UNIVERSIDADE FEDERAL DE PELOTAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

TEC2/TEC4: REDES MULTIMÍDIA (RMM)

Unidade 9

Voice-over-IP (VoIP)
Session Initiation Protocol (SIP)

Prof. Guilherme Corrêa
gcorrea@inf.ufpel.edu.br

Sumário

- Voz-sobre-IP (VoIP)
- Limitações do IP
- Eliminação de *jitter*
- Recuperação da Perda de Pacotes
- Estudo de Caso: VoIP com *Skype*
- Session Initiation Protocol (SIP)

Sumário

- **Voz-sobre-IP (VoIP)**
- Limitações do IP
- Eliminação de *jitter*
- Recuperação da Perda de Pacotes
- Estudo de Caso: VoIP com *Skype*
- Session Initiation Protocol (SIP)

Voz-over-IP (VoIP)

- *requisitos de atraso fim-a-fim na VoIP*: necessário manter aspecto de conversação
  - atrasos maiores são notados e comprometem a interatividade
  - < 150 ms: bom
  - > 400 ms: ruim
  - inclui atrasos no nível da aplicação (empacotamento, reprodução)
- *inicialização da sessão:* como o "chamador" avisa endereço de IP, número de porta, algoritmo de codificação?
- *serviços de valor agregado:* encaminhamento de chamada, gravação
- *serviços de emergência:* 911

Características de VoIP

- áudio de fala: rajadas de fala alternadas, períodos de silêncio
  - 64 kbps durante o período de fala
  - pacotes gerados apenas durante períodos de fala
  - blocos de 20 ms a 8 Kbytes/s: 160 bytes de dados
- cabeçalho no nível de aplicação adicionado a cada bloco (*chunk*)
- bloco+cabeçalho encapsulado em um segmento UDP ou TCP
- aplicação envia segmento para o *socket* a cada 20 ms durante o período de fala

Sumário

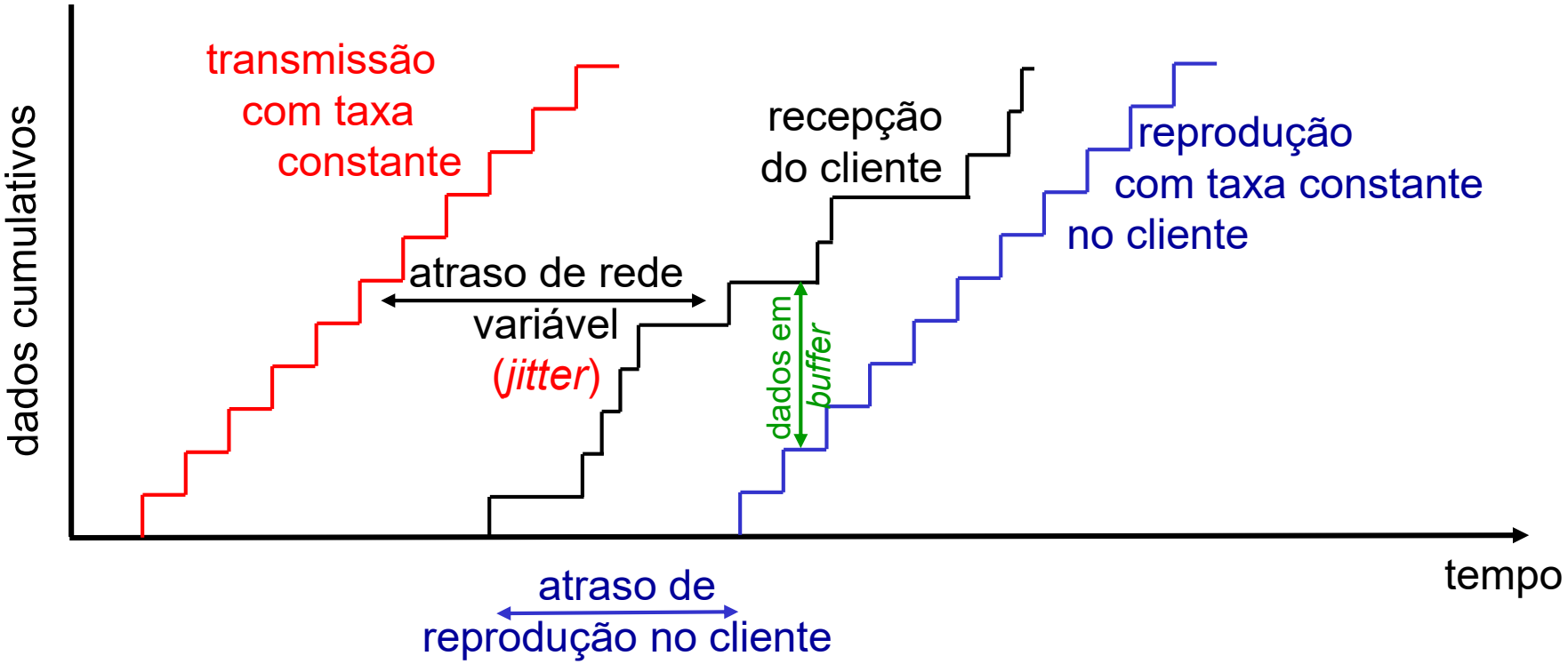
- Voz-sobre-IP (VoIP)
- **Limitações do IP**
- Eliminação de *jitter*
- Recuperação da Perda de Pacotes
- Estudo de Caso: VoIP com *Skype*
- Session Initiation Protocol (SIP)

Limitações do IP para VoIP

- *perda de pacote na rede:* datagrama IP perdido devido a congestionamento na rede (transbordo do buffer de roteador)
- *perda de pacote por atraso:* datagrama IP chega tarde demais para reprodução no receptor
  - atrasos: processamento, fila na rede; atrasos no sistema final (remetente, receptor)
  - atraso tolerável (tipicamente): 400 ms
- *tolerância a perda:* dependendo da codificação de voz usada e do modo de ocultamento de perda no receptor, taxas de perdas de 1% a 20% podem ser toleradas

Limitações do IP para VoIP

- *variação de atraso de pacote (jitter):*



- atraso fim-a-fim de dois pacotes consecutivos: diferença pode ser mais ou menos de 20 ms (diferença no tempo de transmissão)

Sumário

- Voz-sobre-IP (VoIP)
- Limitações do IP
- **Eliminação de *jitter***
- Recuperação da Perda de Pacotes
- Estudo de Caso: VoIP com *Skype*
- Session Initiation Protocol (SIP)

Eliminação de *jitter*

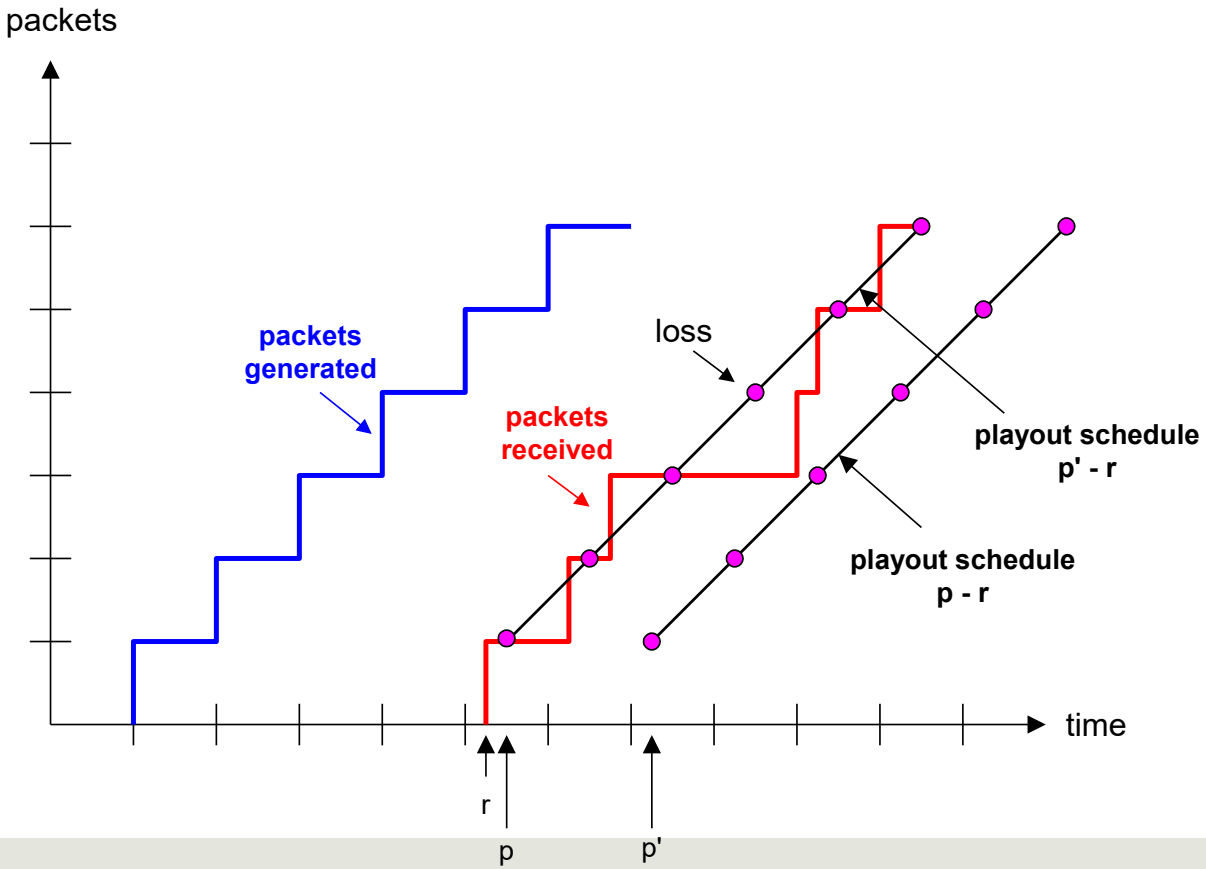
- receptor deve reproduzir de forma contínua, independente de variações de atraso aleatório
- Mecanismos:
  - *marca de tempo*: remetente precede cada parte com o instante em que foi gerada
  - *atraso de reprodução*: receptor guarda (*buffer*) os pacotes até que *x* pacotes estejam disponíveis para reprodução

Atraso por Reprodução Fixa

- receptor tenta reproduzir cada bloco exatamente *q* ms depois do bloco ser gerado
  - bloco tem marca de tempo *t*: reproduz bloco em *t*+*q*
  - bloco chega depois de *t*+*q*: tarde demais (dado "perdido")
- tradeoff na escolha de *q*:
  - *q* grande: menos pacotes perdidos
  - *q* pequeno: experiência interativa melhor

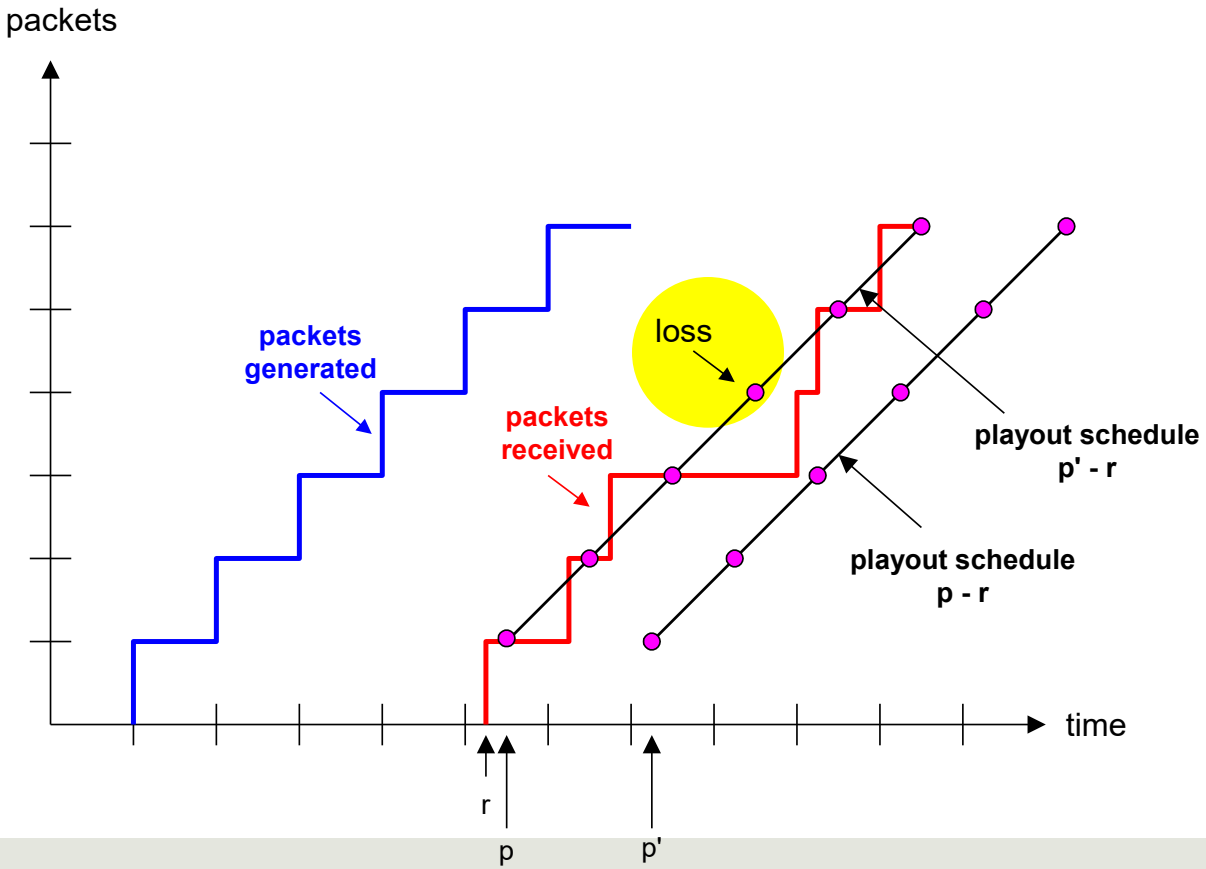
Atraso por Reprodução Fixa

- remetente gera pct a cada 20 ms durante período de fala
- primeiro pacote chega no tempo *r*
- primeiro esquema de reprodução: começa em *p*
- segundo esquema de reprodução: começa em *p'*



Atraso por Reprodução Fixa

- remetente gera pct a cada 20 ms durante período de fala
- primeiro pacote chega no tempo *r*
- primeiro esquema de reprodução: começa em *p*
- segundo esquema de reprodução: começa em *p'*



Atraso por Reprodução Adaptativa

- *objetivo:* baixo atraso de reprodução e perda por chegada atrasada pequena
- *abordagem:* atraso de reprodução com ajuste dinâmico
  - estima atraso de rede, ajusta atraso de reprodução no início de cada rajada de voz
  - períodos de silêncio comprimidos e alongados
  - blocos ainda reproduzidos a cada 20 ms durante rajadas
- estimativa adaptativa do atraso de pacote: (Média Móvel Exponencialmente Ponderada, lembre da estimativa de RTT no TCP):

$$d_i = (1-\alpha)d_{i-1} + \alpha\,(r_i - t_i)$$

- d_i: *atraso estimado depois do i-ésimo pacote*
- α: *constante pequena, ex.: 0.1*
- $r_i - t_i$: *atraso medido do i-ésimo pacote* = *momento de recepção* - *momento de envio (timestamp)*

Atraso por Reprodução Adaptativa

- também é útil estimar a média de desvio do atraso, v_i :

$$v_i = (1-\beta)v_{i-1} + \beta\,|r_i - t_i - d_i|$$

- estimativas d_i e v_i calculadas para cada pacote recebido, mas usadas apenas no começo da rajada de voz

- para o primeiro pacote na rajada, tempo de reprodução é:

$$playout\text{-}time_i = t_i + d_i + Kv_i$$

e os pacotes restantes na rajada são reproduzidos periodicamente

Atraso por Reprodução Adaptativa

Pergunta: Como o receptor sabe se o pacote é o primeiro em uma rajada?

- se não houver perda, receptor olha para as marcas de tempo sucessivas
  - diferença de *timestamps* sucessivas > 20 ms = nova rajada
- com possibilidade de perda, receptor deve olhar para marcas de tempo e números de sequência
  - diferença de *timestamps* sucessivas > 20 ms ***and*** número de sequência sem buracos = nova rajada

Sumário

- Voz-sobre-IP (VoIP)
- Limitações do IP
- Eliminação de *jitter*
- **Recuperação da Perda de Pacotes**
- Estudo de Caso: VoIP com *Skype*
- Session Initiation Protocol (SIP)

Recuperação de Perda de Pacotes

Desafio: recuperação de perda de pacote dado um pequeno atraso tolerável entre transmissão original e reprodução

- cada ACK/NAK leva aprox. um RTT
- alternativa: *Forward Error Correction (FEC)*
  - envia bits suficientes para permitir recuperação sem retransmissão (*lembrem da técnica de paridade bidimensional na camada de enlace!*)

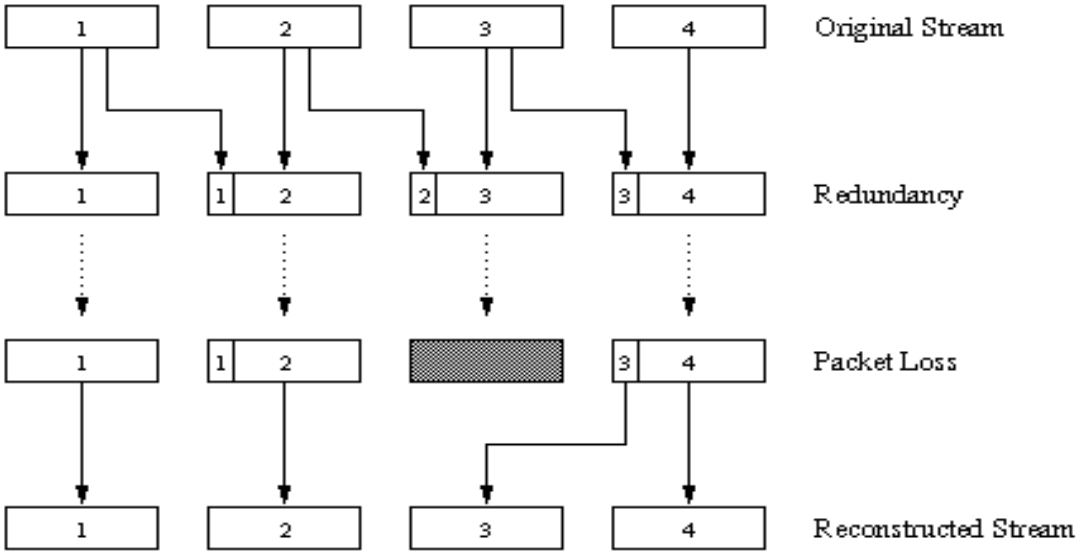
FEC simples

- para cada grupo de *n* blocos, cria um bloco redundante por XOR dos *n* blocos originais
- envia *n+1* blocos, aumentando largura de banda em *1/n*
- pode reconstruir os *n* blocos originais se no máximo um bloco for perdido

Recuperação de Perda de Pacotes

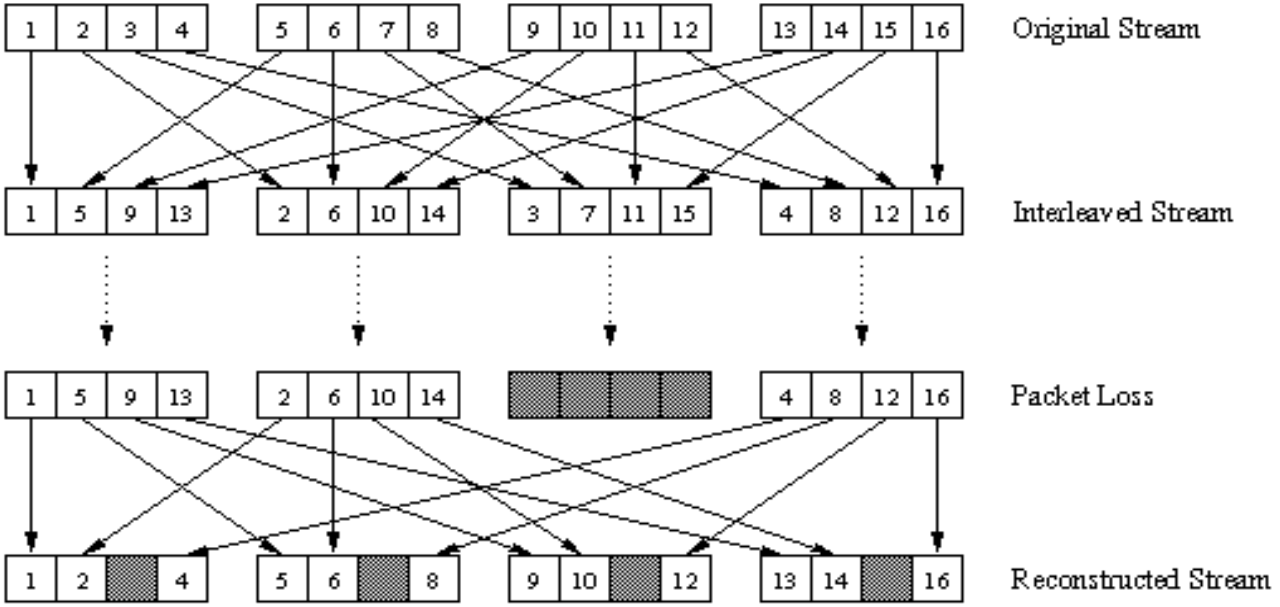
outro esquema FEC:

- fluxo de baixa qualidade "de carona"
- envia áudio de baixa resolução como info redundante
- ex.: fluxo nominal PCM a 64 kbps e fluxo GSM redundante a13 kbps



- perda não consecutiva: receptor pode ocultar erros
- generalização: poderia também anexar representação de baixa resolução do (n-1)-ésimo e do (n-2)-ésimo bloco

Recuperação de Perda de Pacotes



intercalar para ocultar perda:

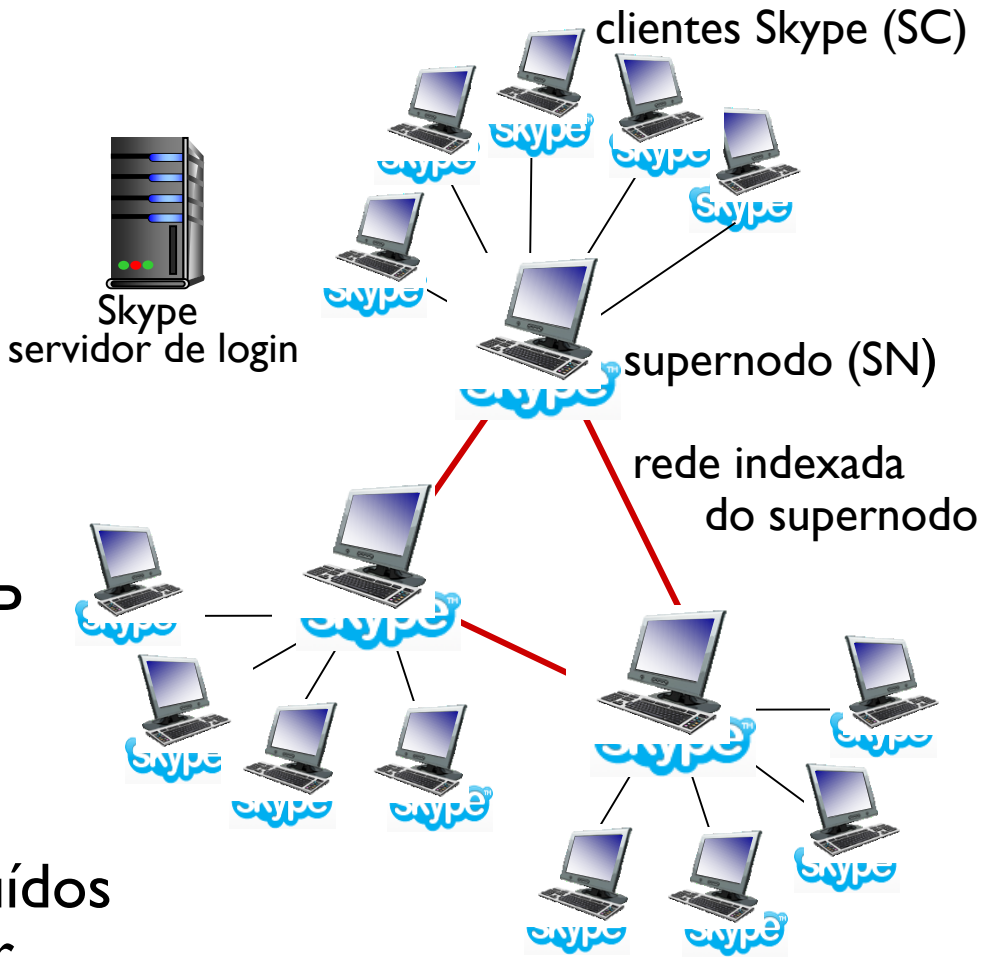
- blocos de áudio divididos em unidades menores, ex.: quatro unidades de 5 ms a cada 20 ms de áudio
- pacotes contêm unidades menores de blocos diferentes
- se pacote é perdido, ainda tem *maior parte* de cada bloco original
- sem overhead de redundância, mas aumenta atraso de reprodução

Sumário

- Voz-sobre-IP (VoIP)
- Limitações do IP
- Eliminação de *jitter*
- Recuperação da Perda de Pacotes
- **Estudo de Caso: VoIP com *Skype***
- Session Initiation Protocol (SIP)

Estudo de Caso: *Skype*

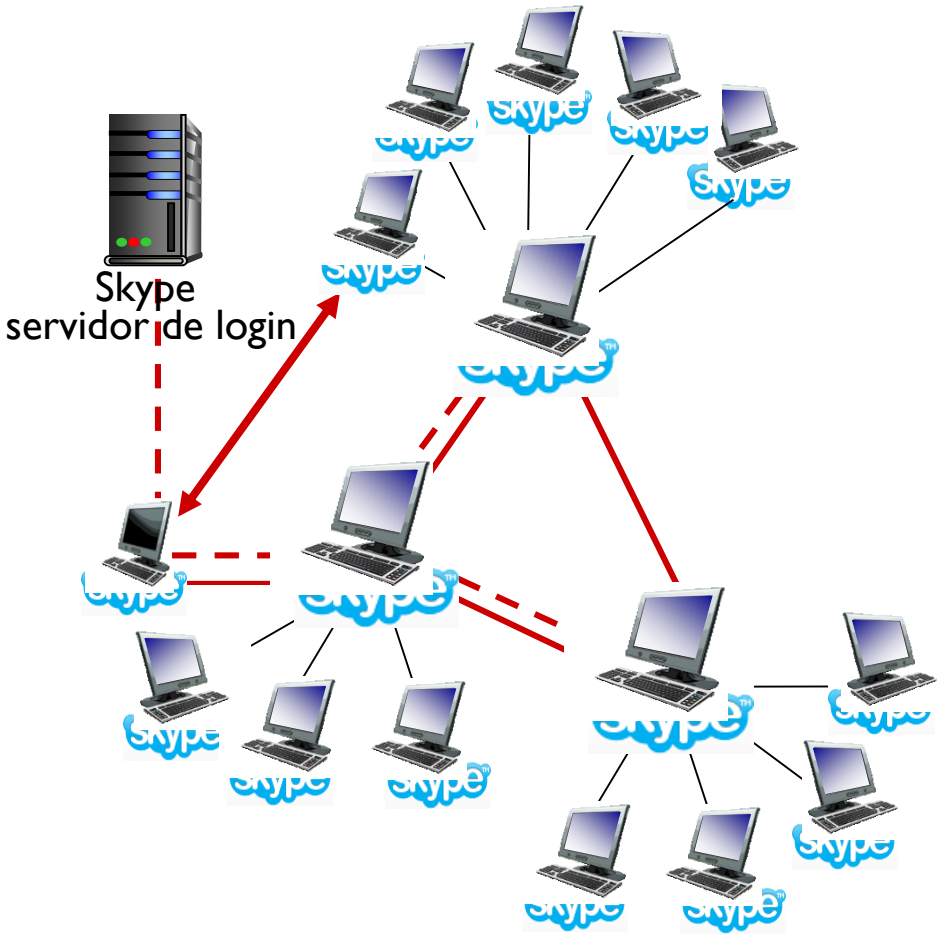
- protocolo proprietário da camada de aplicação (inferido via eng. reversa)
  - msgs criptografadas
- componentes P2P:
  - clientes (SC): pares skype conectam-se diretamente um ao outro por chamada VoIP
  - super nodos (SN): pares Skype com funções especiais
  - índices da rede: distribuídos entre SNs para localizar SCs
  - servidor de login



Estudo de Caso: *Skype*

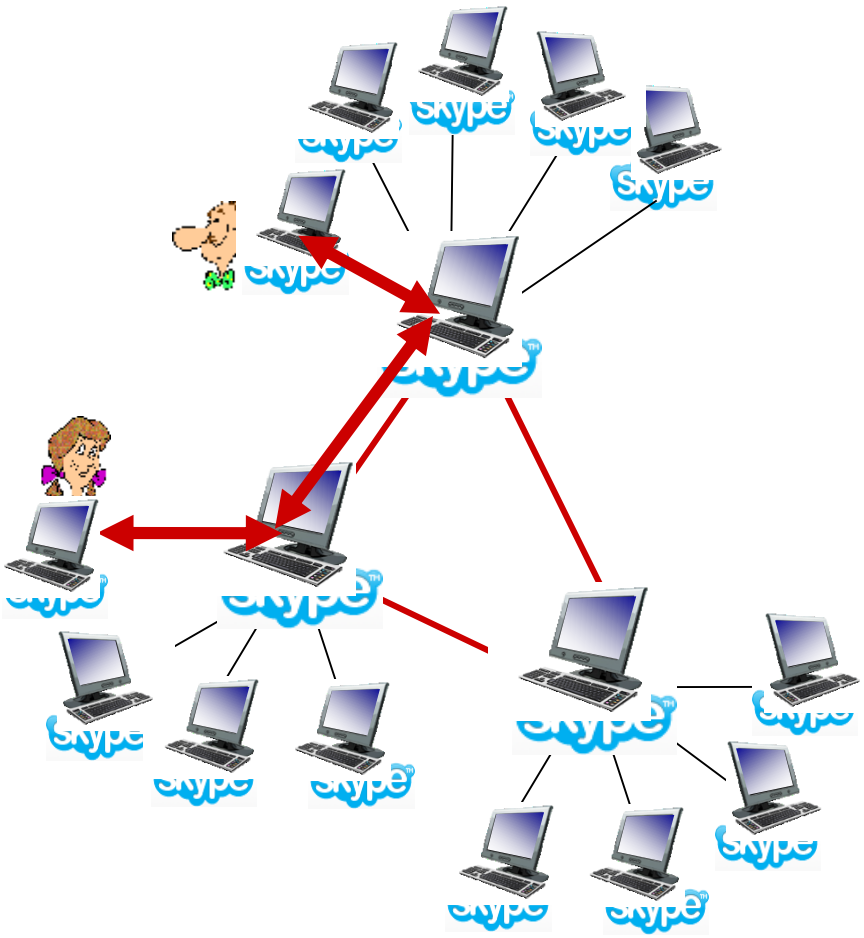
Operação do cliente Skype:

1. entra na rede Skype contatando o SN (end. IP guardado) usando TCP
2. faz login (username, password) no servidor de login centralizado
3. obtém end. IP para o cliente chamado a partir do SN e sua rede indexada ou lista de amigos do cliente
4. inicia chamada diretamente para o cliente chamado



Estudo de Caso: *Skype*

- *problema:* Alice e Bob estão atrás de NATs
  - NAT não permite que par externo inicie conexão para par interno
  - par interno *pode* iniciar conexão com externo
- *solução de repasse:* *Alice* e *Bob* mantêm conexão aberta para os seus SNs
  - Alice sinaliza ao seu SN para conectar com Bob
  - SN de Alice conecta com SN de Bob
  - SN de Bob se comunica com Bob sobre a conexão que Bob abriu originalmente para o seu SN



Sumário

- Voz-sobre-IP (VoIP)
- Limitações do IP
- Eliminação de *jitter*
- Recuperação da Perda de Pacotes
- Estudo de Caso: VoIP com *Skype*
- **Session Initiation Protocol (SIP)**

SIP: *Session Initiation Protocol*

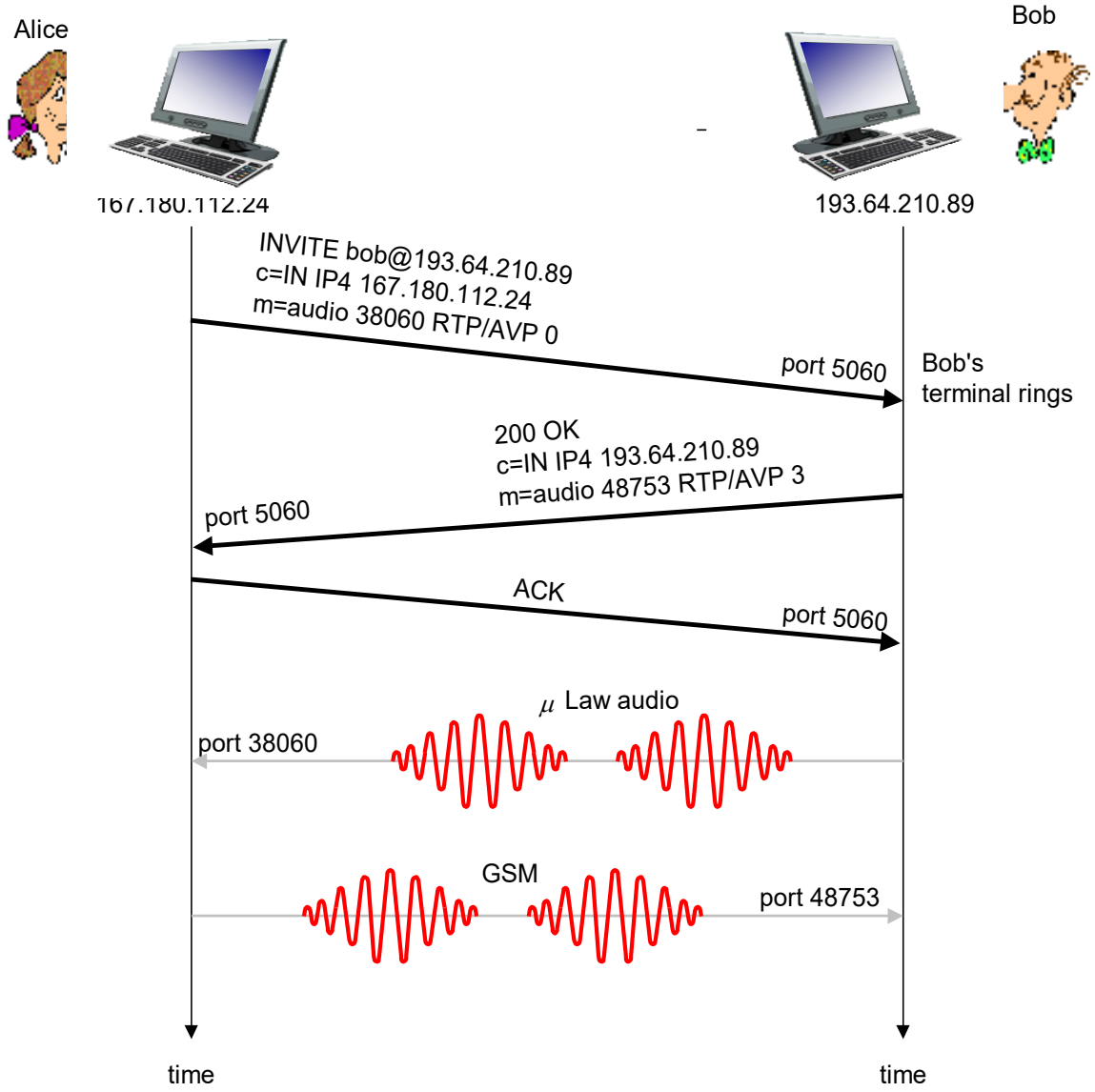
visão de longo prazo:

- todas chamadas telefônicas e de videoconferência vão ser realizadas pala internet
- pessoas identificadas por nomes ou endereços de e-mail, ao invés de números telefônicos
- pode alcançar receptor *(se receptor desejar)*, não importa onde este esteja, não importa qual end. de IP esteja atualmente usando

Serviços do SIP

- SIP provê mecanismos para configuração de chamada:
  - para quem chama avisar a quem é chamado que quer iniciar uma ligação
  - para interlocutores concordarem com tipo de mídia e codificação
  - para terminar ligação
- determina end. de IP atual de quem é chamado:
  - mapeia identificador mnemônico ao end. de IP atual
- gerenciamento de chamada:
  - adicionar novos fluxos de mídia durante a chamada
  - modificar codificação durante a chamada
  - convidar outros
  - transferir e pausar chamadas

Exemplo: config. chamada p/ IP conhecido



- mensagem **INVITE** do SIP de Alice indica número de porta, end. IP e codificação que ela prefere receber (PCM μlaw)
- mensagem **200 OK** de Bob indica número de porta, end. IP, codificação preferencial (GSM)
- msg SIP podem ser enviadas sobre TCP ou UDP; aqui no ex. são enviadas sobre RTP/UDP
- número de porta SIP padrão é 5060

Configurando chamada (mais)

- negociação do codec:
  - suponha que Bob não tem codificador PCM μlaw
  - Bob vai responder com **606 Not Acceptable Reply**, listando os seus codificadores
  - Alice pode então enviar novo **INVITE**, anunciando codificador diferente
- rejeitando uma chamada
  - Bob pode rejeitar com respostas "ocupado", "volto logo", "ausente", "pagamento requisitado", "proibido"
- mídia pode ser enviada sobre RTP ou outro protocolo

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
```

Notas:

- sintaxe de mensagem HTTP
- sdp = *session description protocol*
- Call-ID é único para cada chamada

- aqui nós não sabemos o end. IP de Bob
  - servidores SIP intermediários necessários
- Alice envia e recebe msgs SIP usando porta SIP padrão 5060
- Alice especifica no cabeçalho que o cliente SIP envia e recebe mensagens SIP sobre UDP

Tradução de nome e localização

- quem chama só sabe o nome ou endereço de e-mail de quem é chamado
- precisa obter end. IP de quem é chamado:
  - o usuário é nômade
  - protocolo DHCP
  - o usuário tem diferentes dispositivos IP (PC, tablet, smartphone, carro)
- resultado baseado em:
  - hora do dia (trabalho, casa)
  - quem chama (não quer que o chefe ligue quando está em casa)
  - status de quem é chamado (*voicemail* quando quem é chamado já está em uma ligação)

Entidade Registradora SIP

- uma função do servidor SIP: *entidade registradora*
- quando Bob inicia cliente SIP, cliente envia mensagem SIP REGISTER para a entidade registradora de Bob

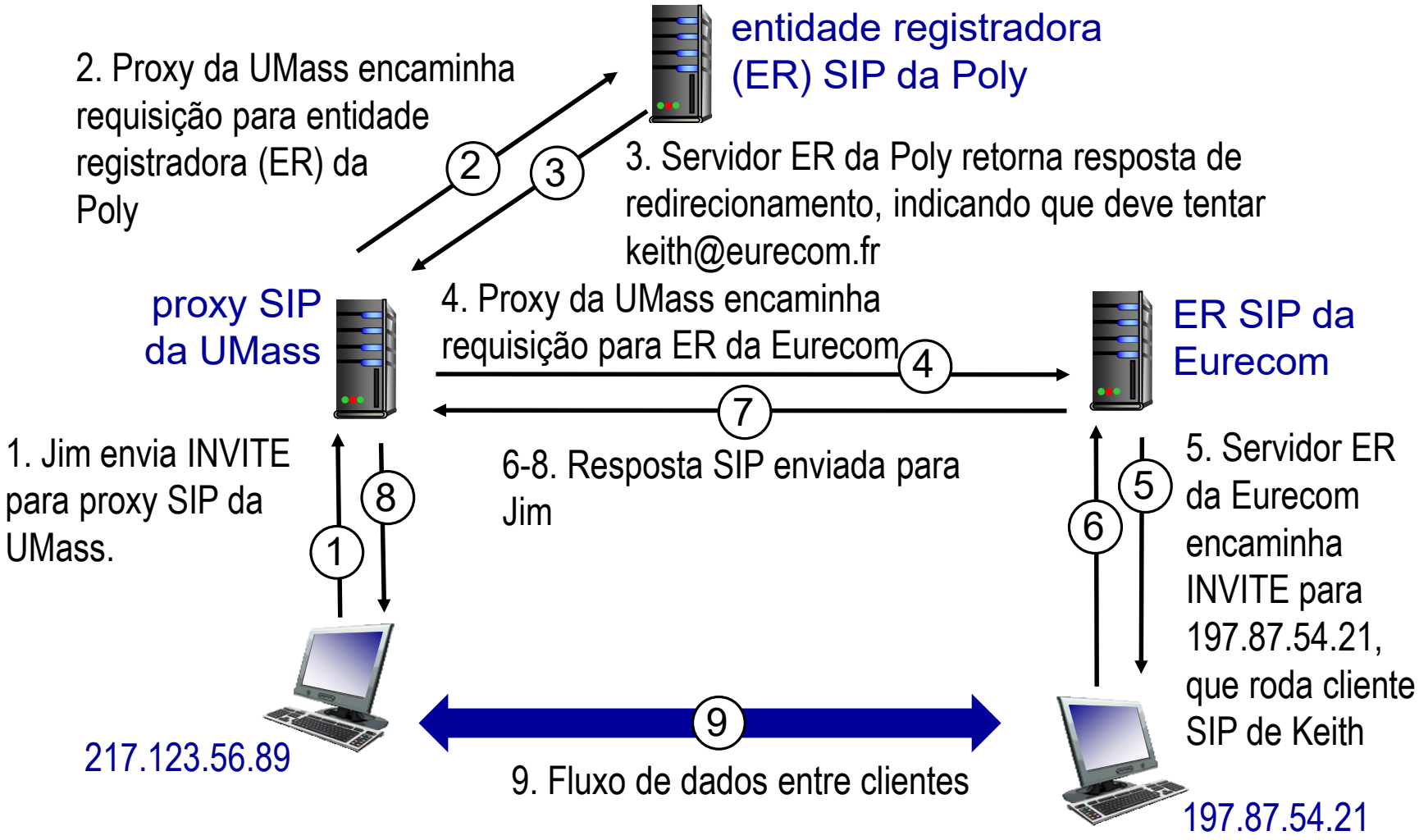
mensagem de registro:

```
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
```

SIP proxy

- outra função do servidor SIP: *proxy*
- Alice envia mensagem INVITE para o seu servidor proxy
  - contém endereço *sip:bob@domain.com*
  - proxy responsável por rotear mensagens SIP para quem é chamado, possivelmente através de múltiplos proxies
- Bob envia resposta de volta através do mesmo conjunto de proxies SIP
- proxy retorna resposta SIP de Bob para Alice
  - contém endereço IP de Bob
- proxy SIP parecido com servidor DNS local

Exemplo SIP: jim@umass.edu liga para keith@poly.edu



UNIVERSIDADE FEDERAL DE PELOTAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

TEC2/TEC4: REDES MULTIMÍDIA (RMM)

Unidade 9

Voice-over-IP (VoIP)
Session Initiation Protocol (SIP)

Prof. Guilherme Corrêa
gcorrea@inf.ufpel.edu.br