ACH 2147 — DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO DISTRIBUÍDOS

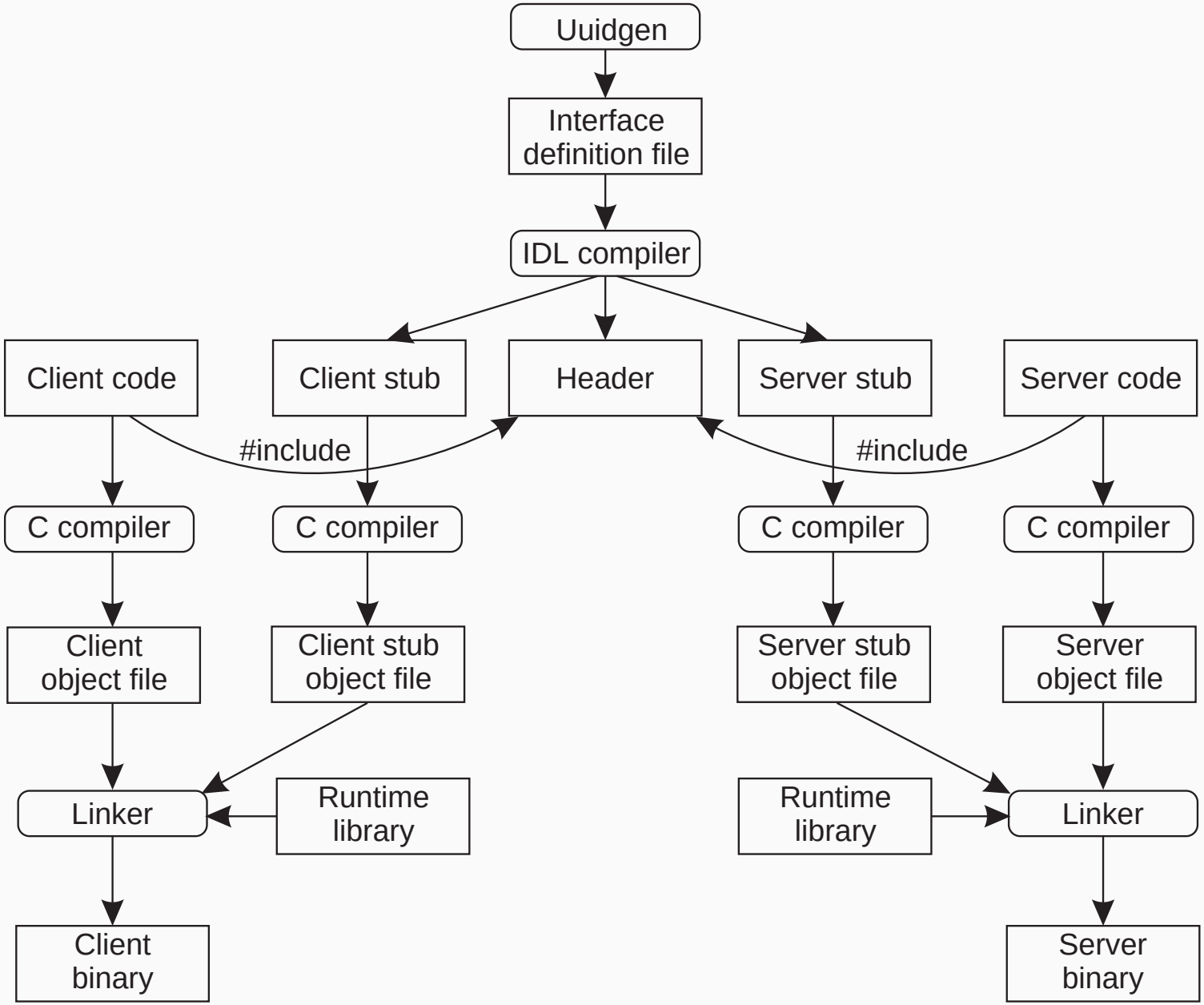
COMUNICAÇÃO

Daniel Cordeiro

25 e 27 de abril de 2018

Escola de Artes, Ciências e Humanidades | EACH | USP

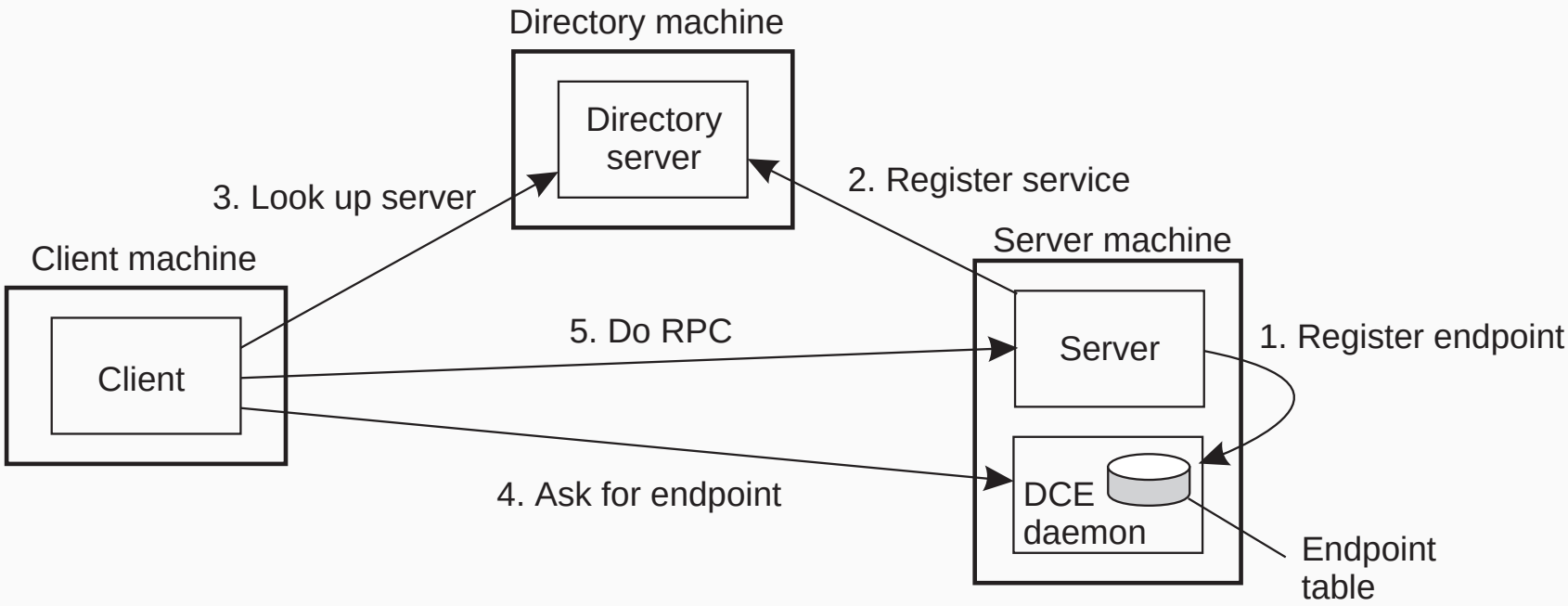
RPC NA PRÁTICA



Problemas

(1) Cliente precisa localizar a máquina com o servidor e,

(2) precisa localizar o servidor.



Comunicação orientada a mensagens

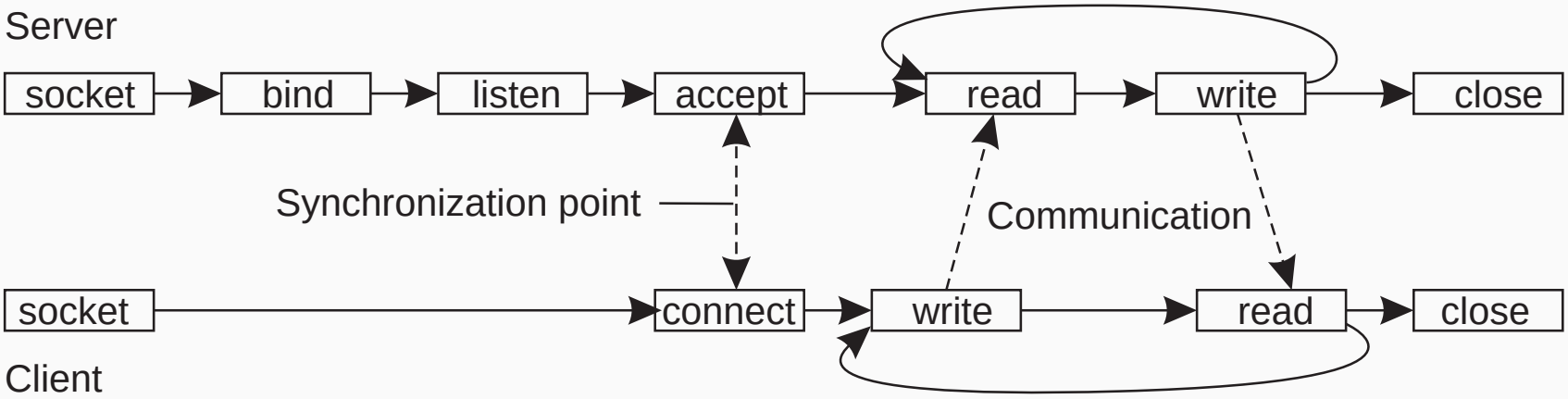
COMUNICAÇÃO ORIENTADA A MENSAGENS

- Mensagens transientes
- Sistema de enfileiramento de mensagens
- *Message brokers*
- Exemplo: IBM Websphere

Berkeley socket interface

`SOCKET`	Cria um novo ponto de comunicação
`BIND`	Especifica um endereço local ao socket
`LISTEN`	Anuncia a vontade de receber *N* conexões
`ACCEPT`	Bloqueia até receber um pedido de estabelecimento de conexão
`CONNECT`	Tenta estabelecer uma conexão
`SEND`	Envia dados por uma conexão
`RECEIVE`	Recebe dados por uma conexão
`CLOSE`	Libera a conexão

MENSAGENS TRANSIENTES: SOCKETS



SOCKETS: CÓDIGO EM PYTHON

```
import socket
HOST = socket.gethostname()             # e.g. 'localhost'
PORT = SERVERPORT                       # e.g.  80
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(N)          # listen to max N queued connection
conn, addr = s.accept()       # new socket + addr client
while 1: # forever
  data = conn.recv(1024)
  if not data: break
  conn.send(data)
conn.close()
```

Ideia geral

Comunicação assíncrona e persistente graças ao uso de filas pelo middleware. Filas correspondem a buffers em servidores de comunicação.

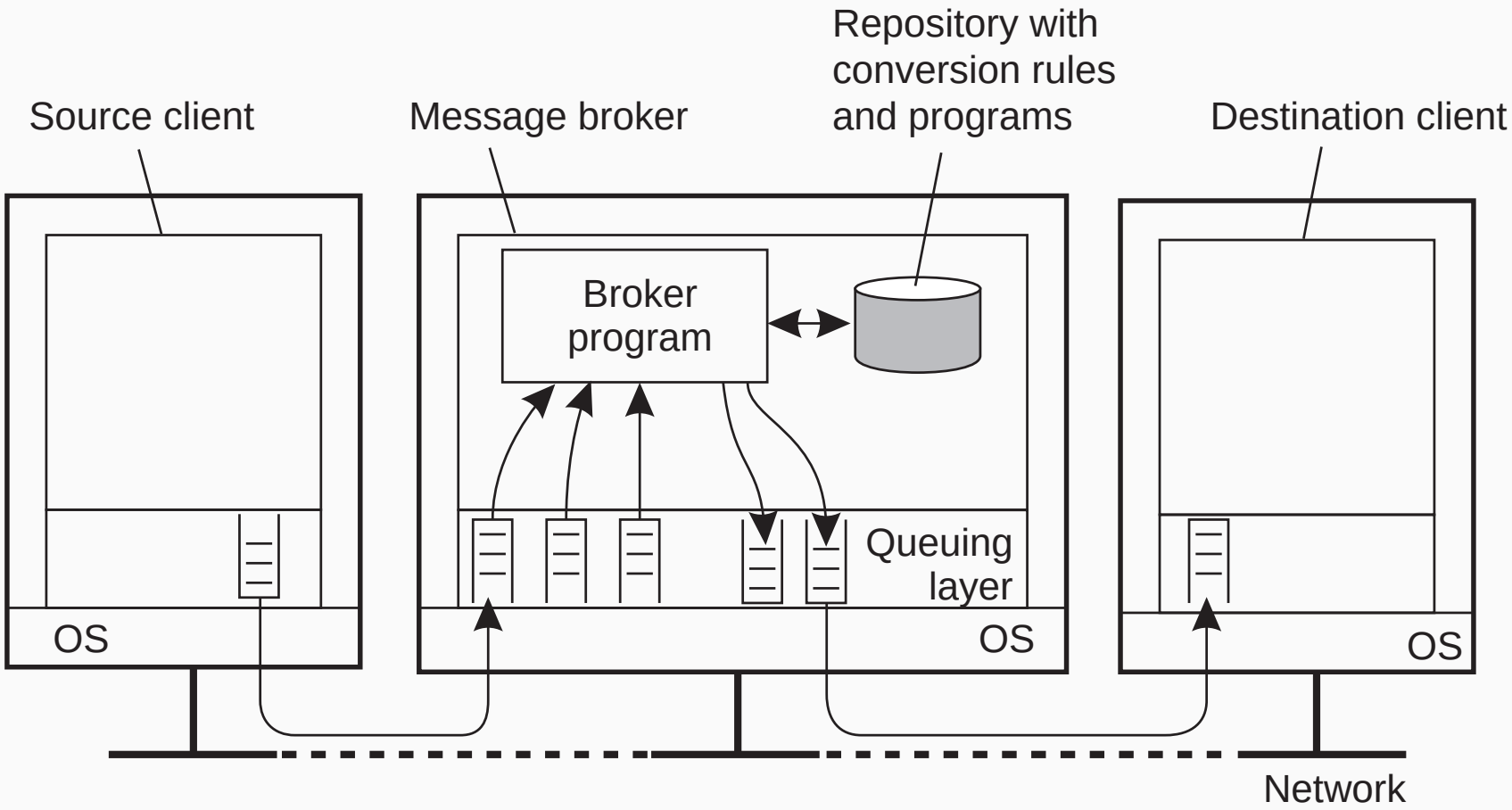
`PUT`	Adiciona uma mensagem à fila especificada
`GET`	Bloqueia até que a fila especificada tenha alguma mensagem e remove a primeira mensagem
`POLL`	Verifica se a fila especificada tem alguma mensagem e remove a primeira. Nunca bloqueia
`NOTIFY`	Instala um tratador para ser chamado sempre que uma mensagem for inserida em uma dada fila

Observação:
Sistemas de filas de mensagens assumem um protocolo comum de troca de mensagens: todas as aplicações usam o mesmo formato de mensagem (i.e., estrutura e representação de dados)

Message broker
Componente centralizado que lida com a heterogeneidade das aplicações:

- transforma as mensagens recebidas para o formato apropriado
- frequentemente funciona como um application gateway
- podem rotear com base no conteúdo $\Rightarrow$ Enterprise Application Integration

MESSAGE BROKER



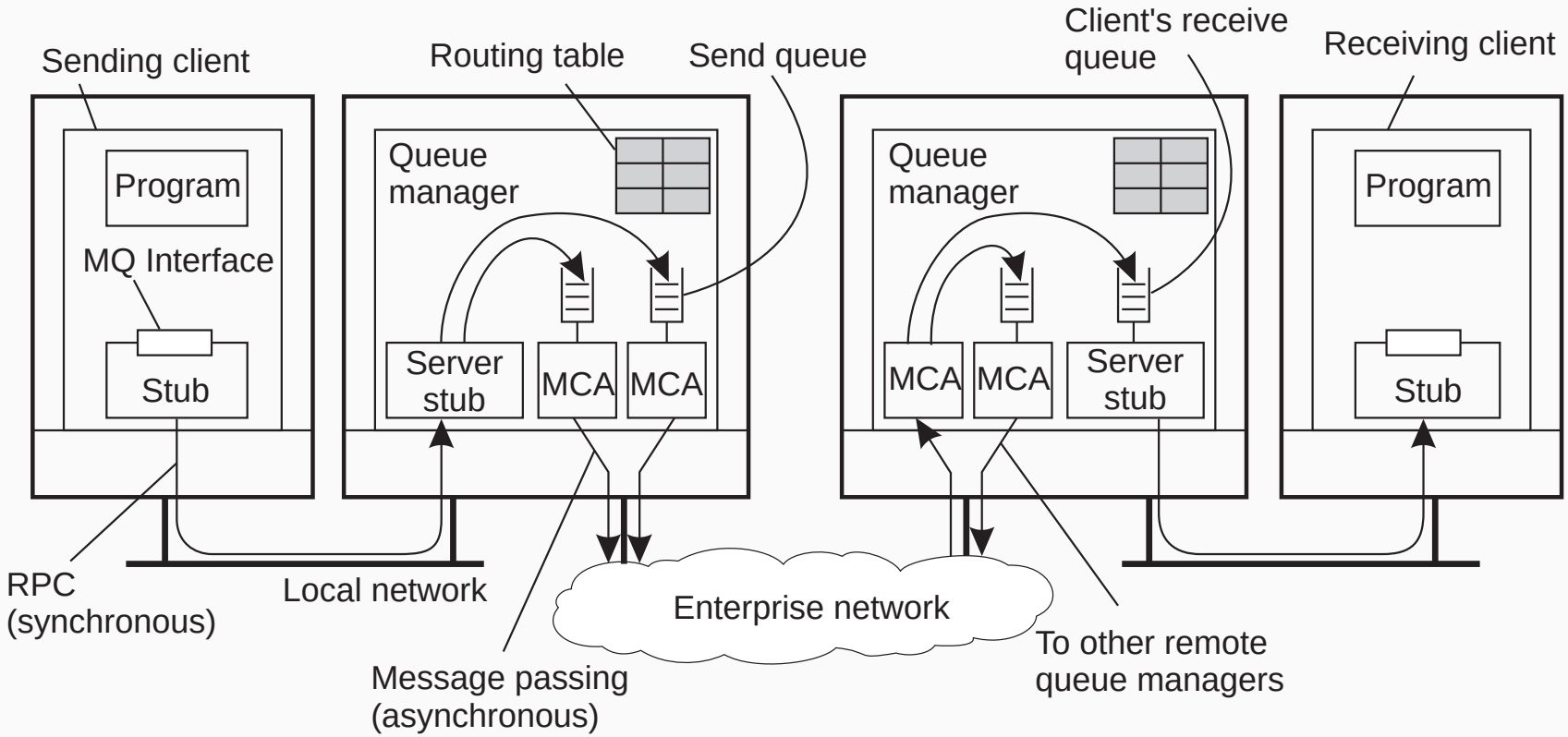
IBM WEBSPHERE MESSAGE QUEUE (MQ)

- Mensagens específicas da aplicação são colocadas e removidas de filas
- As filas são controladas por um gerenciador de filas
- Processos podem colocar mensagens apenas em filas locais, ou usando um mecanismo de RPC

Transferência de mensagens

- Mensagens são transferidas entre filas
- Mensagens transferidas entre filas em diferentes processos requerem um canal
- Em cada ponta do canal existe um agente de canal, responsável por:
  - configurar canais usando ferramentas de rede de baixo nível (ex: TCP/IP)
  - (Des)empacotar mensagens de/para pacotes da camada de transporte
  - Enviar/receber pacotes

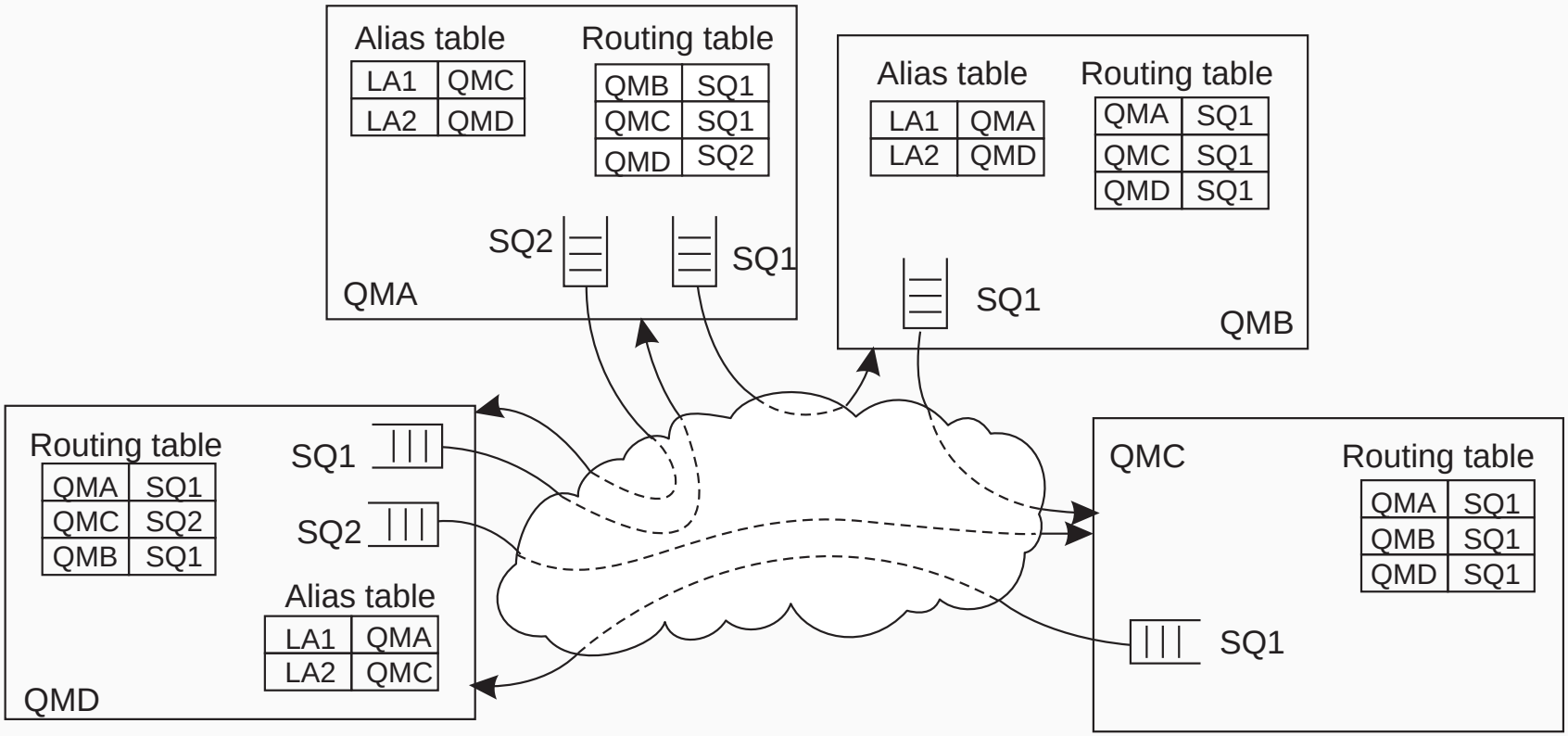
IBM WEBSPHERE MQ



- Canais são unidirecionais
- Agentes de canais são automaticamente iniciados quando uma mensagem chega
- Pode-se criar redes de gerenciadores de filas
- Rotas são configuradas manualmente (pelo admin do sistema)

Roteamento

O uso de nomes lógicos, combinados com resolução de nomes para filas locais, permitem que uma mensagem seja colocada em uma fila remota.



CONEXÃO ORIENTADA A FLUXO DE DADOS

- Apoia a distribuição de mídia contínua
- Fluxos em sistemas distribuídos
- Gerenciamento de fluxo

Observação
Todas as ferramentas de comunicação discutidas até agora são essencialmente discretas, isso é, úteis para trocas de informação de forma independente do tempo

Mídia contínua
É caracterizada pela dependência temporal de seus valores:

- Áudio
- Vídeo
- Animações
- Dados de sensores (temperatura, pressão, etc.)

Modos de transmissão

Diferentes garantias temporais relacionadas à transferência de dados:

Assíncrono: sem restrições sobre **quando** o dado deve ser entregue

Síncrono: define um atraso máximo para a entrega de cada um dos pacotes

Isócrono: define um máximo e um mínimo para o atraso (o *jitter* é limitado).

Definição
Um fluxo (contínuo) de dados é uma ferramenta de comunicação orientada a conexão que oferece transmissão isócrona de dados.

Características comuns de fluxos

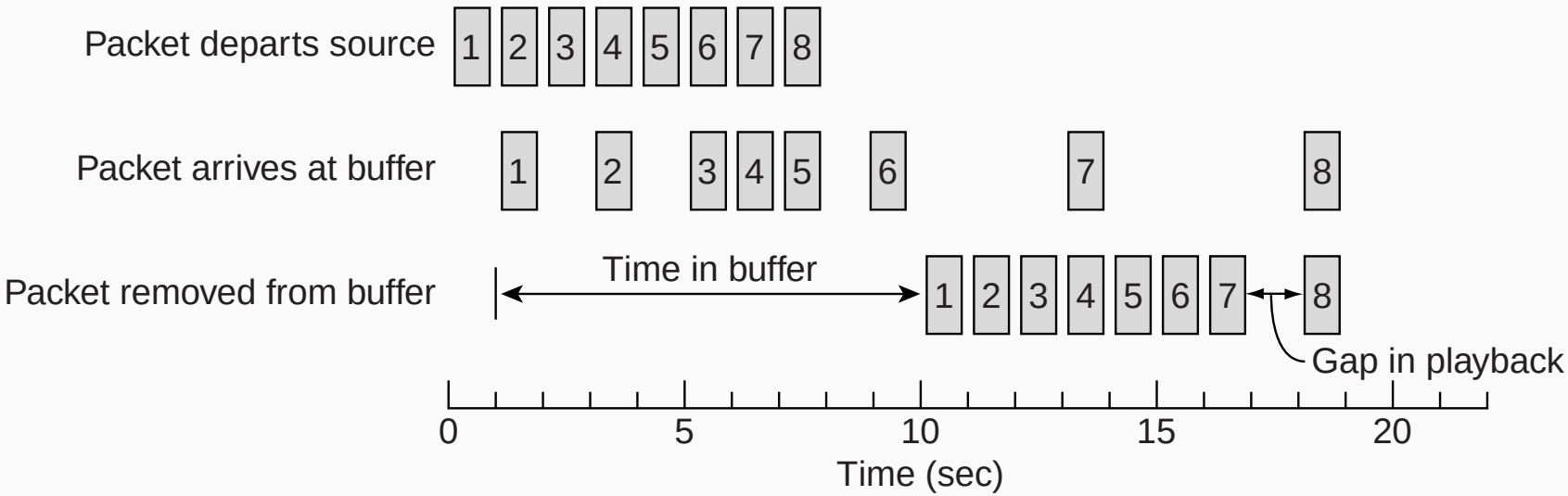
- Fluxos são unidirecionais
- Geralmente há uma única origem e um ou mais destinos (*sinks*)
- Frequentemente o destino e/ou origem encapsulam algum hardware (ex: câmera, monitor, etc.)
- Fluxo simples: um único fluxo de dados (ex: áudio ou vídeo)
- Fluxo complexo: múltiplos fluxos de dados (ex: áudio estéreo, combinações de áudio/vídeo, etc.)

Como especificar Qualidade de Serviço (QoS) em fluxos de dados, que precisam ser entregues no tempo certo? Métricas comuns:

- o bit rate no qual os dados devem ser transportados
- o atraso máximo até que a sessão tenha sido configurada (i.e., quando a aplicação pode começar a enviar os dados)
- o atraso máximo de ponta a ponta (i.e., quanto tempo até que o dado chegue ao seu destinatário)
- O atraso máximo de ida e volta (*round-trip delay*)
- A variação máxima do atraso de ida e volta, o *jitter*

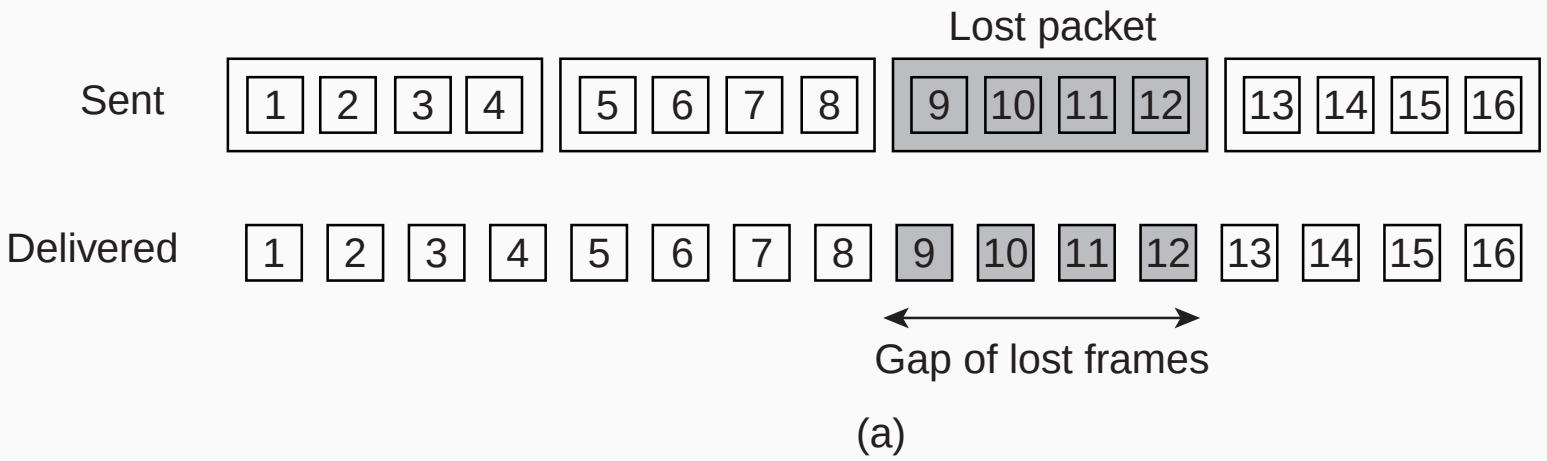
Existem algumas ferramentas do nível de rede, tais como os serviços diferenciados pelos quais alguns pacotes podem ser priorizados.

Além disso:
Use buffers para reduzir o *jitter*.

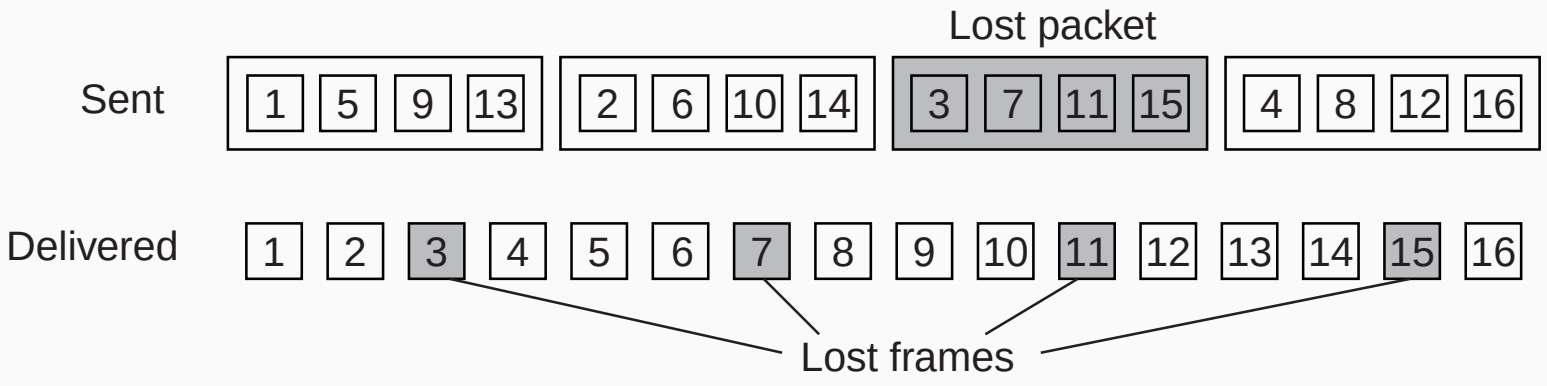


Problema

Como reduzir os efeitos de perdas de pacotes (quando múltiplas amostras são enviadas em um único pacote)?


Lost packet
Sent
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Delivered
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Gap of lost frames

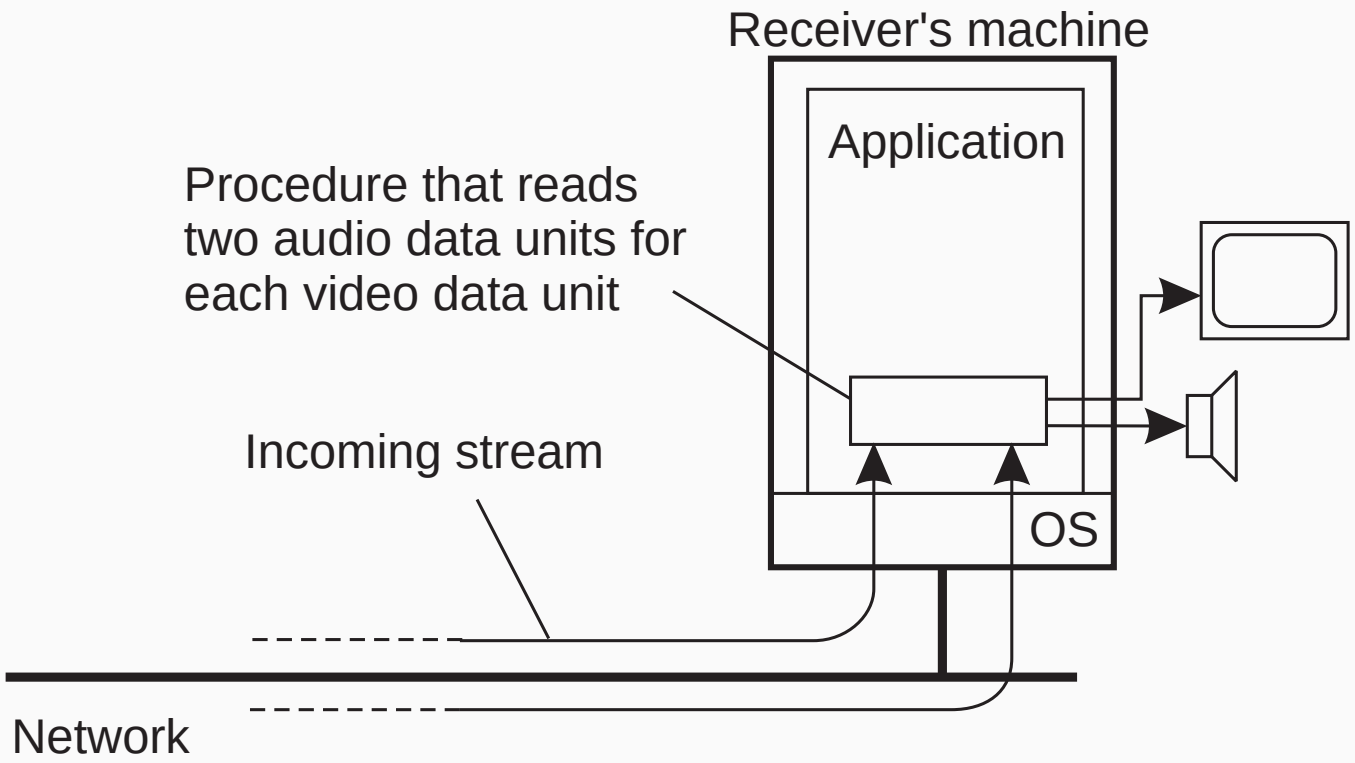
(a)


Lost packet
Sent
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16
Delivered
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Lost frames

(b)

Problema
Em um fluxo complexo, como manter seus diferentes fluxos sincronizados?

Exemplo
Imagine tocar dois canais, que juntos formam um som estéreo.
Diferença deve ser menor do que 20–30 μ segundos!



Alternativa

Multiplexar todos os fluxos em um único fluxo, e depois demultiplexá-los no destinatário. Sincronização é tratada nos pontos de multiplexação e demultiplexação (MPEG).