软件架构与中间件

涂志莹
tzy_hit@hit.edu.cn

哈尔滨工业大学

软件架构与中间件

Software Architecture and Middleware

第3章
计算层的软件架构技术

第3章 计算层的软件架构技术

3.1 计算层的软件架构技术挑战

3.2 分布式编程模型

3.3 负载均衡

3.4 消息队列

3.5 分布式服务框架

3.4 消息队列

回顾

- In a distributed system, processes
  - run on different machines
  - exchange information through message passing
- Successful distributed systems depend on communication models that hide or simplify message passing

成功的分布式系统依赖于隐藏或
简化消息传递的通信模型

进程间通信

Interprocess Communication

- Modern distributed systems consist of thousands or even millions of processes scattered across a network such as the Internet 分布在网络中的多进程
  - To study distributed systems we need to examine how processes on different machines exchange information
  - Challenge: the underlying network is unreliable!

底层的网络是不可靠的

- **Interprocess communication** is at the heart of all distributed systems

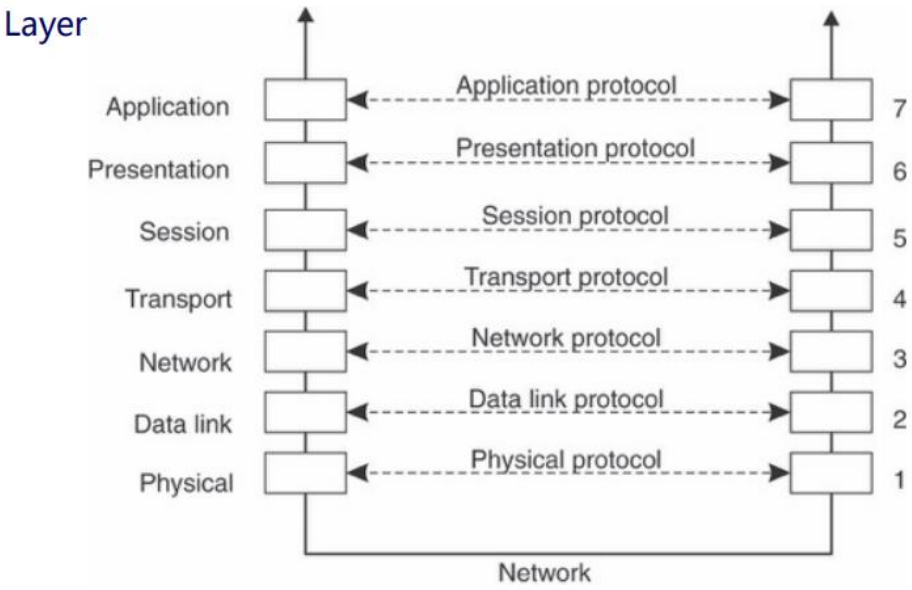
进程间通信

Interprocess Communication

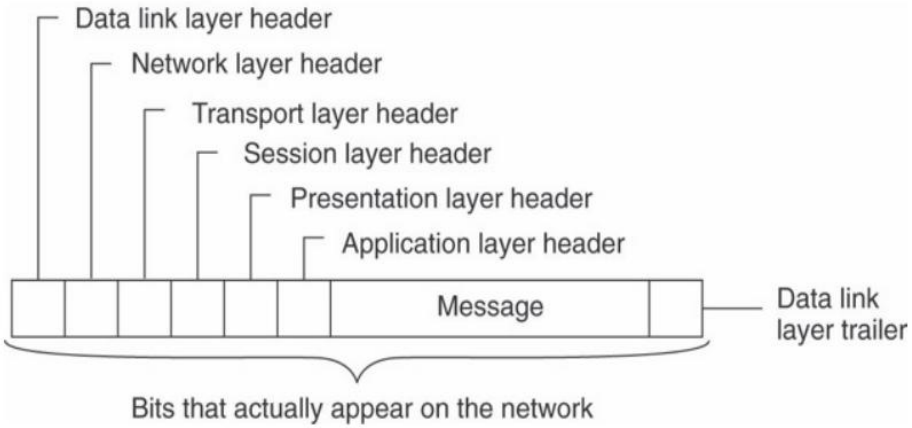
- How can processes on different remote machines exchange information?
  - No primitives based on shared memory (like in non-distributed parallel systems) 没有基于共享内存的原语
- Communication in distributed systems
  - Based on **low-level message passing** 基于低层级的消息传递
  - Offered by the underlying network 由底层网络支撑
  - Harder than using shared memory 比用共享内存更难

分层协议：OSI模型

Layered Protocols: OSI Model



A typical message as it appears on the network.



通信模型

Widely used **models for communication**:

- Remote Procedure Call (RPC)
  - Hide most of the intricacies of message passing
  - Ideal for client/server applications

隐藏了大多数复杂的信息传递

理想的客户端/服务器应用程序

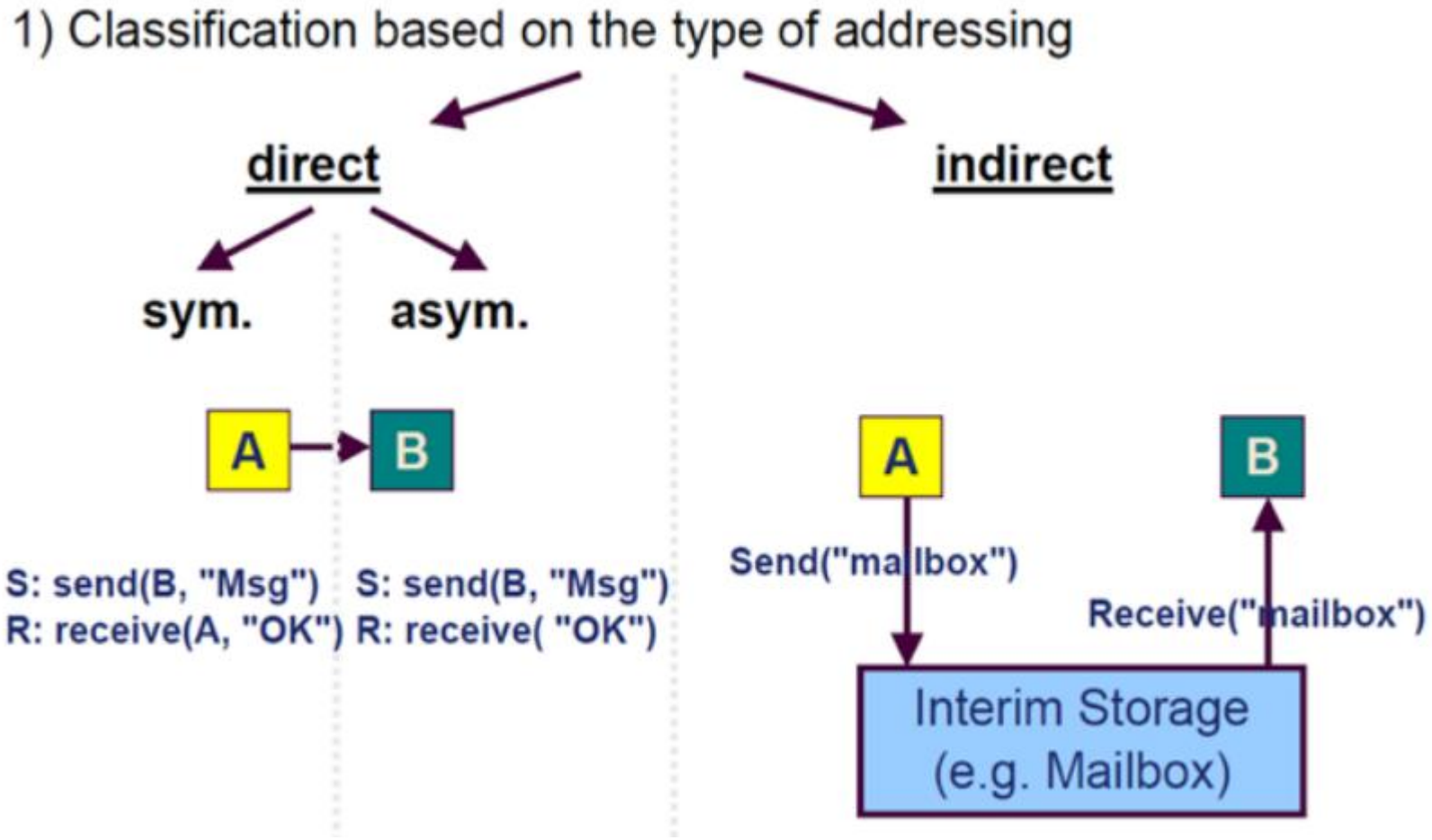
- Message-Oriented Middleware (MOM)
  - High-level message queuing model, similar to email
  - Communication does not follow the rather strict pattern of client/server interaction.

高级消息排队模型，类似于电子邮件

通信并不遵循相当严格的客户机/服务器交互模式。

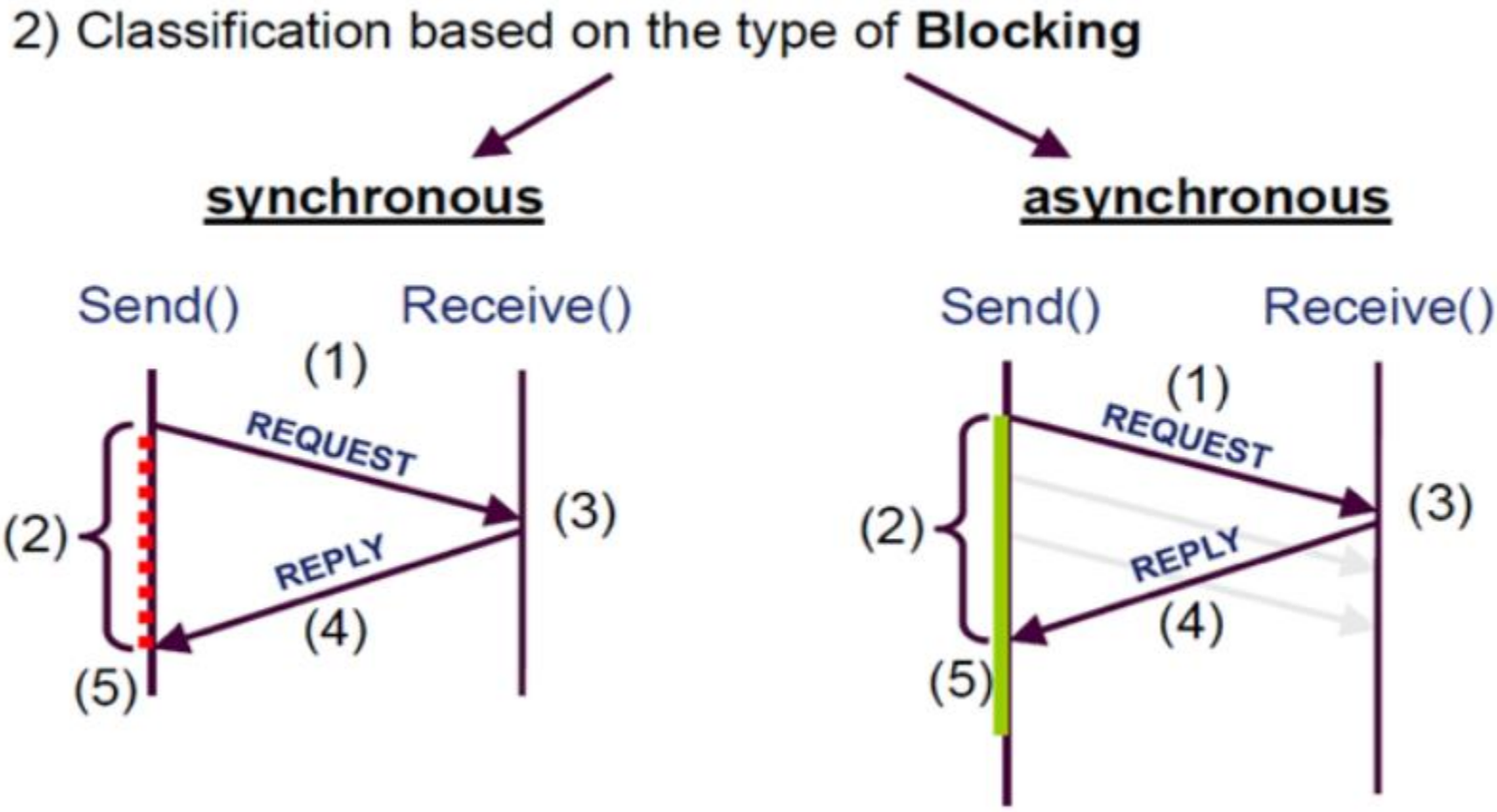
通信模型

基于寻址类型的分类



通信模型

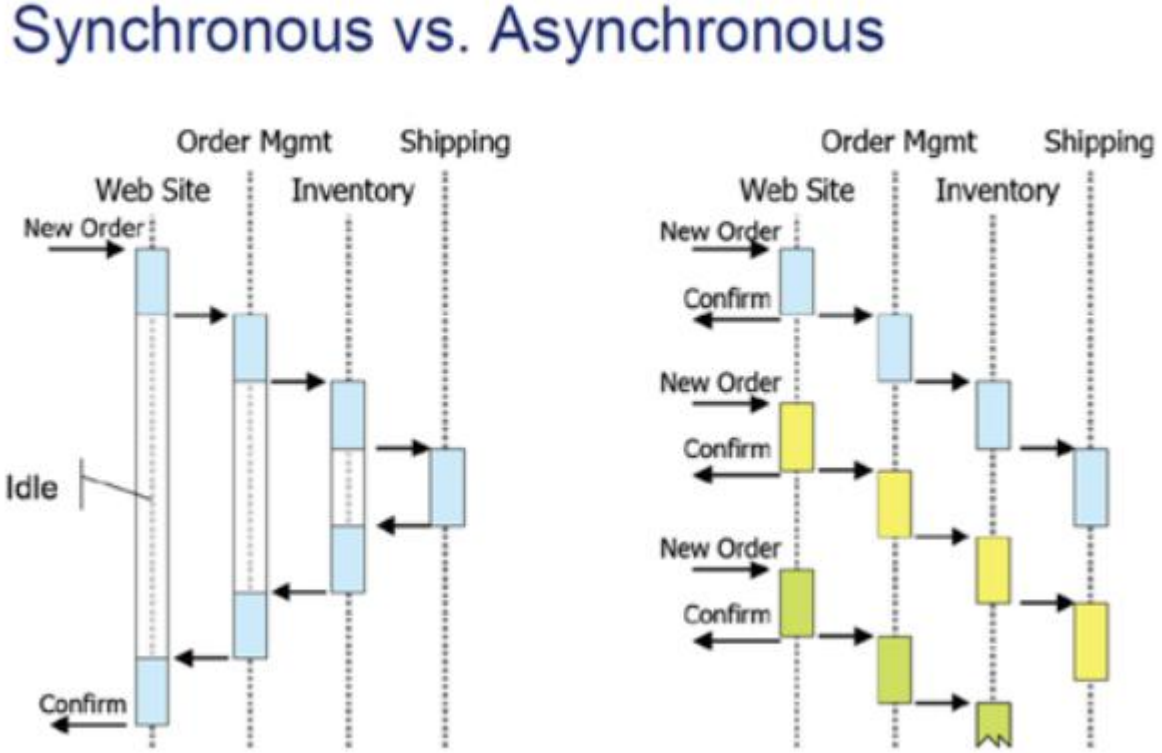
基于阻塞类型的分类



通信模型

基于阻塞类型的分类

- 示例



通信模型

基于缓存类型的分类

3) Classification based on the type of **Buffering**

not buffered

(transient) 瞬态

Messages are immeadiately delivered.

Server has to run at the begin of communication.

Direct connection to the server necessary.

buffered

(persistent) 持久

Messages can be buffered.

通信模型

基于内容类型的分类

4)Classification based on type of **content**:

- Events
- Commands
- Data
- Streams

通信模型

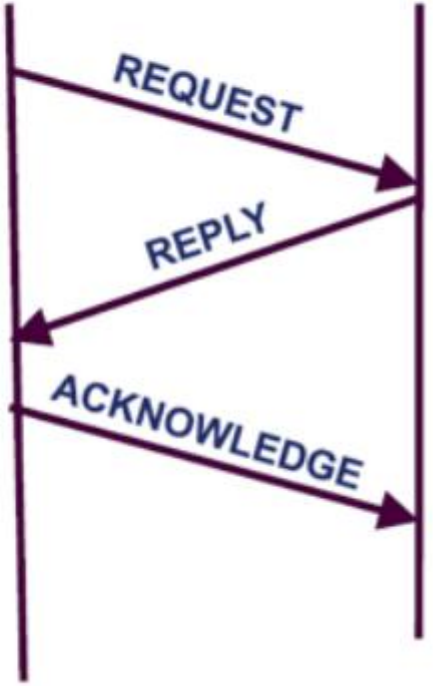
基于确认类型的分类

5) Classification based on the type of **Confirmation**

a) unconfirmed

b) confirmed

c) 3-Way-Handshake



通信模型

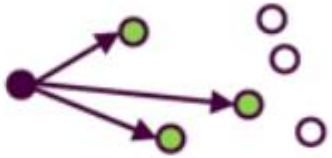
基于接收节点数的分类

6) Classification based on the number of receivers

a) Point to Point (Unicast)

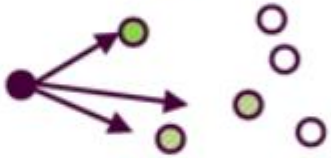
点对点

b) Multicast



多播

c) Anycast

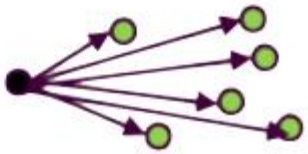


任播

d) Geocast

地域性群播

e) Broadcast



广播

通信模型

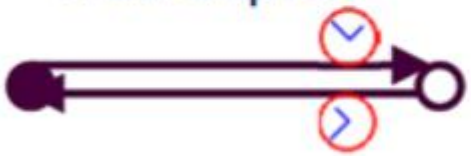
基于通讯方向的分类

7) Classification based on the direction of communication

a) Unidirectional: Simplex 单向

b) Bidirectional

- Half duplex 双向半双工



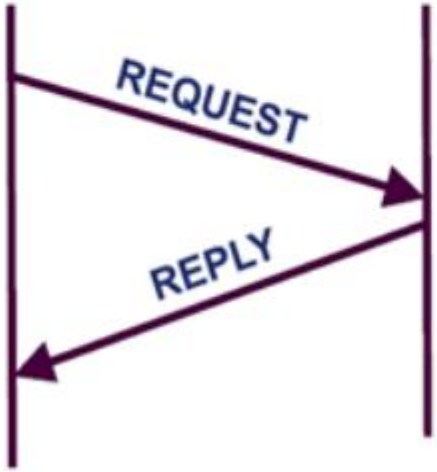
- Full duplex (bidirectional) 双向全双工

通信模型

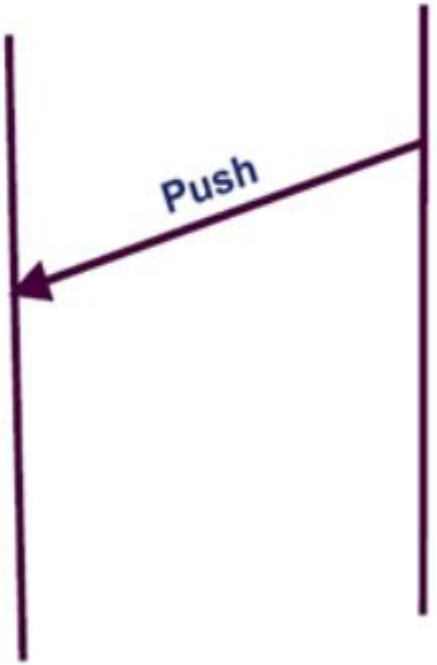
基于发起方的分类

8) Classification based **Initiation**

a) Client-initiated
Pull 拉取



b) Server-initiate
Push 推送



通信模型

基于消息存储的分类

9）Classification based on Storage

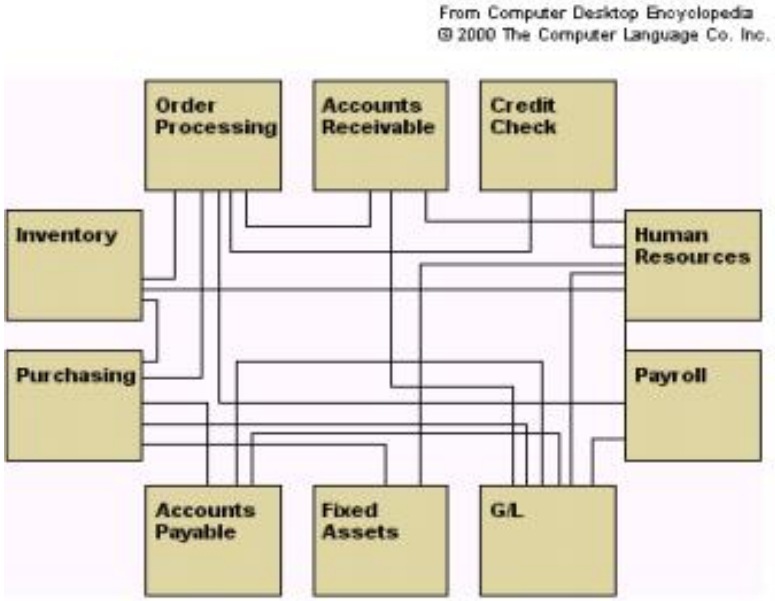
- Persistent communication 持续通信
  - Message stored by the communication middleware **as long as it takes to deliver it** 由通信中间件存储的消息，只要它需要传递它
  - Receiving application need not be executing when the message is submitted. Receiver will get message next time it runs.
  - Example: **email systems** 收发双方无需同步，接收器将在下次运行时得到消息。
- Transient communication 瞬态通信
  - Message is stored by the communication system only **as long as the sending and receiving applications are executing.**
  - If the middleware cannot deliver a message due to a transmission interrupt, or because the recipient is currently not active, **the message will simply be discarded**
  - Example: **transport-level communication services** (if the router can't deliver, it drops the message) 传输级通信服务

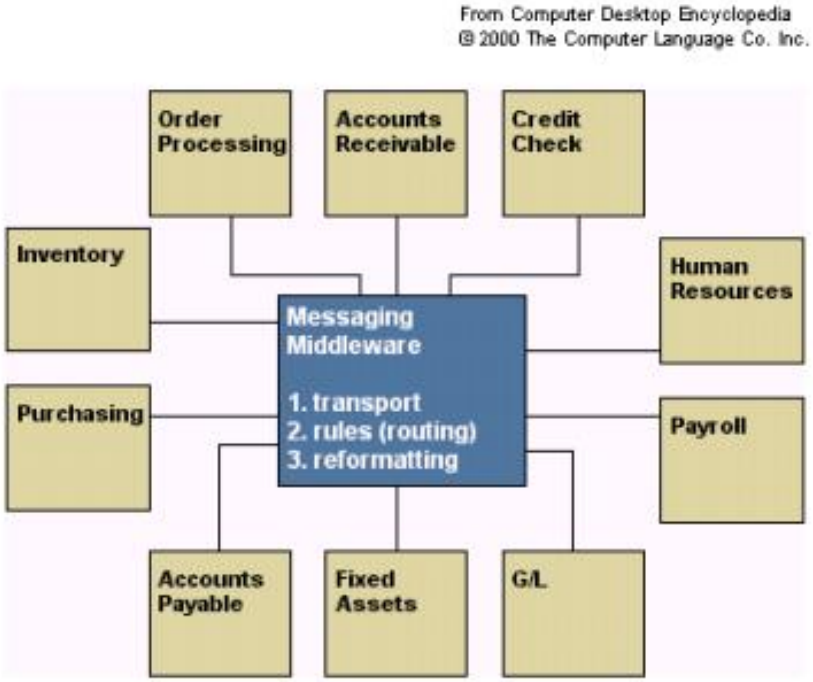
收发双方均运行时，才会存储消息

中间件传输中断或收件人无法传递消息，则丢弃该消息

面向消息的中间件

Message Oriented Middleware (or "MOM") is one particular form of middleware, which is capable of facilitating the transportation of asynchronous messages from one component to another.

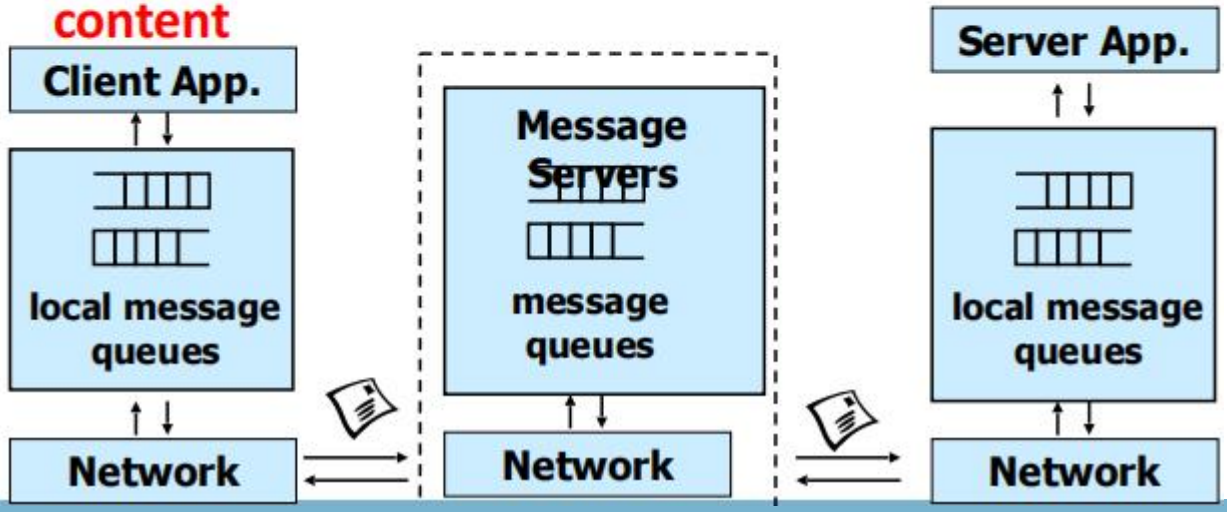




面向消息的中间件

- 基于消息的通信
- 消息存储在消息队列里
- 消息服务器解耦了客户端和服务端
- 关于消息内容的各种假设

- Communication using **messages**
- Messages stored in **message queues**
- **Message servers** decouple client and server
- Various assumptions made about **message content**



MOM的形式

Forms of MOM:

- Message Queuing
- Publish-Subscribe

- 消息排队
- 发布-订阅

面向消息的持久性通信

Message-Oriented Persistent Communication

Message Queuing Systems or **Message-Oriented Middleware (MOM)**

- Important class of message-oriented middleware services
- Persistent asynchronous communication
- Offer intermediate-term storage capacity for messages
- No need for either the sender or receiver to be active during message transmission
- Support message transfers that are allowed to take minutes.

- 面向消息的中间件服务的重要类别
- 持久异步通信
- 为消息提供中间形态的存储容量
- 不需要发送方或接收方在消息传输期间保持活跃状态
- 支持允许消息传输的时间开销。

MOM的属性

Asynchronous interaction 异步交互

- Client and server are only **loosely coupled**
- Messages are queued
- Good for application integration

Support for **reliable** delivery service 可靠服务交付

- Keep queues in persistent storage

Processing of messages by intermediate message server(s) 通过中间消息服务器处理消息

- May do filtering, transforming, logging, ...
- Networks of message servers

Natural for database integration 支持数据库集成

消息队列模型

- **Basic idea**: applications communicate by inserting messages in specific queues
- Messages forwarded over a series of communication servers before being finally being delivered to the destination
- Receiver can be down when message was sent
- No need for the sender to be executing when a message is picked up by the receiver
- Sender is offered the **guarantee that its message will eventually be inserted into the recipient's queue,** but **no guarantee about when**

应用程序通过在特定的队列中插入消息来进行通信

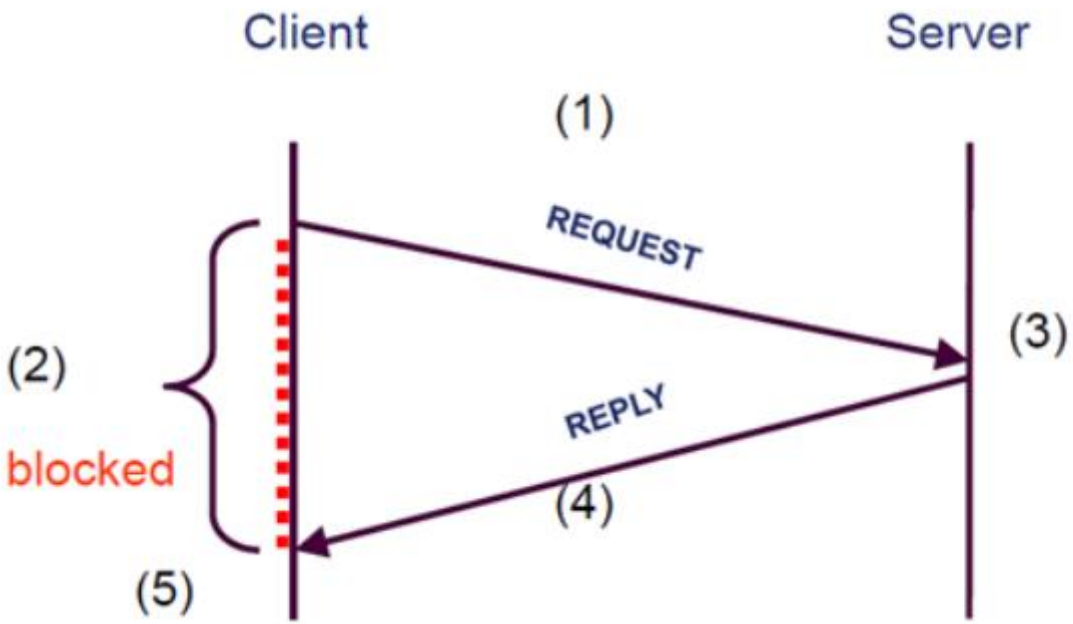
在最终被传递到目的地之前，通过一系列通信服务器转发的消息

接收双方无需同步

发送者可以保证其消息最终将被插入到收件人的队列中，但不能保证何时插入

同步通信

Synchronous Communication



同步通信

Synchronous Communication

Also known as 'Rendevouz'

- **Tightly coupled**

Pros:

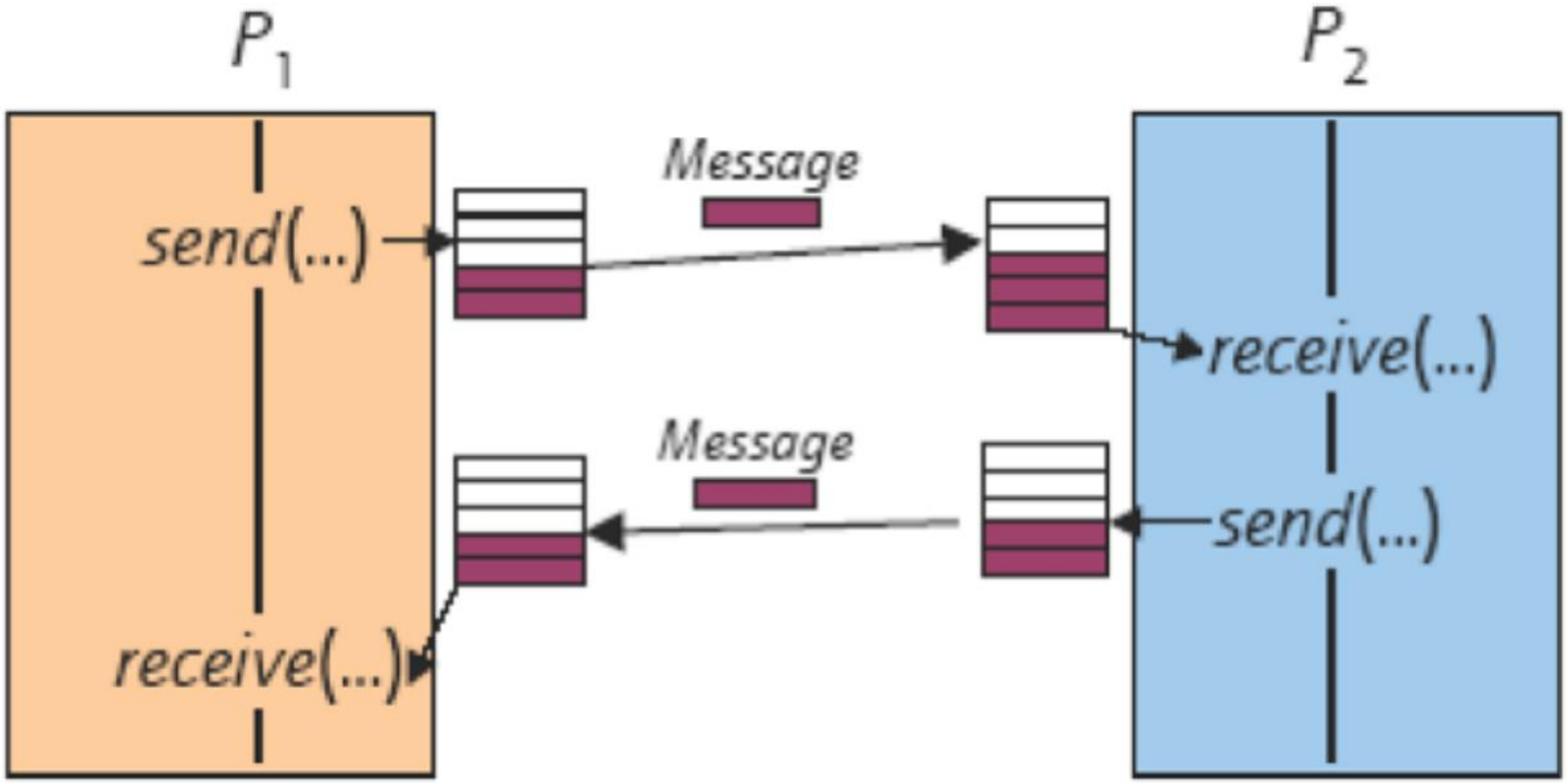
- Client is informed about message delivery
- No buffering necessary

Cons:

- Sender and receiver have to run on the same time
- Direct connection necessary
- Client is blocked
- Lower degree of parallelity

异步（持久）通信

Asynchronous (Persistent) Communication



异步（持久）通信

Asynchronous (Persistent) Communication

- **Loosely coupled**

Pros:

- Higher degree of parallelity
- Enables faster sending than transmitting
- May compensate speed deviation between sender and receiver (but not speed differences) 可以补偿发送方和接收方之间的速度偏差（但不补偿速度差异）
- Sender and receiver have not to run on the same time

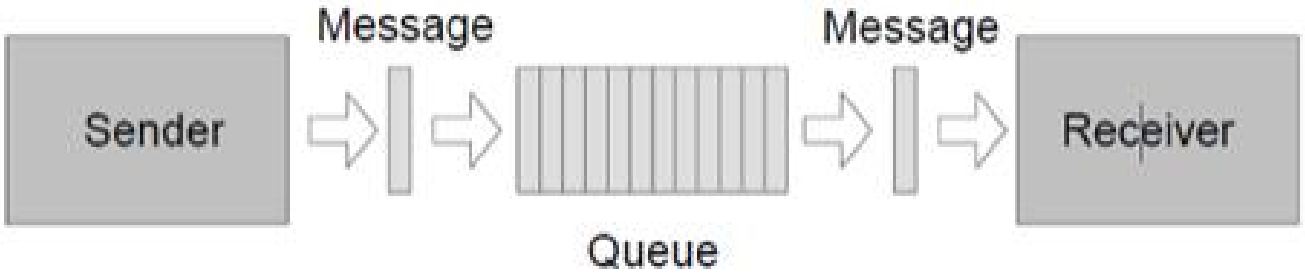
Cons:

- Complicates synchronisation
- Sender has no information about message delivery
- Possible buffer overflows 存在缓冲区溢出的风险

异步（持久）通信

Asynchronous (Persistent) Communication

- Asynchronous data transport between processes
- Data is transmitted in messages
- Based on queues
  - ➢ Sender puts messages into a queues
  - ➢ Message is transmitted to receiver
  - ➢ Receiver reads message from queues



消息队列

Queues

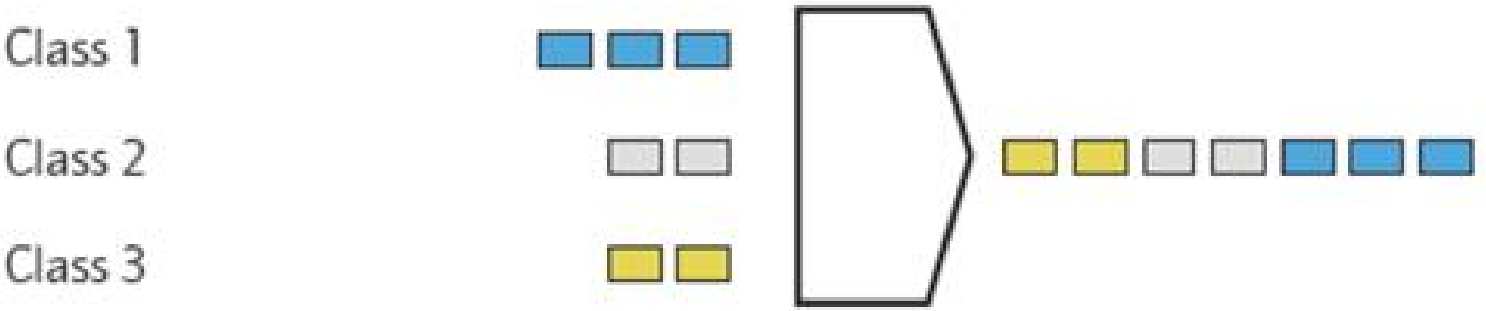
- Are named message destination that serve as an intermediate between sender and receiver
- Allow processes to execute and fail independently
- Can mask process failures and communication failures

- 作为发送和接收之间的命名消息目的地
- 允许进程独立执行和失败
- 可以掩盖进程失败和通信失败

消息优先级-最高优先级优先

Message Priorities – Highest Priority First

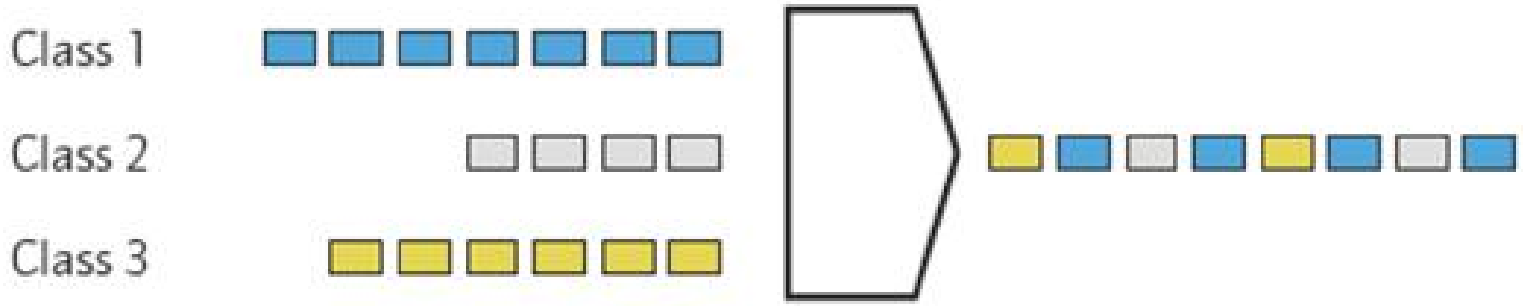
Example: Class 1 has a high, class 2 a medium, and class 3 a low priority



信息优先级-加权公平调度

Message Priorities – Weighted Fair Scheduling

Example: Class 1 has weight 2, the other have both weight 1

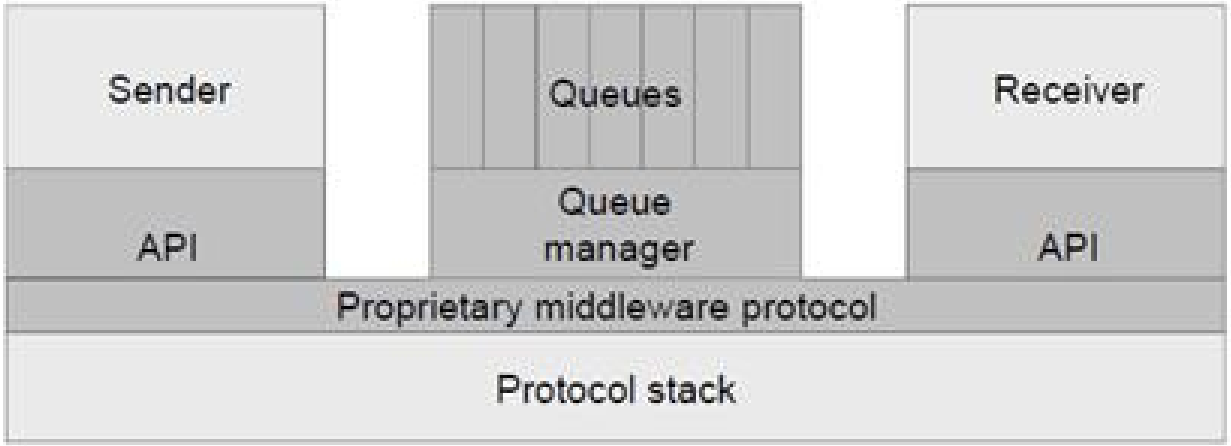


面向消息的中间件

Message Oriented Middleware

- Additional / Optionally
  - Support of multiple Messaging models
  - Queue management
  - Connection Management
  - Quality-of-Service (QoS)
  - Data transformation

- 支持多种消息传递模型
- 队列管理
- 连接管理
- 服务质量（QoS）
- 数据转换



队列管理器

Queue Manager

- Is a specialized kind of database
- Provides queuing functionality via an API to applications
- Provides means for administration
  - Creation/deletion of queues
  - Allows to start and to stop queues
  - Alter properties of existing queues
  - Allows monitoring of performance, failures, and recoveries

- 创建/删除队列
- 允许启动和停止队列
- 更改现有队列的属性
- 允许监视性能、故障和恢复

- Often queue managers can be configured to forward messages to other queue managers

通常，队列管理器可以配置为将消息转发给其他队列管理器

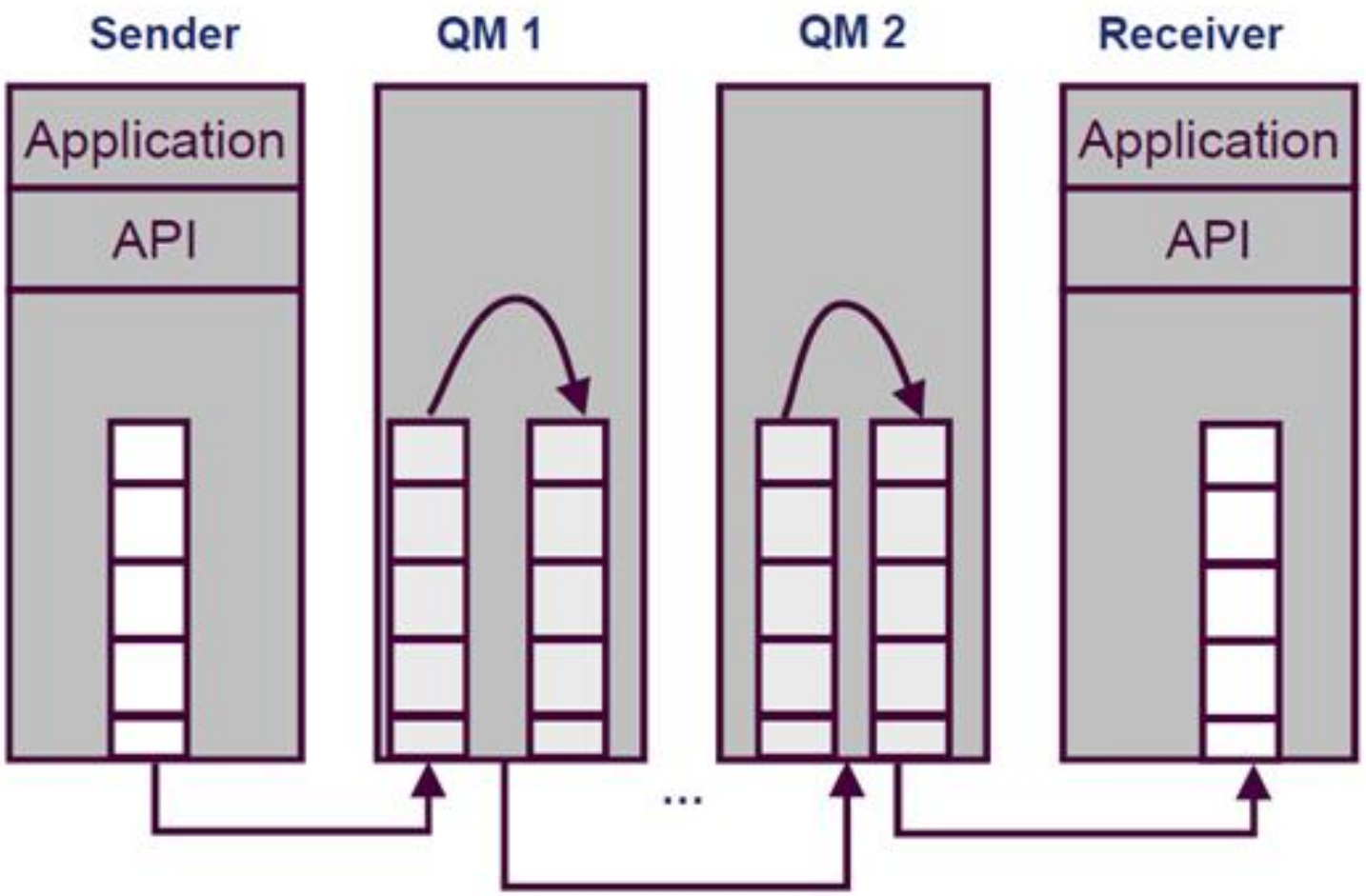
队列管理器

Additional / Optional Queue Manager Functionality

- Often called **Message Broker** • 消息代理
- Message Transformation
  - Primary function = Reformatting data/information • 重新格式化数据
  - "Rosetta stone" of the system -> universal translation • 全局翻译
  - Understands the structure/formats of sources and targets • 了解源代码和目标的结构/格式
- Intelligent Routing
  - Flow control
  - See 'Content based routing'
  - Message Dictionary
- Rules Engine
- Repository
- Filtering
- Access control

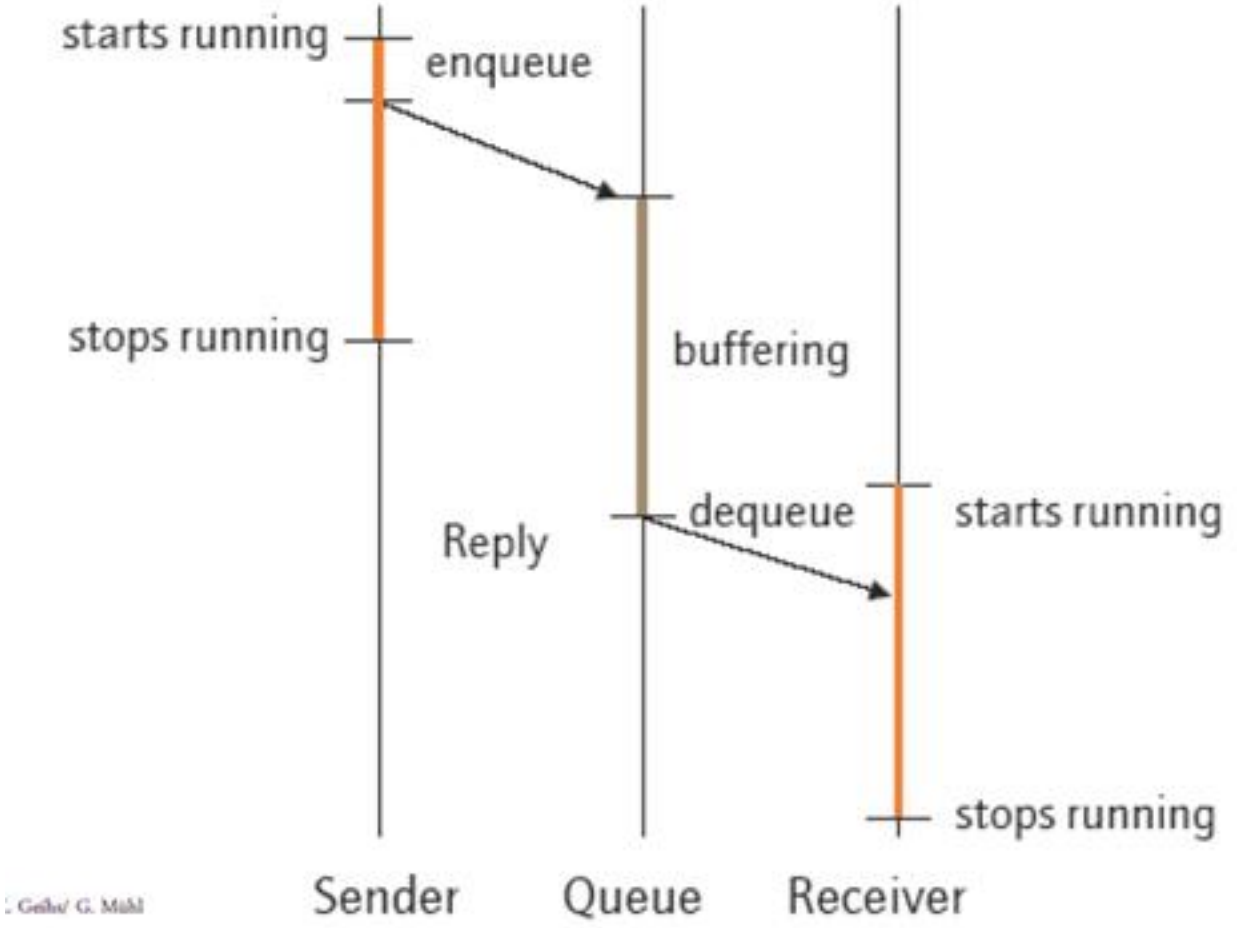
面向消息的中间件

Message Oriented Middleware



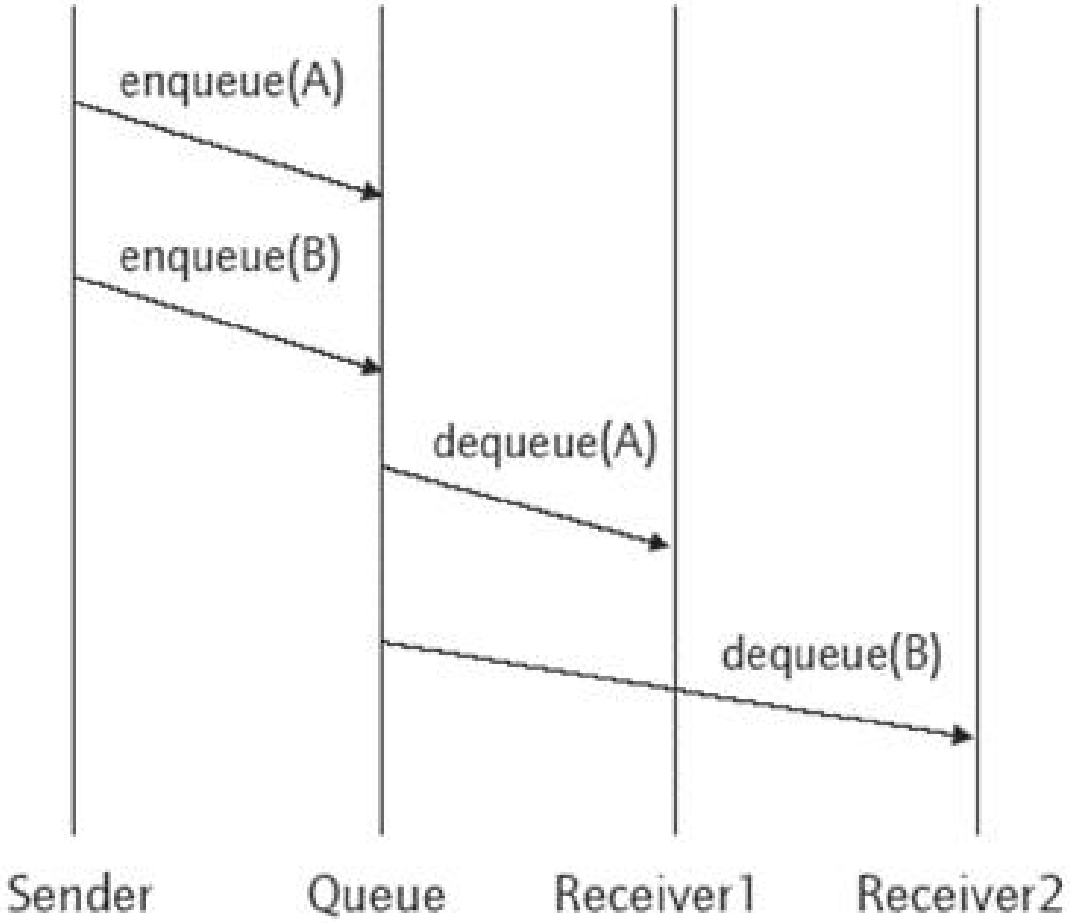
按队列进行时间解耦

Time Decoupling by Queues



按队列划分的位置解耦

Location Decoupling by Queues



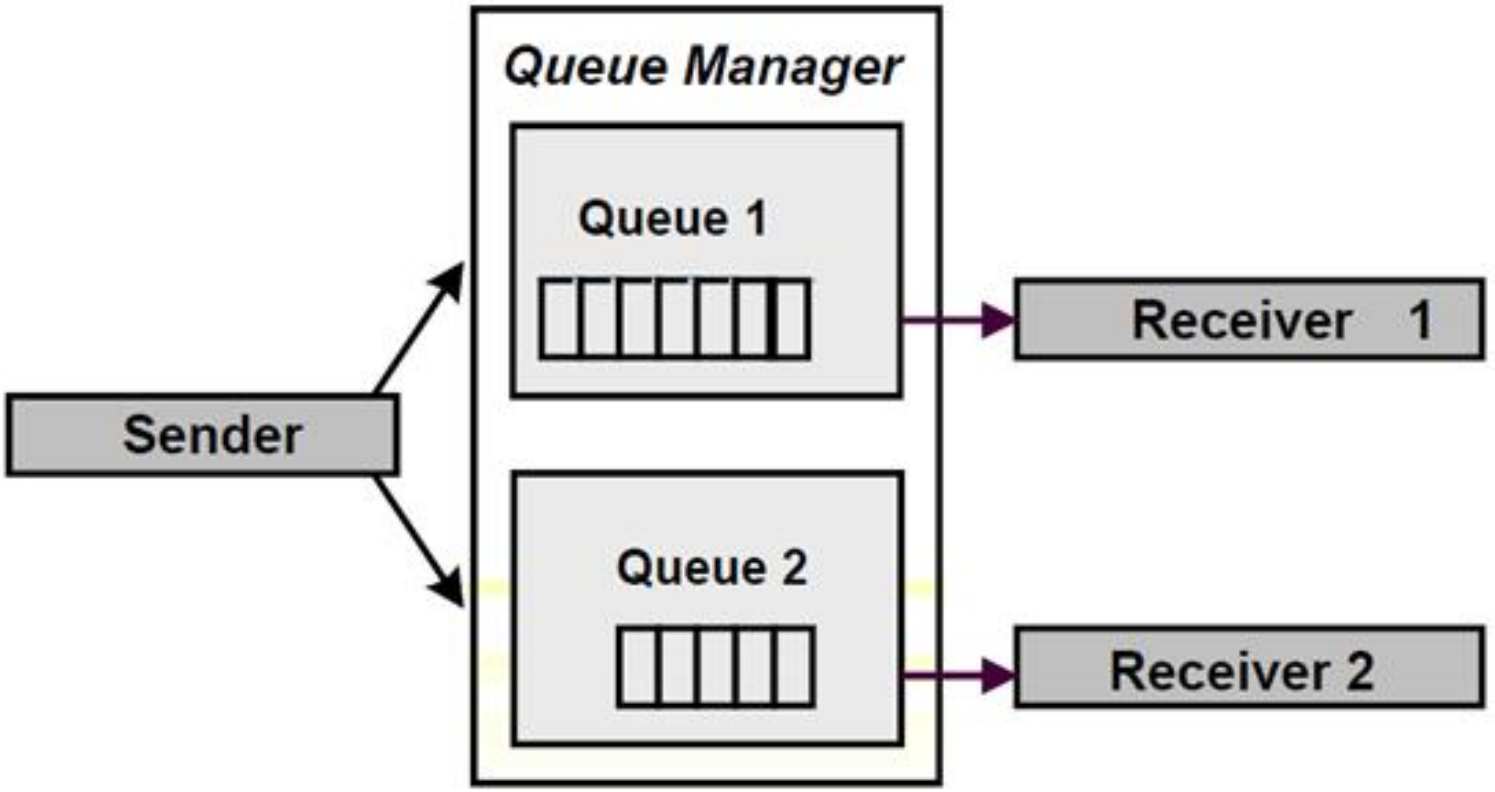
队列消息传递模式

Queuing Messaging Pattern

- **One-to-One**
  - One sender, one receiver
  - Point-to-Point
  - Message Passing (MP)
- **One-to-Many**
  - One sender, many receivers
- **Many-to-One**
  - Many senders, one receiver
- **Many-to-Many**
  - Many senders, many receivers
  - -> Publish and Subscribe

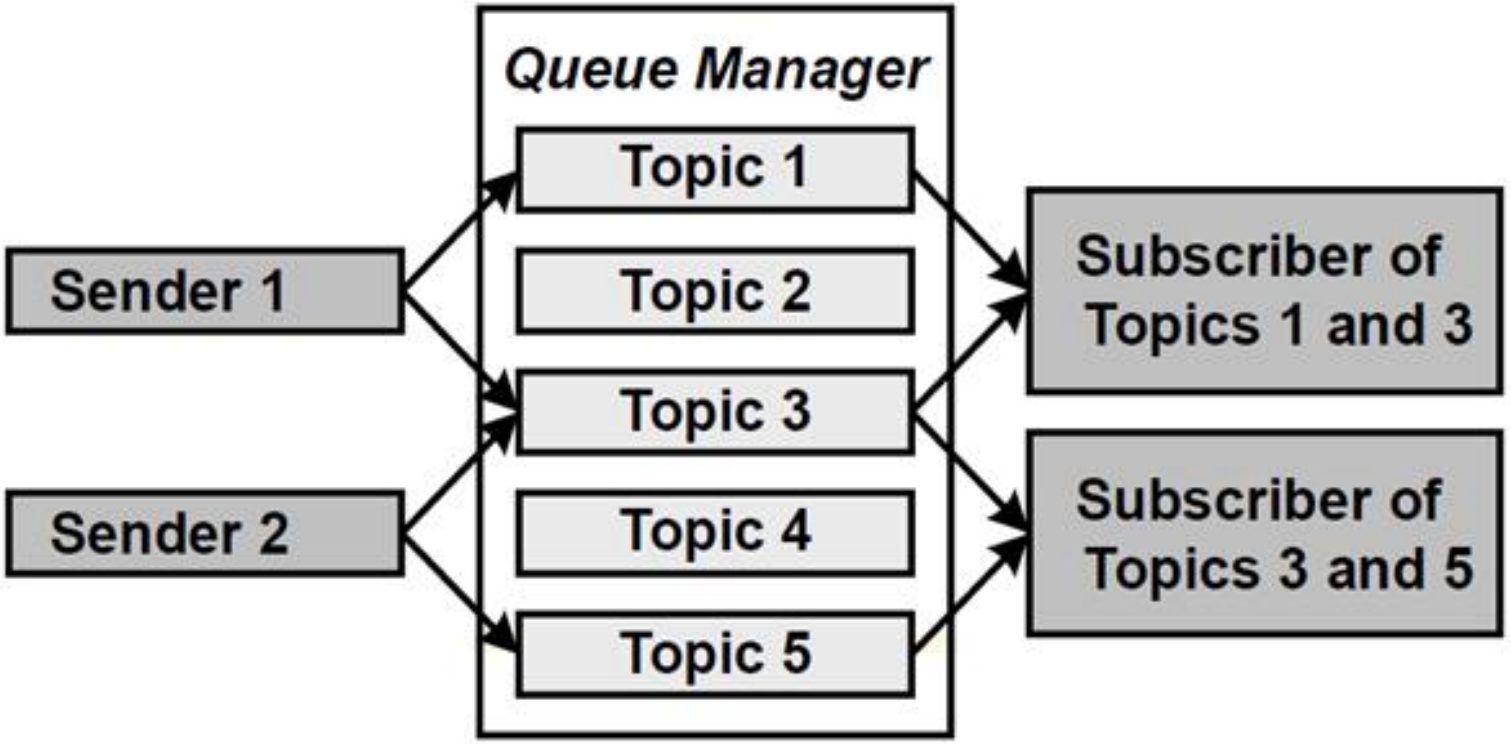
消息传递/点对点

Message Passing / Point-to-Point



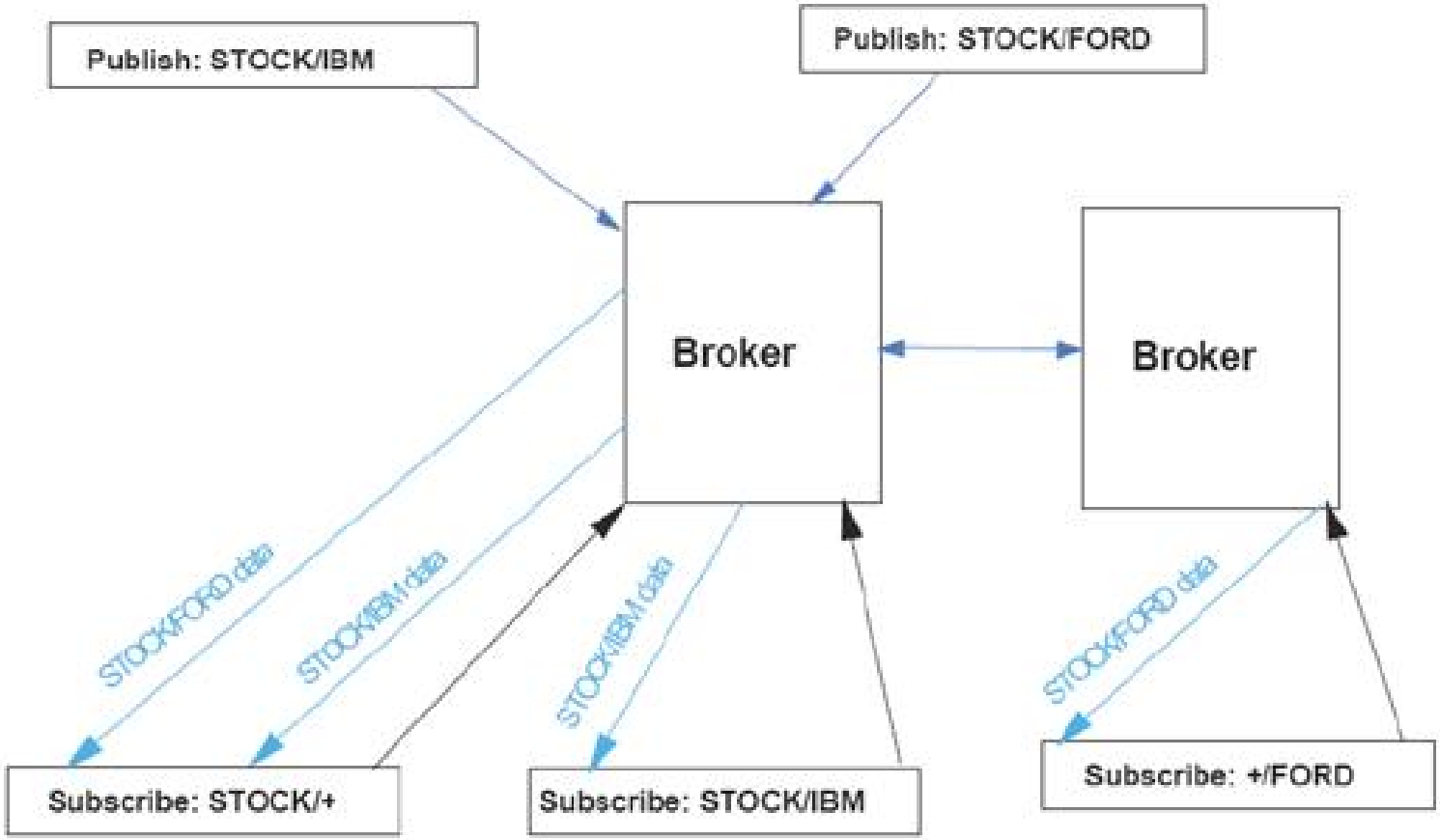
发布和订阅

Publish and Subscribe



发布和订阅

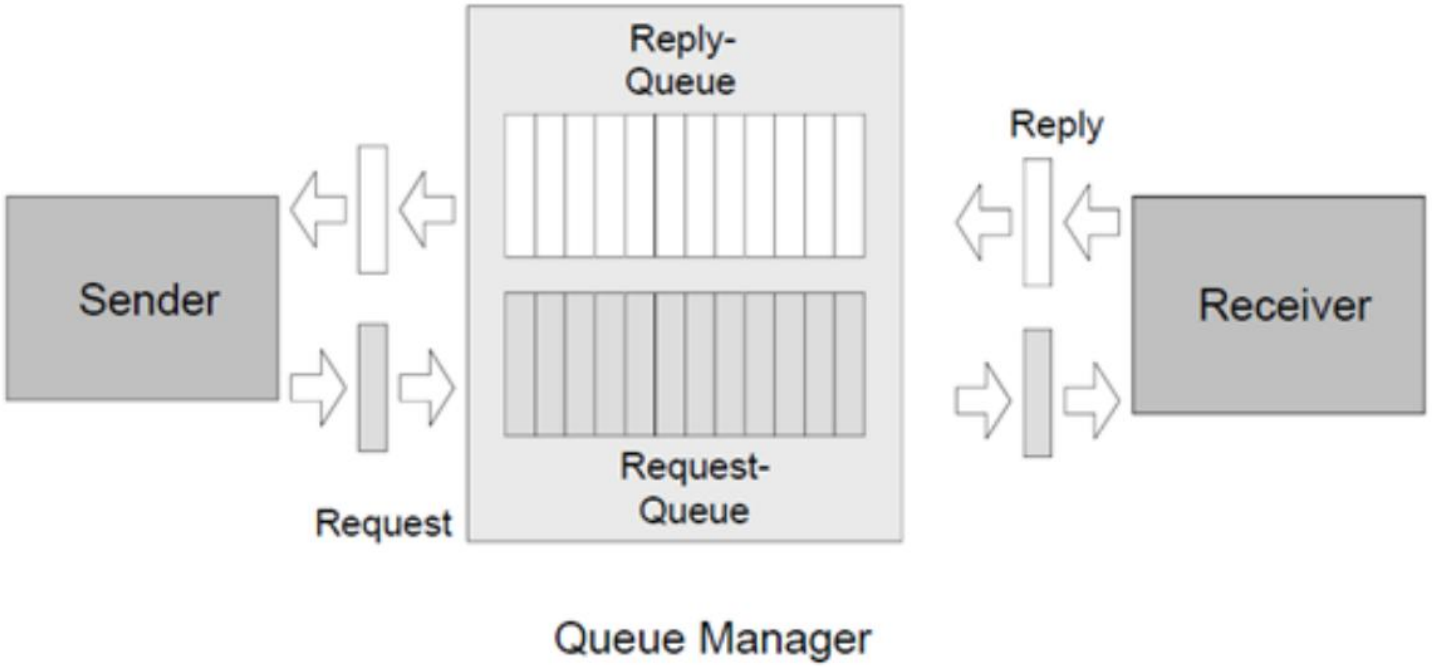
Publish and Subscribe



请求/回复队列

Request/Reply with Queues

- Correlation ID used to match request with *corresponding* reply at the client



基于消息的系统/ MOM产品

Message Based Systems / MOM Products

- Canonical example: IBM MQSeries
- (Sockets) – Berkeley Sockets
- OpenJMS
- Apache ActiveMQ (maintained by Fusesource)
- IBM WebSphere MQ (IBMs MQSeries)
- Amazon Simple Queue Service (SQS)
- TIBCO TIBCO Rendezvous
- Oracle Advanced Queuing
- Microsoft Microsoft Message Queuing (MSMQ)
- SUN Sun ONE Message Queue (JMS)
- RabbitMQ
- JBoss HornetQ

第3章

计算层的软件架构技术

Thanks for listening

涂志莹

哈尔滨工业大学计算机学院

企业与服务计算研究中心