PTC 2550 - Aula 17

4.4 Protocolos para aplicativos de conversação em tempo real

(Kurose, p. 587 - 626)
(Peterson, p. 444-454)

02/06/2017

Muitos slides adaptados com autorização de J.F Kurose and K.W. Ross, All Rights Reserved

Multimídia: estrutura de tópicos

4.1 Aplicativos de rede multimídia

4.2 *Streaming* de vídeo armazenado

- 4.2.1 Distribuição de vídeo e CDN
- 4.2.2 Ferramentas para reprodução contínua

4.3 Voz-sobre-IP

4.4 Protocolos para aplicativos de conversação em tempo real

4.5 Suporte de rede para multimídia

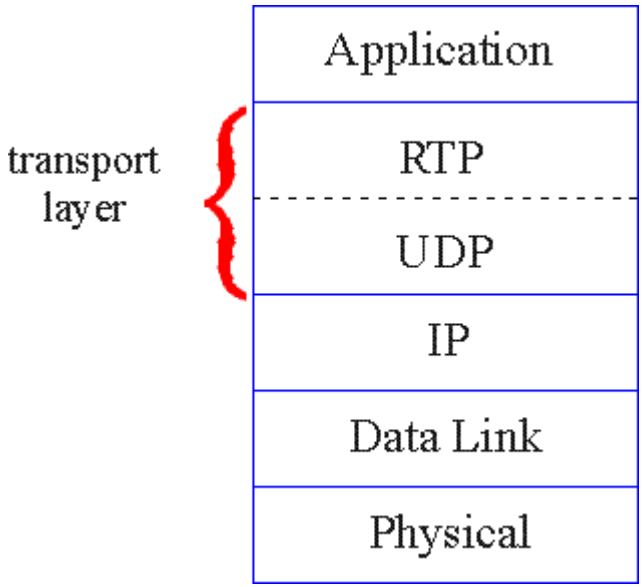
Real-Time Protocol (RTP)

- RTP: protocolo da camada de aplicação
- Especifica a estrutura para pacotes carregando dados de áudio ou vídeo
- RFC 3550 [2003]
- Cabeçalho RTP fornece
  - identificação do tipo de carga (dados)
  - número sequencial do pacote
  - marca de tempo (*time stamping*)
- RTP roda em sistemas finais
- Pacotes RTP encapsulados em segmentos UDP
- interoperabilidade: se dois aplicativos VoIP rodam RTP, podem conversar
- Mais informações: aqui
- Exemplo de app aberto que usa RTP

RTP roda sobre o UDP

Bibliotecas RTP fornecem interface com a camada de transporte que estende o UDP:

- endereços IP e números de portas (já tinha)
- identificação do tipo de "carga"
- número sequencial do pacote
- marca de tempo



RTP: Exemplo

Exemplo: enviando voz codificada em PCM 64 kbps sobre RTP

- aplicação divide dados codificados em bloco; por exemplo, 20 ms = 160 bytes em um bloco
- bloco de áudio + cabeçalho RTP forma pacote RTP, que é encapsulado em um segmento UDP

- Cabeçalho RTP indica tipo de codificação de áudio em cada pacote
  - transmissor pode mudar codificação de áudio durante sessão
- Cabeçalho RTP também contém números sequenciais e marcas de tempo

RTP e QoS

- RTP *não* fornece nenhum mecanismo para assegurar a entrega de dados no prazo ou qualquer outra garantia de QoS
- Encapsulamento RTP visto apenas nos sistemas finais (*invisível* para roteadores intermediários)
  - roteadores fornecem serviço "melhor possível" não fazendo nenhum esforço adicional para que pacotes RTP cheguem ao destino sem atraso

Cabeçalho RTP

tipo de carga (7 bits): indica o tipo de codificação sendo usado no momento. Se transmissor muda codificação durante chamado, informa receptor por esse campo

- Tipo 0: PCM lei μ, 64 kbps
- Tipo 3: GSM, 13 kbps
- Tipo 7: LPC, 2.4 kbps
- Tipo 26: Motion JPEG
- Tipo 31: H.261
- Tipo 33: MPEG2 vídeo

sequencial (16 bits): incrementado de 1 a cada pacote RTP enviado

- permite detectar perda de pacote e restaurar sequência de pacotes

Cabeçalho RTP

- *marca de tempo (32 bits):* instante de amostragem do primeiro byte nesse pacote de dados RTP
  - para áudio, marca de tempo incrementa de 1 a cada período de amostragem (por exemplo, a cada 125 μs para amostragem de 8 kHz)
  - se aplicação gera blocos de 160 amostras codificadas, marca de tempo aumenta de 160 em 160 para cada pacote RTP quando a fonte está ativa. Marca de tempo continua aumentando a taxa constante mesmo quando fonte não está ativa.

- *Campo SSRC (32 bits):* identifica fonte de um *stream* RTP. Cada sessão *stream* tem um SSRC diferente

SIP: Session Initiation Protocol [RFC 3261; RFC 5411]

visão de longo prazo:

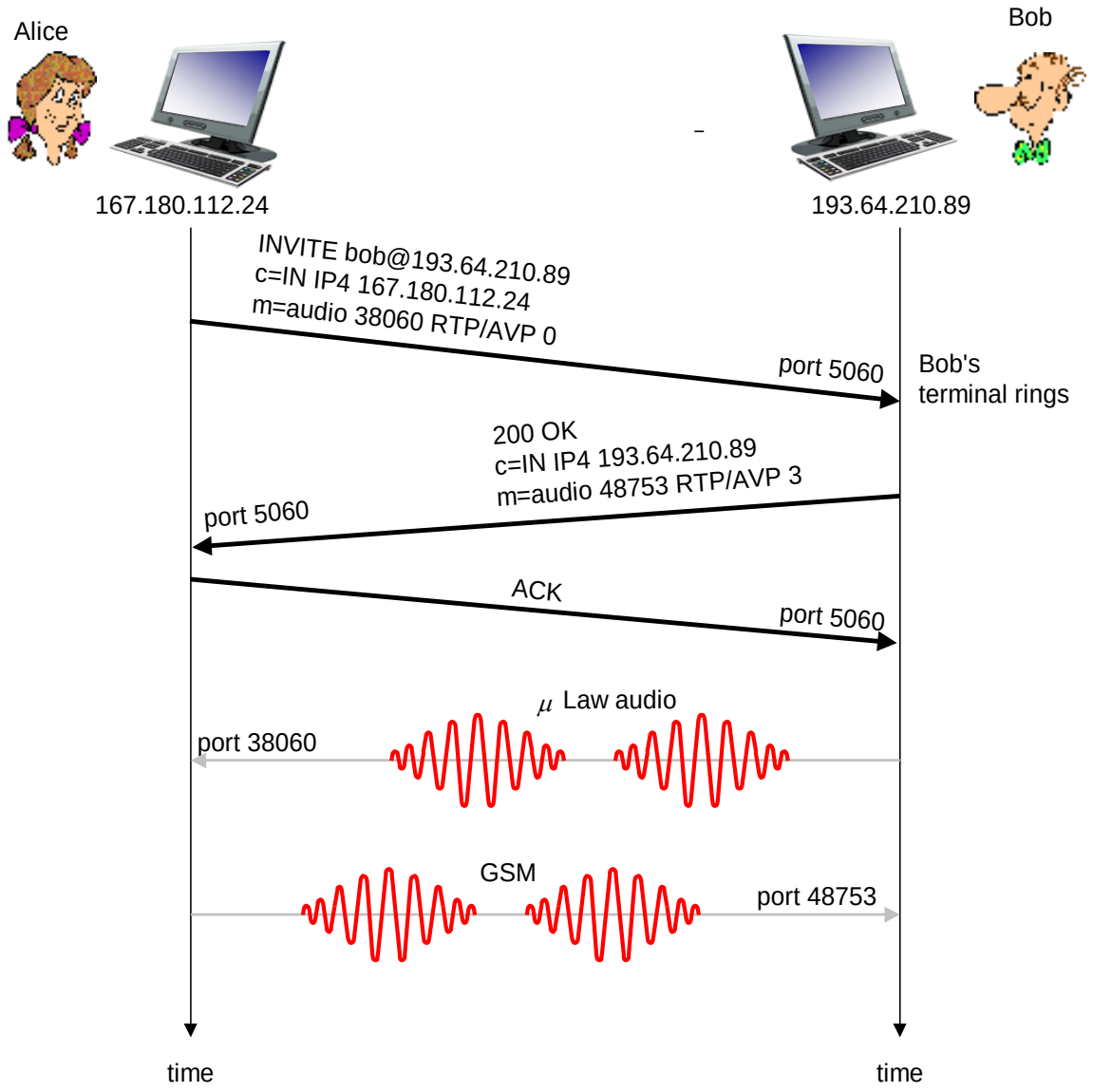
- todas as chamadas telefônicas e videoconferência ocorrem pela Internet
- pessoas identificadas por nomes ou endereços de e-mail ao invés de por números de telefone
- pode-se encontrar a pessoa chamada (se ela assim o desejar), não importa em que rede ela esteja ou o IP do dispositivo que ela esteja usando no momento

- Exemplo de cliente gratuito

Serviços SIP

- SIP prove mecanismos para estabelecimento de chamada:
  - quem faz a chamada notificar o destinatário saber que quer estabelecer uma ligação
  - ambos concordarem com tipo de mídia e codificação
  - finalizar a ligação
- determina endereço IP atual da pessoa chamada:
  - mapeia identificador mnemônico ao endereço IP atual
- gerencia a chamada:
  - adiciona novos *streams* de mídia durante a chamada
  - muda codificação durante chamada
  - convida outros
  - transfere, segura chamadas

Exemplo: fazendo chamada *no caso de IP conhecido*



- A mensagem `INVITE` de Alice indica seu número de porta, endereço IP e codificação em que ela prefere responder (PCM lei μ)
- A mensagem `200 OK` do Bob indica seu número de porta , endereço IP e codificação preferencial (GSM)
- Mensagens SIP podem ser enviadas via TCP ou UDP; aqui enviadas sobre RTP/UDP
- Número de porta SIP "bem conhecido" 5060

Fazendo uma chamada (mais)

- negociação do codec:
  - suponha que Bob não tenha um codificador PCM lei μ
  - Bob então envia resposta `606 Not Acceptable Reply`, listando seus codificadores. Alice pode então enviar nova mensagem *`INVITE`*, pedindo codificador diferente
- rejeitando uma chamada
  - Bob pode rejeitar com respostas "ocupado," "mudou-se," "pagamento necessário," "proibido"
- Lista de códigos de resposta
- mídia pode ser enviada sobre RTP ou alguns outros protocolos

Exemplo de mensagem SIP

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:

- sintaxe de mensagem "estilo" HTTP
- SDP = *session description protocol*
- ID de chamada (`Call-ID`) é único para cada chamada

- Aqui não sabemos o endereço IP atual do Bob
  - Servidores SIP intermediários necessários
- Alice envia e recebe mensagens SIP usando a porta padrão SIP 5060
- Alice especifica no cabeçalho que cliente SIP recebe e envia mensagens sobre UDP

Tradução de nome e localização de usuário

- Alice quer chamar Bob, mas sabe apenas o seu nome de usuários ou e-mail.
- Precisa obter endereço IP do *host* atual de Bob:
  - usuários é movel
  - protocolo DHCP
  - usuários tem diferentes dispositivos IP (PC de casa, PC do trabalho, *notebook*, *smartphone*, etc.)
- resultados podem ser baseados em:
  - horário do dia
  - quem chama (não quer receber chamadas do trabalho em casa)
  - situação de quem é chamado (chamada enviada por correio de voz se já está falando com alguém)

Registrador SIP

- uma das funções de um servidor SIP: registrar
- quando Bob inicializa cliente SIP, cliente envia mensagem SIP `REGISTER` para o servidor de registros de Bob

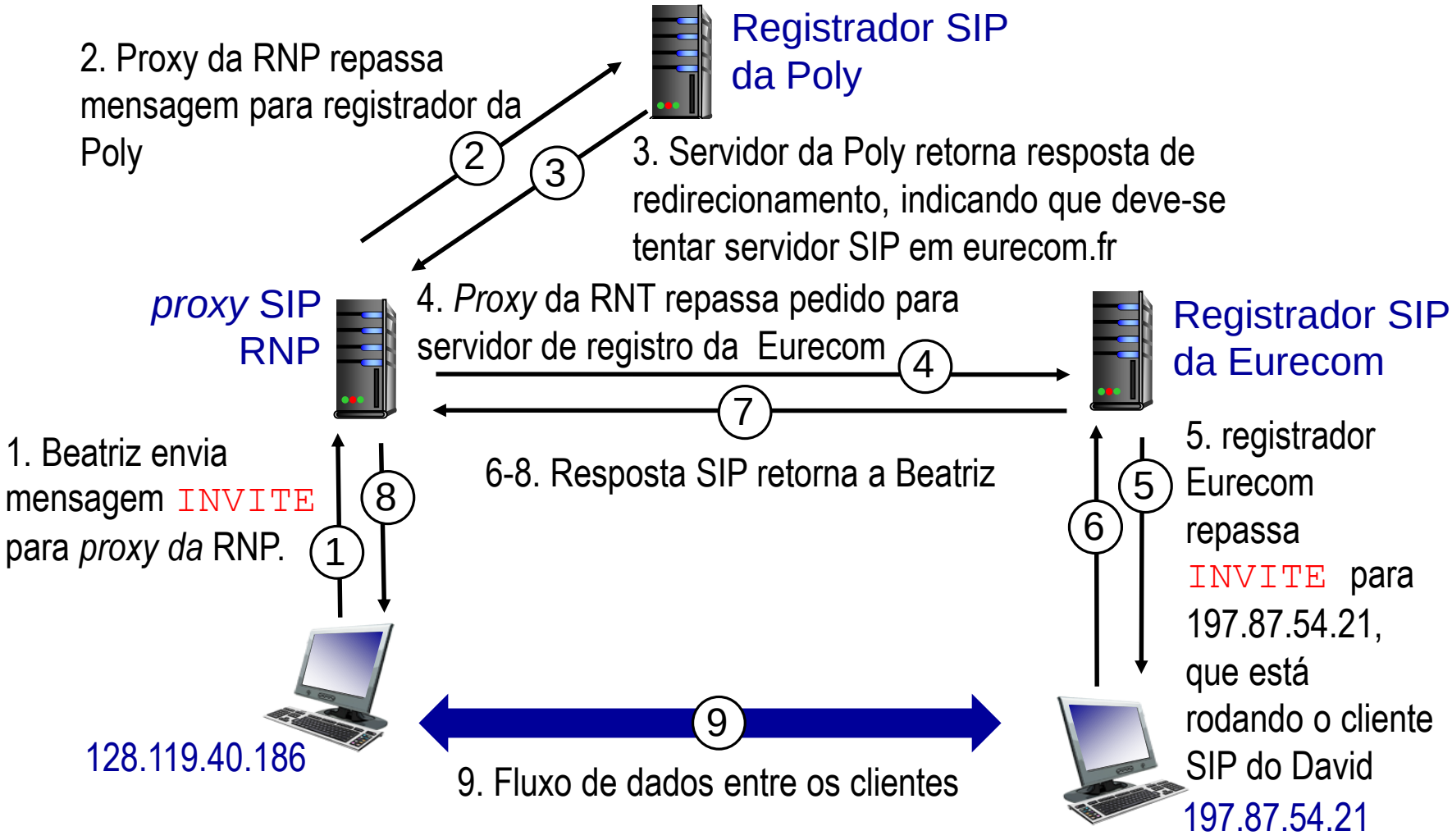
Mensagem `REGISTER`

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

Proxy SIP

- outra função do servidor SIP: *proxy*
- Alice envia mensagem `INVITE` para seu servidor *proxy*
  - contém endereço `sip:bob@domain.com`
  - proxy responsável por rotear mensagem SIP até Bob, possivelmente através de múltiplos *proxies*
- Bob envia resposta através do mesmo conjunto de *proxies* SIP
- *proxy* retorna mensagem resposta SIP de Bob para Alice
  - contém o endereço IP do Bob
- *proxy* SIP é análogo ao servidor DNS local mais *setup* TCP

Exemplo SIP : beatriz@rnp.br chama david@poly.edu



Comparação com H.323

- H.323: outro protocolo de sinalização para multimídia interativa em tempo real
- H.323: conjunto de protocolos verticalmente complete e integrado para conferência multimídia: sinalização, registro, controle de admissão, transporte, codecsc
- SIP: compomente único. Trabaça com RTP, mas não obrigatoriamante. Pode ser combinado com outros protocolos e serviços
- H.323 vem da ITU (telefonia)
- SIP vem da IETF: empresta muito dos seus conceitos do HTTP
  - SIP tem "cara" de Web; H.323 tem "cara" de telefonia
- SIP usa princípio KISS: Keep It Simple Stupid