

# PTC 2550 - Aula 16

## 4.3 Voz sobre IP

(Kurose, p. 587 - 626)

(Peterson, p. 444-454)

31/05/2017

# 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**

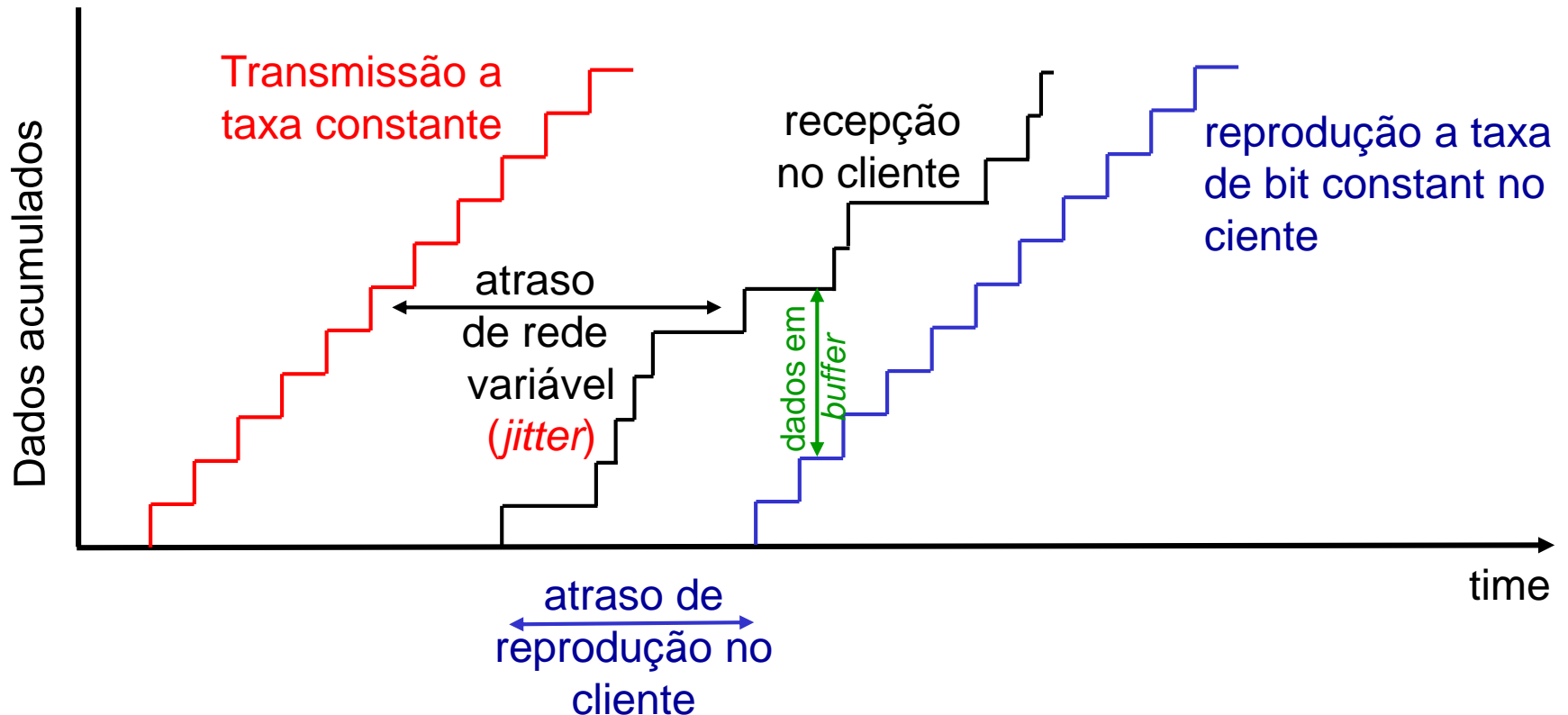
4.4 Protocolos para aplicativos de conversação em tempo real

4.5 Suporte de rede para multimídia

# VoIP: características

- voz: alterna rajadas de fala e períodos de silêncio.
- Situação típica:
  - Pacotes gerados apenas durante rajadas de fala
  - 64 kbps durante rajadas de fala
  - Blocos de 20 ms a taxa de 64 kbps: 160 bytes de dados
- Cabeçalho da camada de aplicação adicionado a cada bloco – **bloco + cabeçalho = pacote**
- Aplicação entrega pacote para socket a cada 20 ms durante rajada de fala
- Pacote de dados encapsulado em segmento UDP ou TCP (por simplicidade vamos considerar UDP – [Skype tenta usar](#))
- Numa rede ideal, pacotes chegam com mesmo atraso e não se perdem: basta receptor reproduzir cada pacote que chega

# Problemas para o VoIP: I) *Jitter* do atraso



- **Jitter:** atraso variáveis entre momentos de transmissão e recepção
- atraso fim-a-fim de dois pacotes consecutivos: diferença pode ser maior ou menor do que 20 ms (diferença no tempo de transmissão)

## Problemas para o VoIP: 2) perda de pacotes

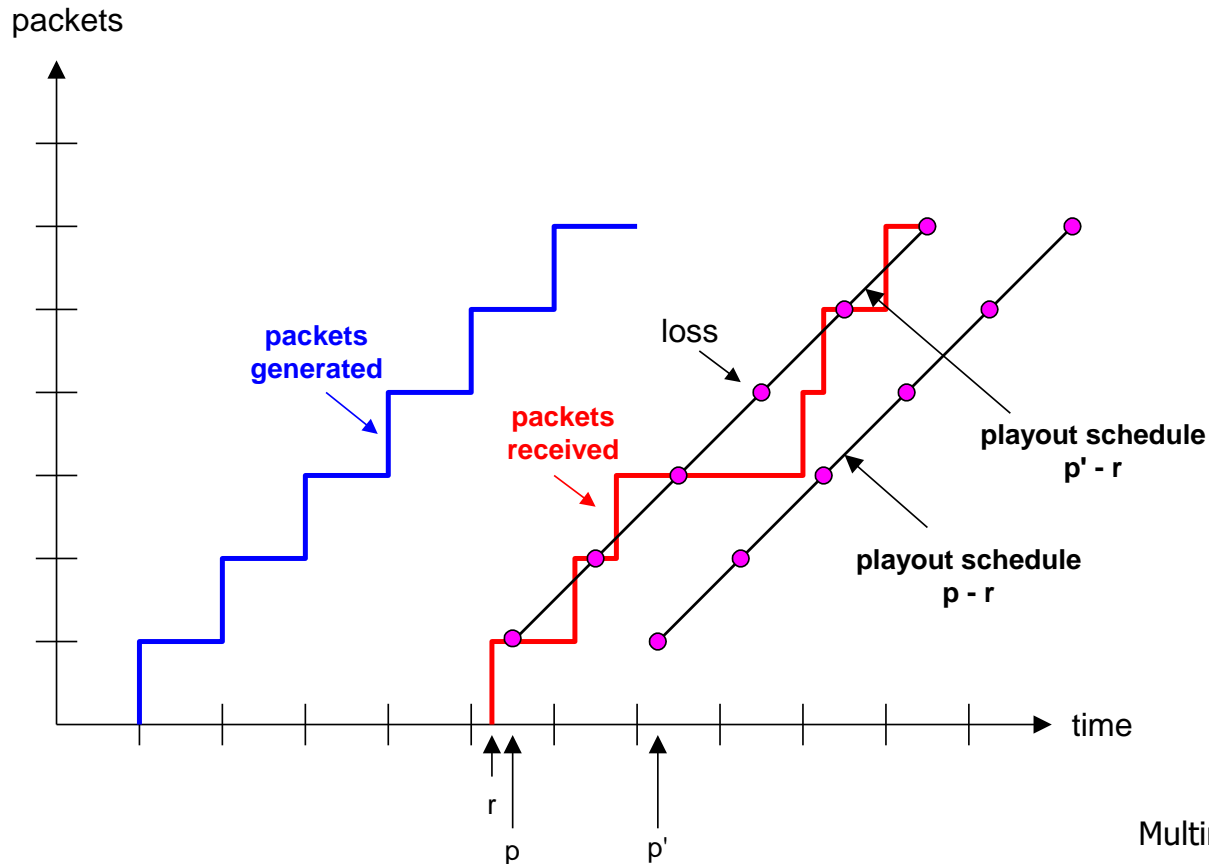
- **Perdas de pacotes na rede:** datagrama IP perdido devido a congestionamento da rede (estouro de buffer de roteador) – retransmissão não resolve...
- **Perdas por atraso:** datagramas IP chegam muito tarde para ser reproduzidas no receptor
  - atrasos: processamento e filas na rede; atrasos nos sistemas finais (transmissor, receptor)
    - < 150 ms não perceptível
    - Entre 150ms e 400 ms - aceitável
    - Acima de 400 ms prejudica muito interatividade da conversa
- **Boa notícia: Tolerância a perdas:** dependendo da codificação da voz e cancelamento da perda, taxas de perda entre de até 20% podem ser toleradas
- Vamos ver como aplicativos resolvem os problemas na sequência... 😊

# VoIP: Atraso de reprodução fixo

- Receptor tenta reproduzir cada bloco exatamente  $q$  ms após o bloco ter sido gerado.
  - cabeçalho do bloco tem marca de tempo  $t$ : receptor reproduz bloco no instante  $t+q$
  - se bloco chega depois de  $t+q$ : dados chegaram tarde demais para reprodução – dados “perdidos”
- compromisso na escolha de  $q$ :
  - $q$  *grande*: menos perdas de pacotes
  - $q$  *pequeno*: melhor experiência interativa
- Ideal abaixo de 150 ms

# VoIP: Atraso de reprodução fixo

- Transmissor gera pacotes a cada 20 ms durante rajada de fala
- Primeiro pacote recebido no instante  $r$
- Exemplo 1) Agendamento de reprodução: começa em  $p$
- Exemplo 2) Agendamento de reprodução: começa em  $p'$



# Atraso de reprodução adaptativo [Ramjee 1994]

- **objetivo:** baixo atraso de reprodução, baixa taxa de perdas por atraso
- **abordagem:** ajuste adaptativo do atraso de reprodução:
  - estima atraso de rede, ajustando atraso de reprodução no começo de cada rajada de voz
  - períodos de silêncio podem ser comprimidos ou alongados devido ao ajuste – não é perceptível
  - blocos ainda reproduzidos a cada 20 ms durante a rajada de fala
- atraso de pacote estimado adaptativamente. (EWMA - *Exponentially Weighted Moving Average*, mesma ideia da estimação do RTT no TCP):

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

*estimativa do atraso após o i-ésimo pacote*

*constante pequena, por exemplo 0,01*

*instante de recepção - instante de envio (marca de tempo no pacote)*

*atraso medido do i-ésimo pacote*



# Atraso de reprodução adaptativo

- Estima-se também o desvio-padrão do atraso,  $v_i$ :

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

- Estimativas de  $d_i$  e  $v_i$  calculadas para todo pacote recebido, mas usado apenas no início de uma rajada de voz
- Para o primeiro pacote na rajada de fala, o instante para reprodução é:

$$\text{tempo-reprodução}_i = t_i + d_i + Kv_i$$

- Pacotes remanescentes na rajada de fala são então reproduzidos periodicamente (a cada 20 ms por exemplo)

# Atraso de reprodução adaptativo

Q: Como o receptor determina se um pacote é o primeiro de uma rajada de voz?

- se não há perdas, receptor olha em marcas de tempo sucessivas
  - se diferença entre marcas sucessivas  $> 20$  ms --> rajada de fala começando.
- com perdas possíveis, receptor precisa olhar tanto marcas de tempo como números sequenciais no cabeçalho do pacote (*não confundir com TCP*)
  - diferença entre marcas sucessivas  $> 20$  ms e números sequenciais sem lacunas --> --> rajada de fala começando.
- Exemplo de pesquisa: [aqui](#)

## VoiP: recuperando-se da perda de pacotes [[IEEE Network](#), [RFC 5109](#)]

**Desafio:** recuperar-se de perda de pacote dado o pequeno atraso tolerável entre a transmissão original e a reprodução

- cada ACK/NAK leva  $\sim 1$  RTT – não adianta para VoIP 😞
- alternativa: **Forward Error Correction (FEC)**
  - ideia: enviar bits suficientes pra permitir recuperação sem retransmissão

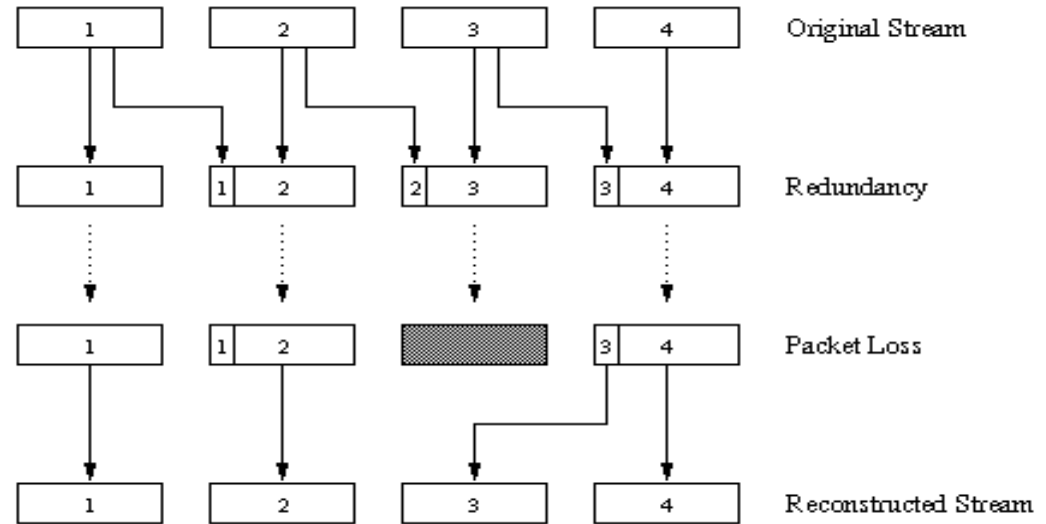
### **FEC simples**

- para cada grupo de  $n$  blocos, cria um bloco redundante fazendo XOR com os  $n$  blocos originais
- envia  $n+1$  blocos, aumentando a largura de banda por um fator  $1/n$
- pode-se reconstruir os  $n$  blocos originais se no máximo 1 bloco se perder entre os  $n+1$  blocos, com atraso para reprodução

# VoiP: recuperando-se da perda de pacotes

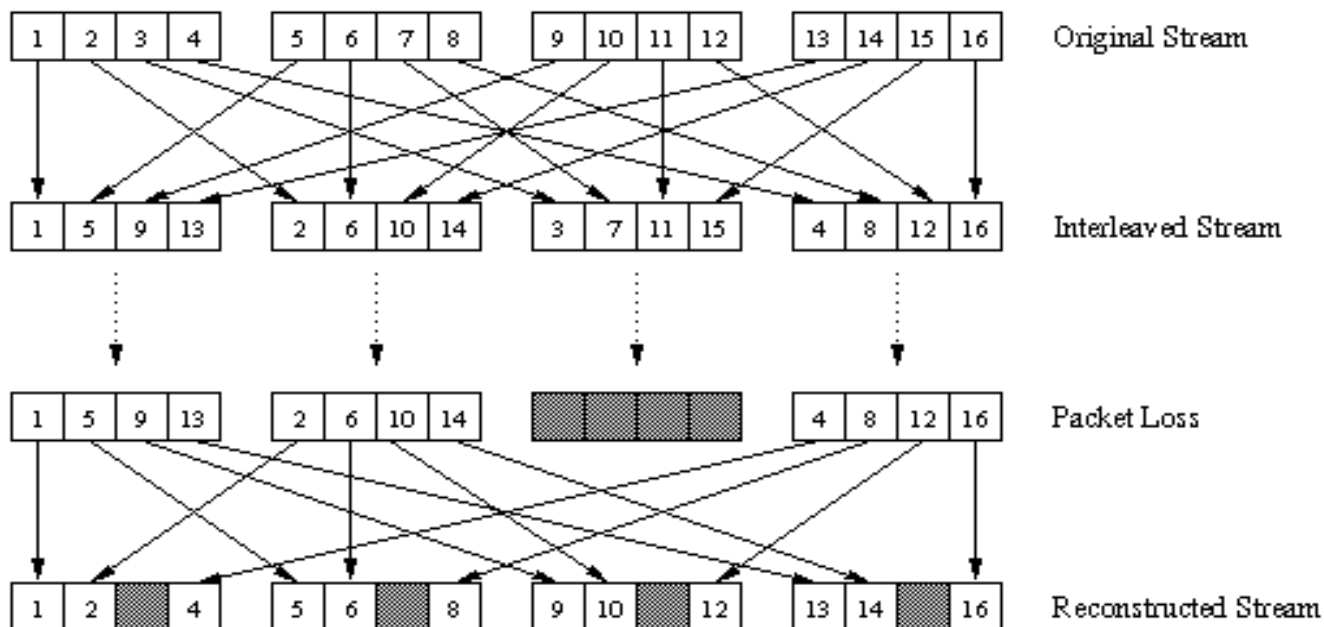
## um FEC mais sofisticado:

- “embutir um fluxo de baixa qualidade na sequência de dados”
- enviar fluxo de áudio de baixa resolução como informação redundante
- por exemplo, fluxo nominal é PCM em 64 kbps e fluxo redundante é [GSM](#) em 13 kbps



- perdas não consecutivas: receptor pode cancelar perda
- generalização: também pode-se anexar bloco (n-1) e (n-2) em baixa taxa no bloco n

# VoIP: recuperando-se da perda de pacotes



## *entrelaçamento para cancelar perda:*

- blocos de áudio divididos em unidade menores, por exemplo, 4 unidades de 5 ms por bloco de áudio de 20 ms
- pacote contém pequenas unidades de diferentes blocos

- se pacote se perder, ainda tem-se *a maior parte* de cada bloco original
- não há aumento de taxa por redundância, mas há aumento no atraso para reprodução – aplicação limitada em VoIP mas muito usada em áudio gravado

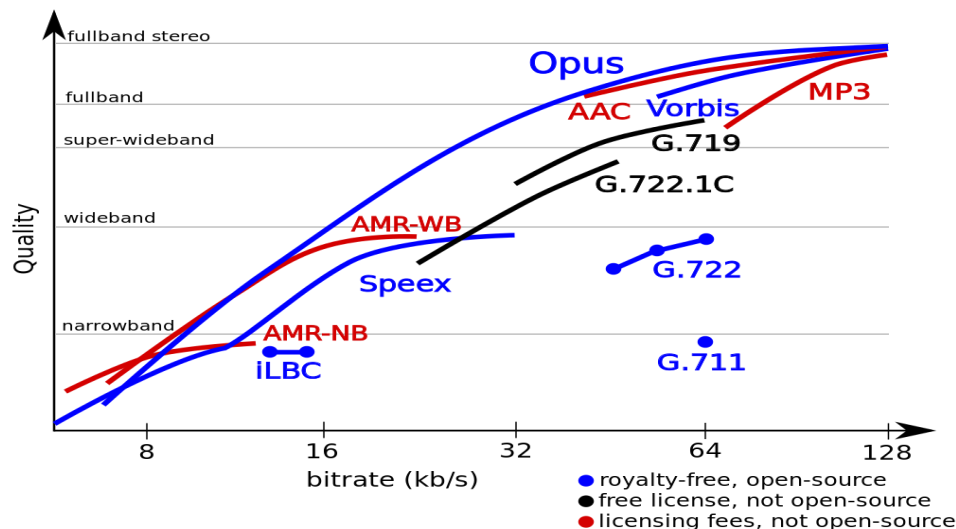
# VoiP: recuperando-se da perda de pacotes

---

- Outras ideias:
  - Repetir bloco perdido
  - Fazer interpolação no trecho de áudio
  - ...
- Tema de pesquisa importante!
- Ver por exemplo artigo da [IEEE Network](#)

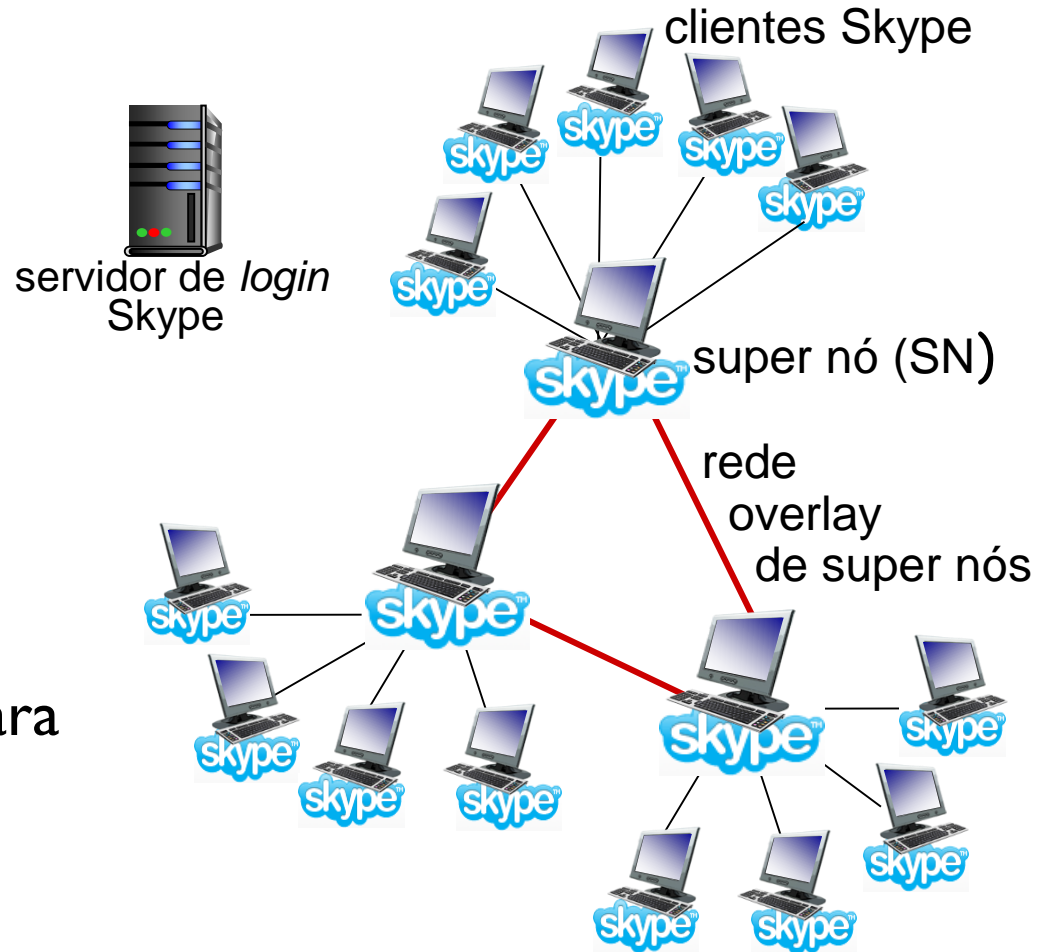
# Voice-over-IP: Skype

- Aplicativo VoIP popular– Chegou a ter 50 milhões de usuários concomitantemente
- Fundado em 2003 e comprado pela Microsoft em 2011
- 40% do tráfego de telefonia internacional em 2014
- protocolo da camada de aplicação proprietário (funcionamento inferido via engenharia reversa, veja por exemplo [[Baset 2006](#)])
  - mensagens criptografadas, mas segurança tem problemas: áudio descriptografado; IP rastreado
- CODEC de áudio [Opus](#); CODEC de vídeo [VP7](#) – taxas variáveis de acordo com qualidade da conexão de 30kbps a 1 Mbps (vídeo)
- Áudio amostrado tipicamente a 16 000 amostras/s – qualidade melhor do que telefonia
- Tenta usar UDP, se não consegue passar vai para TCP



# Voice-over-IP: Skype

- Componentes P2P:
  - **clientes:** *peers* Skype conectam-se diretamente entre si para uma chamada VoIP
  - **super nós (SN):** *peers* Skype com funções especiais
  - **rede overlay:** entre SNs para localizar clientes Skype
  - **servidor de login**

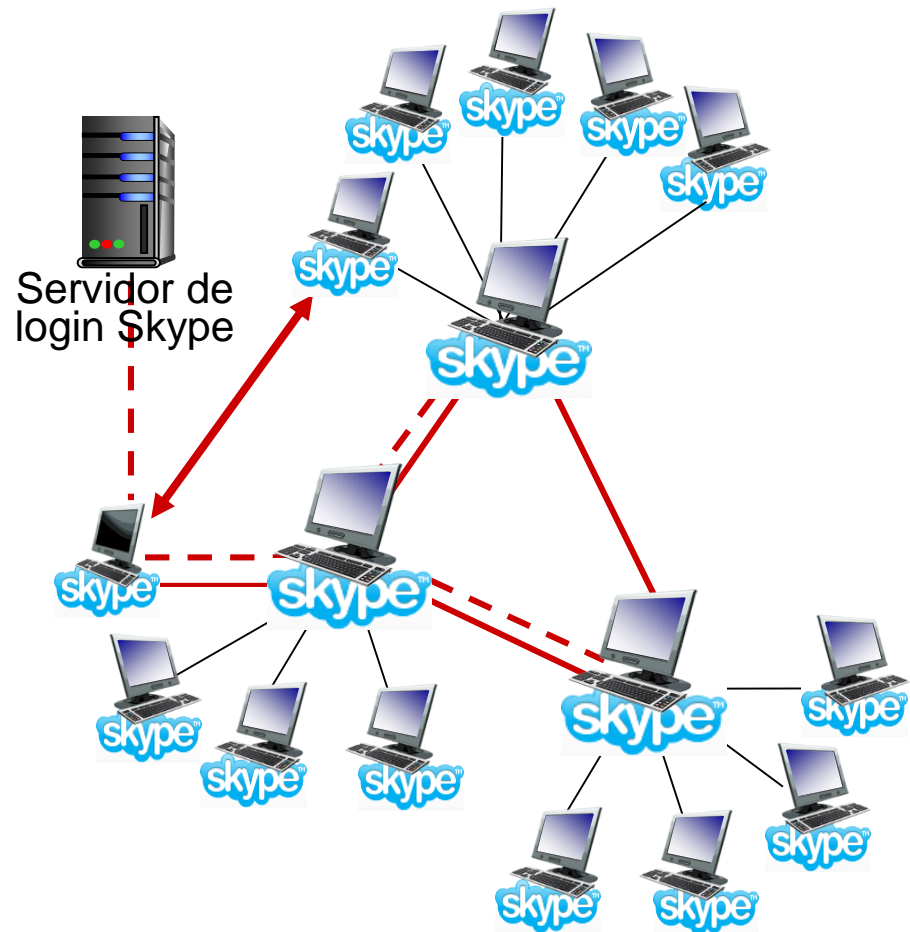




# P2P VoIP: Skype

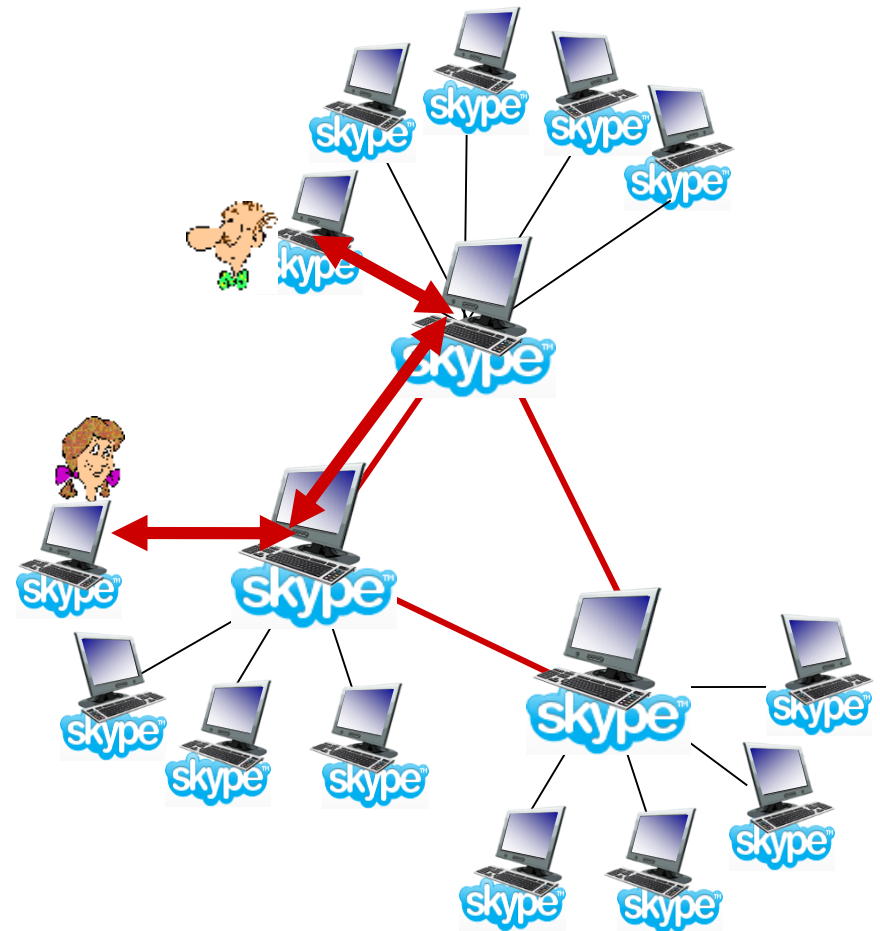
## Operação cliente Skype:

1. junta-se à rede Skype contactando um SN (endereço IP em *cache*) usando TCP
2. loga-se (nome de usuário, senha) a um servidor de *login Skype* centralizado
3. obtém endereço IP do chamado do SN, *overlay* SN
  - ou da lista de contatos do cliente
4. inicia chamada diretamente com destinatário



# Skype: peers como retransmissores

- **problema:** tanto Alice quanto Bob estão atrás de NATs
  - NAT impede que *peer* externo inicie conexão com *peer* interno
  - *peer* interno ao NAT consegue iniciar conexão a *peer* externo
- **solução com retransmissor (relay) :** Alice e Bob mantêm conexão aberta com seus SNs
  - Alice avisa seu SN que quer se conectar a Bob
  - O SN da Alice se conecta ao SN do Bob
  - O SN de Bob se conecta a Bob por meio da conexão aberta que Bob inicialmente começou com seu SN



# Voice-over-IP: Skype

---

- Outras aplicações e ferramentas interessantes do Skype:
  - chamada entre  $N$  usuários
  - videoconferência com múltiplos usuários
- Precisa de técnicas para não sobrecarregar a rede e os *peers*!
- *Mais sobre as ferramentas utilizadas na página 727 do Kurose e referências indicadas lá.*