *proxy*, um novo campo *via* é acrescentado na requisição. O objetivo é indicar qual é a rota de retorno que deve ser tomada de volta para o originador da requisição.
 - Exemplo → **Via: SIP/2.0/UDP proxy.192.10.5.1:5060**
Via: SIP/2.0/UDP 192.9.4.3:5060

- **Call-ID**:
 - Usado para identificar a sessão, permanecendo sempre o mesmo valor.
 - Ajuda a detectar respostas duplicadas.
 - Exemplo → **94non40jnb4890hjt**

Corpo das Mensagens: O Protocolo SDP

- ✓ No caso de sessões multimídia, o protocolo SIP carrega como corpo de suas mensagens um outro protocolo chamado **SDP** – *Session Description Protocol*.
- ✓ Este protocolo é especificado na **RFC 2327**.
- ✓ Assim, uma sessão multimídia é descrita pelo protocolo SDP e é estabelecida através do protocolo SIP.
- ✓ O protocolo SIP, então, transporta em seu corpo a mensagem SDP.

Corpo das Mensagens: O Protocolo SDP

- ✓ O propósito básico do SDP é **conter informações** sobre os **fluxos de mídia** da sessão para permitir que as partes envolvidas possam estabelecer canais com as configurações apropriadas.
- ✓ Para este propósito, o protocolo SDP inclui em sua mensagem campos que descrevem:
 - O **nome** da sessão e o **propósito**;
 - O **tempo** que a sessão permanecerá ativa, quando a sessão é limitada em tempo;

Corpo das Mensagens: O Protocolo SDP

- ✓ Continuação:
 - A **mídia** que será trocada na sessão; e
 - Os **parâmetros** para se receber a mídia, que incluem: o **tipo de mídia** (áudio, vídeo, etc...), o **protocolo de transporte** (RTP/UDP/IP, etc.), e o **formato da mídia** (H.261, G.711, etc...).

Corpo das Mensagens: O Protocolo SDP

- ✓ As mensagens SDP consistem de um **número finito de linhas** de campos que descrevem os parâmetros da sessão.
- ✓ Estes campos possuem a seguinte formato:
 - **<tipo>=<valor>**
 - O **<tipo>** é um caractere que identifica o campo.
 - O **<valor>** é uma *string* que contém o valor do campo.

Corpo das Mensagens: O Protocolo SDP

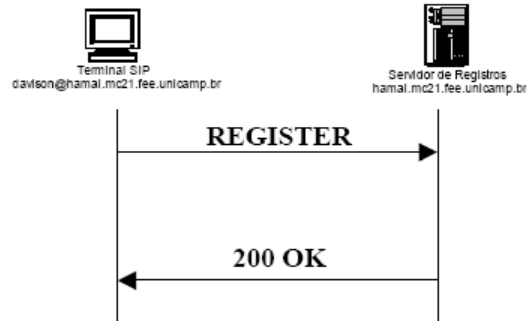
- ✓ Os campos da mensagem SDP são subdivididos da seguinte forma:
 - **Descritores da Sessão**
 - v= (versão do protocolo)
 - o= (identificação do requisitante da sessão)
 - s= (nome da sessão)
 - i= (informação sobre a sessão)
 - u= (página da Internet que contenha a descrição da sessão)
 - e= (endereço de e-mail do requisitante)
 - p= (número do telefone do requisitante)
 - c= (informações sobre a conexão)
 - b= (largura de banda exigida pela sessão)
 - z= (ajuste do relógio entre as localidades)
 - k= (chave criptográfica)
 - a= (atributos da sessão)

Registrando um Terminal

- ✓ Quando um terminal SIP inicia, ele **deve** efetuar o **registro** dos seus endereços IP e SIP em um servidor de registros.
- ✓ Suponhamos que um terminal SIP deseje registrar o seu endereço SIP, davison@hamal.mc21.fee.unicamp.br, no servidor de registros do domínio.
- ✓ O endereço IP do Terminal SIP é `regulus.mc21.fee.unicamp.br`.
- ✓ Portanto, o Terminal SIP gera uma requisição REGISTER.

Registrando um Terminal

✓ Transação REGISTER



Fonte: Davison

© Antônio M. Alberti

Registrando um Terminal

✓ Mensagem REGISTER

```
REGISTER sip:hamal.mc21.fee.unicamp.br SIP/2.0
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4bnie7480nvo4
Max-Forwards: 70
To: Davison <sip:davison@hamal.mc21.fee.unicamp.br>
From: Davison <sip:davison@hamal.mc21.fee.unicamp.br>;tag=4n99b4
Call-ID: 94non40jnb4890hjt
CSeq: 4956 REGISTER
Contact: <sip:davison@regulus.mc21.fee.unicamp.br>
Expires: 3600
Content-Length: 0
```

Fonte: Davison

© Antônio M. Alberti

Registrando um Terminal

✓ Resposta 200 OK

SIP/2.0 200 Ok
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4bnie7480nvo4
;received=143.106.50.80
Max-Forwards: 70
To: Davison <sip:davison@hamal.mc21.fee.unicamp.br>
From: Davison <sip:davison@hamal.mc21.fee.unicamp.br>;tag=4n99b4
Call-ID: 94non40jnb4890hjt
CSeq: 4956 REGISTER
Contact: <sip:davison@regulus.mc21.fee.unicamp.br>
Expires: 3600
Content-Length: 0

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

- ✓ A transação mais importante do protocolo SIP é a transação INVITE.
- ✓ A mensagem INVITE é que estabelece as sessões SIP.
- ✓ Suponhamos que dois terminais: SIP 1 e SIP 2, já tenham efetuado o registro utilizando a requisição REGISTER.
- ✓ Vamos agora explorar como se dá o estabelecimento de uma sessão entre estes dois terminais.

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

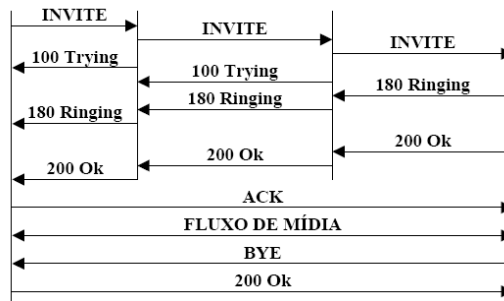
✓ Transação INVITE

Terminal SIP 1
davison@hamal.
mc21.fee.unicamp.br

Servidor de Registros e
Proxy do domínio
hamal.mc21.fee.unicamp.br

Servidor de Registros e
Proxy do domínio inatel.br

Terminal SIP 2
alberti@inatel.br



Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

✓ Requisição INVITE Terminal SIP 1

```
INVITE sip:alberti@inatel.br SIP/2.0
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4knib74u
Max-Forwards: 70
To: Antonio Marcos Alberti <sip:alberti@inatel.br>
From: Davison Gonzaga da Silva <sip:davison@hamal.mc21.fee.unicamp.br>;tag=487jni9
Call-ID: j94nbgh46yu48nbfi
CSeq: 7982 INVITE
Contact: <sip:davison@regulus.mc21.fee.unicamp.br>
Content-Type: application/sdp
Content-Length: 142
v=0
o=davison 165468471 216549871332 IN IP4 143.106.50.80
s=audio call
u=http://www.mc21.fee.unicamp.br
e=davison@decom.fee.unicamp.br
c=IN IP4 143.106.50.80/7000
t=0 0
m=audio 7000 udp 0
```

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

✓ Resposta 100 TRYING Proxy 1

SIP/2.0 100 Trying
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4knib74u;
received=143.106.50.80
Max-Forwards: 69
To: Antonio Marcos Alberti <sip:alberti@inatel.br>
From: Davison Gonzaga da Silva <sip:davison@hamal.mc21.fee.unicamp.br>;tag=487jni9
Call-ID: j94nbgh46yu48nbfi
CSeq: 7982 INVITE
Content-Length: 0

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

✓ Requisição INVITE Proxy 1

INVITE sip:alberti@inatel.br SIP/2.0
Via: SIP/2.0/UDP hamal.mc21.fee.unicamp.br:5060;branch=z9hG4k70bneoijb94
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4knib74u;
received=143.106.50.80
Max-Forwards: 70
To: Antonio Marcos Alberti <sip:alberti@inatel.br>
From: Davison Gonzaga da Silva <sip:davison@hamal.mc21.fee.unicamp.br>;tag=487jni9
Call-ID: j94nbgh46yu48nbfi
CSeq: 7982 INVITE
Contact: <sip:davison@regulus.mc21.fee.unicamp.br>
Content-Type: application/sdp
Content-Length: 142
v=0
o=davison 165468471 216549871332 IN IP4 143.106.50.80
s=audio call
u=http://www.mc21.fee.unicamp.br
e=davison@decom.fee.unicamp.br
c=IN IP4 143.106.50.80/7000
t=0 0
m=audio 7000 udp 0

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

✓ Resposta 200 OK Terminal SIP 2

```
SIP/2.0 200 Ok
Via: SIP/2.0/UDP inatel.br:5060;branch=z9hG4knavoin0485604;
received=143.106.56.200
Via: SIP/2.0/UDP hamal.mc21.fee.unicamp.br:5060;branch=z9hG4k70bneoiqb94;
received=143.106.50.69
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4knib74u;
received=143.106.50.80
Max-Forwards: 70
To: Antonio Marcos Alberti <sip:alberti@inatel.br>;tag=bniehb496cv
From: Davison Gonzaga da Silva <sip:davison@hamal.mc21.fee.unicamp.br>;tag=487jni9
Call-ID: j94nbgh46yu48nbfi
CSeq: 7982 INVITE
Contact: <sip:davison@regulus.mc21.fee.unicamp.br>
Content-Type: application/sdp
Content-Length: 142
```

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

✓ Resposta 200 OK Terminal SIP 2 (cont.)

```
v=0
o=alberti 165468471 216549871332 IN IP4 143.106.50.80
s=audio call
u=http://www.mc21.fee.unicamp.br
e=alberti@decom.fee.unicamp.br
c=IN IP4 143.106.56.193/7000
t=0 0
m=audio 7000 udp 0
```

Fonte: Davison

© Antônio M. Alberti

Estabelecendo uma Sessão

✓ Requisição ACK Terminal SIP 1

```
ACK sip:alberti@dte.inatel.br SIP/2.0
Via: SIP/2.0/UDP regulus.mc21.fee.unicamp.br:5060;branch=z9hG4knib74u
Max-Forwards: 70
To: Antonio Marcos Alberti <sip:alberti@inatel.br>;tag=bniehb496cv
From: Davison Gonzaga da Silva <sip:davison@hamal.mc21.fee.unicamp.br>;tag=487jni9
Call-ID: j94nbgh46yu48nbfi
CSeq: 7982 ACK
Contact: <sip:davison@regulus.mc21.fee.unicamp.br>
Content-Length: 0
```

Fonte: Davison

© Antônio M. Alberti

Arquitetura OPT

- ✓ Introdução
- ✓ Visão Sistêmica dos Componentes da Arquitetura
- ✓ Media Gateway
- ✓ Media Gateway Controller
- ✓ Exemplo de Estabelecimento de uma Chamada
- ✓ Media Gateway Control Protocol

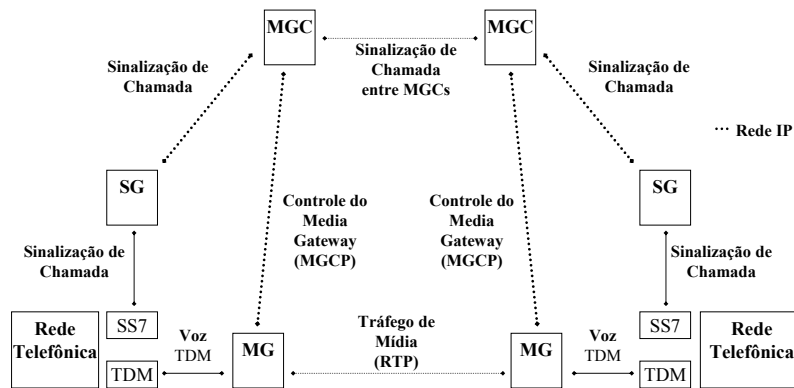
© Antônio M. Alberti

Introdução

- ✓ A arquitetura **Open Packet Telephony** é uma plataforma de convergência de **voz** e **dados** desenvolvida pela Cisco Systems.
- ✓ O termo **Open** vem do fato da arquitetura utilizar padrões públicos e não proprietários.
- ✓ Visa um **melhor aproveitamento** da **rede telefônica tradicional legada**, disponibilizando portanto mecanismos que permitam a interconexão com estas redes.

© Antônio M. Alberti

Visão Sistêmica dos Componentes da Arquitetura



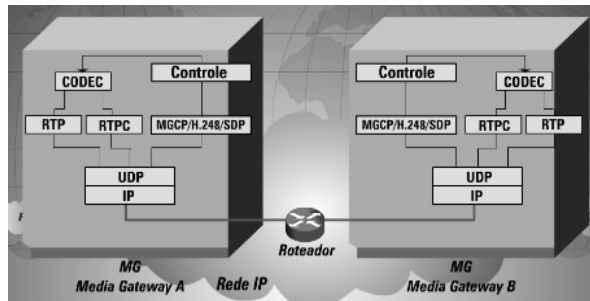
Fonte: Polidoro

SG: Signaling Gateway MG: Media Gateway MGC: Media Gateway Controller

© Antônio M. Alberti

Media Gateway

- ✓ **Realiza:**
 - Codificação de mídias entre redes diferentes.
 - Compressão e empacotamento de voz.
 - Cancelamento de eco.
 - Transmissão de voz usando RTP e RTCP.



Fonte: Polidoro

© Antônio M. Alberti

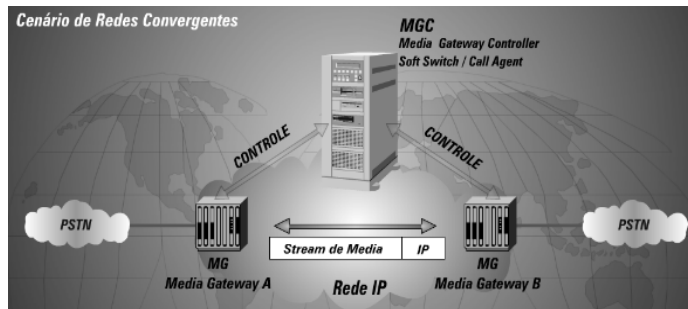
Media Gateway Controller

- ✓ O MGC é o elemento chave da arquitetura OPT, implementando toda a **inteligência** da rede.
- ✓ O MGC também é conhecido por **Call Agent** ou **Softswitch**.
- ✓ **Realiza:**
 - Resolução de endereços.
 - Gerenciamento de chamadas.
 - Tarifação.
 - Controle dos MGs utilizando o protocolo **MGCP** ou **MEGACO (H.248)**.

© Antônio M. Alberti

Media Gateway Controller

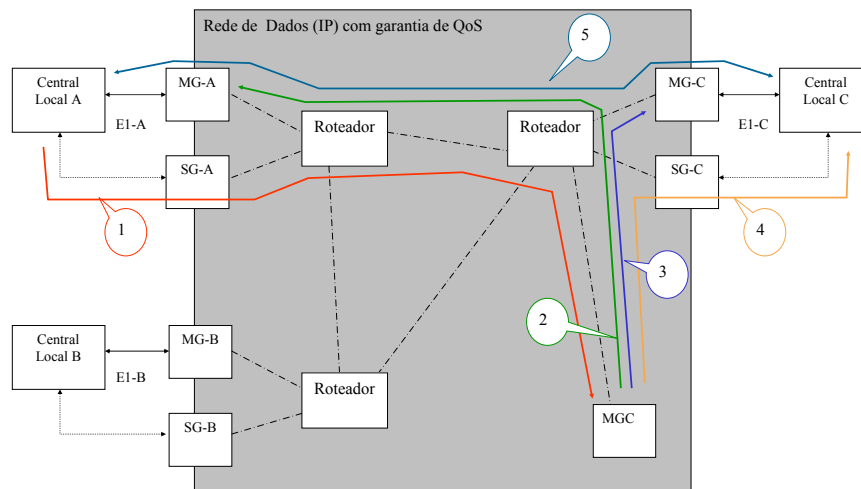
- ✓ O MGC estabelece uma **associação** entre o MG responsável pelo terminal de origem e o MG responsável pelo terminal de destino, criando uma espécie de **matriz de comutação virtual**.



Fonte: Polidoro

© Antônio M. Alberti

Exemplo de Estabelecimento de uma Chamada



Fonte: Polidoro

© Antônio M. Alberti

Exemplo de Estabelecimento de uma Chamada

✓ Transação 1

- A Central A envia uma mensagem SS7 para a central C realizando uma reserva de recursos e informando qual canal do tronco E1-A será utilizado.
- Esta mensagem é encapsulada pelo SG-A para um formato compatível com a rede de pacotes (SIGTRAN) e enviada ao MGC que baseado nela executa o passo 2.
- O MGC neste momento está interceptando a mensagem que iria para a central C.

Exemplo de Estabelecimento de uma Chamada

✓ Transação 2

- O MGC, através do protocolo MGCP ou MEGACO, cria uma associação interna ao MG-A entre o canal do E1-A fornecido no passo 1 e um *codec*/empacotador de mídia.
- Este “codec” será tanto responsável pela conversão das amostras de voz provenientes do canal E1 para o formato de pacotes como pelo sentido inverso.
- O endereçamento de destino dos pacotes provenientes da conversão será o MG-C.

Exemplo de Estabelecimento de uma Chamada

- ✓ **Transação 3**
 - O MGC, através do protocolo MGCP ou MEGACO, cria uma associação interna ao MG-C entre um canal do E1-C e um *codec*/empacotador de mídia.
 - Este “codec” será tanto responsável pela conversão das amostras de voz provenientes do canal E1 para o formato de pacotes como pelo sentido inverso.
 - O MGC neste ponto gera as mensagens como se fosse a central A.

Exemplo de Estabelecimento de uma Chamada

- ✓ **Transação 4**
 - O MGC gera uma mensagem encapsulada no formato SIGTRAN endereçada ao SG-C que realizará a “conversão” desta mensagem para o formato SS7.
 - Depois de realizada a conversão esta mensagem será enviada à central C realizando reserva de recursos e informando o canal do tronco E1 (selecionado no passo 3) em que estarão chegando as amostras de voz da chamada.
 - Neste ponto do estabelecimento da chamada a central C “pensa” que está recebendo as mensagens SS7 diretamente da central A.

Exemplo de Estabelecimento de uma Chamada

- ✓ **Transação 5**
 - As associações foram previamente estabelecidas nos MG-A e MG-C, que neste momento começam a trocar pacotes de mídia (Voz) diretamente entre si.

 - Para as centrais A e C todo o processo descrito anteriormente se passa de maneira transparente, ou seja, toda troca de sinalização, reserva de recursos e canais de voz ocorrem como sendo um processo normal de sinalização SS7.

Referências Bibliográficas

- ✓ **Monografias**
 - **POLIDORO**, M. M., “Arquitetura Open Packet Telephony”, Monografia Curso de Especialização, Inatel, Orientador: J. M. C. Brito.
 - **NARDIN**, A. M., **PORTES**, “Voz Sobre IP - Topologias e Protocolo SIP”, Monografia de Curso de Especialização, Inatel, Orientador: J. M. C. Brito.
- ✓ **Cursos**
 - **BRITO**, José Marcos C., “Introdução às Redes Multimídia”, Inatel.
 - **RODRIGUES**, Paulo Henrique de Aguiar, **LUSTOSA**, Leandro Caetano Gonçalves, **PEIXOTO**, João Carlos, **DAVID**, Fabio, “Telefonia IP”, Material Didático Disponível em: www.voip.nce.ufrj.br/courses/graduacao/tr9/aula9-1pp.pdf.

Referências Bibliográficas

✓ Artigos

- **LEOPOLDINO**, Graciela Machado, **MEDEIROS**, Rosa Cristina Martins, “H.323: Um padrão para sistemas de comunicação multimídia baseado em pacotes”, Revista *NewsGeneration* da RNP, volume 5, número 6, 2001.

✓ Teses

- **SILVA**, Davison Gonzaga, “Implementação de um Sistema SIP para o Sistema Operacional Linux”, Tese de Mestrado, FEEC, UNICAMP, 2003.