

# Communication Paradigms

**INF 5040 autumn 2015**

**lecturer: Roman Vitenberg**

INF5040, Roman Vitenberg

1

## What is a communication paradigm?

- It is a set of communication primitives
  - Network adapters allow us to send data using MAC-layer primitives
  - Transport layer endows us with sockets
  - How could we raise the abstraction further?
- Study of a communication paradigm:
  - Properties
  - Target applications
  - Underlying implementation concepts

INF5040, Roman Vitenberg

2

## Communication properties

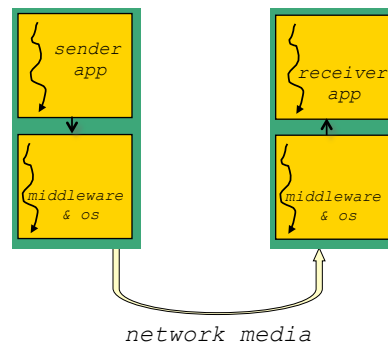
- Addressing scheme and space decoupling
  - Underlying protocol addresses (IP) – no decoupling
  - Logical aliases – partial decoupling
    - DNS and NAT translation, service names, email aliases
  - Content-based addressing – full decoupling
    - Interactions are declarative
- Persistence level
  - Fully persistent
  - Fully transient
  - Intermediate

INF5040, Roman Vitenberg

3

## Communication properties

- Synchrony
  - Fully synchronous
  - Fully asynchronous
  - Intermediate
    - middleware-level sync
    - man-in-the-middle
    - others
- Time decoupling



INF5040, Roman Vitenberg

4

## (some) **Communication paradigms**

- Remote procedure call
  - Object-based (CORBA, Java RMI, DCOM)
  - Earlier data-based (DCE, Sun RPC)
- Message-oriented communication
- Stream-oriented communication
- Software-based distributed shared memory (DSM)

INF5040, Roman Vitenberg

5

## (some) **Message-oriented communication paradigms**

- Raw socket programming
- Message-passing interface (MPI)
- Message-oriented middleware (MOM)
- Publish-subscribe communication

INF5040, Roman Vitenberg

6

## Raw socket programming

- Addressing scheme: IP addresses
- No time decoupling
- Transient
- Mainly used for building higher-level abstractions

INF5040, Roman Vitenberg

7

## Message-programming interface (MPI)

- Addressing scheme
  - A group of nodes assigned logical addresses
- Failures are considered fatal
- Transient without time decoupling
- Data-oriented (advanced data manipulation)
  - Basic API: `MPI_send`, `MPI_recv`
  - Data-oriented API: `MPI_scatter`, `MPI_gather`
- Use: parallel computation in fast networks

INF5040, Roman Vitenberg

8

## Message-oriented middleware (MOM)

- Addressing scheme: logical queue name
- Persistent
- Full time decoupling

`put(msg, dest queue name)`    `get(local queue name)`

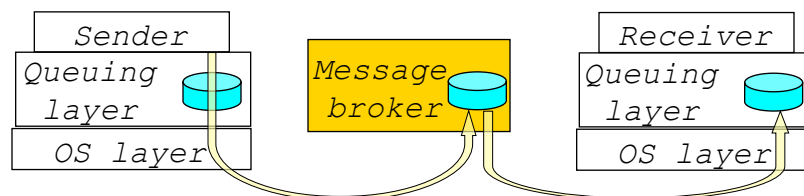


INF5040, Roman Vitenberg

9

## Routing in MOM

- Handles queue name to address translation
  - Hierarchical names: {queue manager, internal id}
- Message brokers perform inter-domain routing with format conversion

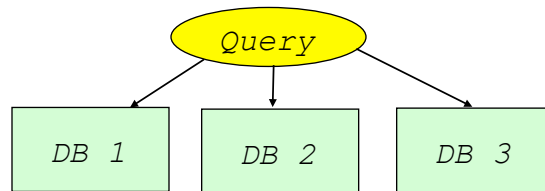


INF5040, Roman Vitenberg

10

## MOM applications & implementations

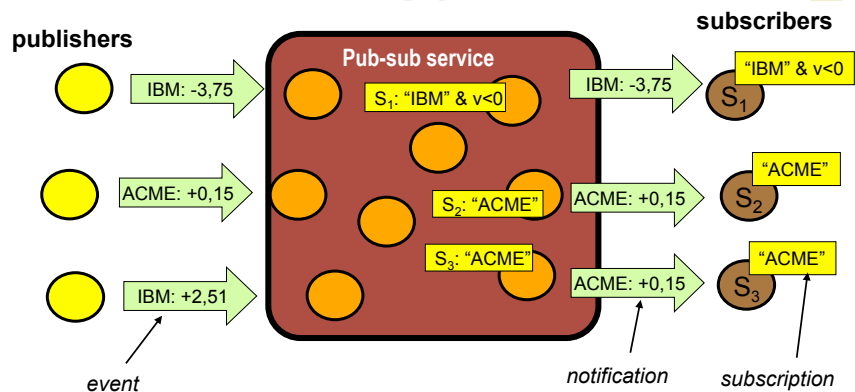
- Implementations: IBM MQ, Oracle AQ
- The E-mail application
- Workflow and other collaborative apps
- Federated information systems



INF5040, Roman Vitenberg

11

## Publish-subscribe communication



- Publishers: objects of interest or observers

INF5040, Roman Vitenberg

12

## Pub-sub properties

- Addressing scheme: through contents
- Full time decoupling
- May be persistent or transient
- Architectural trend through the past decade
  - Centralized (one server or a cluster of replicated servers)  
↓
  - Statically configured infrastructure of message brokers  
↓
  - Autonomous overlay of subscribers

INF5040, Roman Vitenberg

13

## Pub-sub applications

- Event-based business processes
- News distribution
  - The research-originated Gryphon system was part of the Web infrastructure serving the Olympic games in 2000
  - More recently: RSS and RSS aggregators
- Delivery of financial data
  - Many stock exchanges around the world
- Intrusion detection and other applications of distributed data mining
- Online games
- Many others ...

INF5040, Roman Vitenberg

14

## Subscription semantics

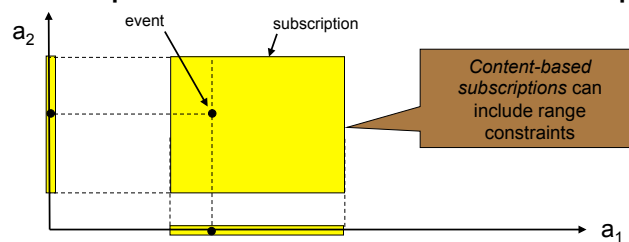
- Topic-based pub-sub:
  - *publish(topic t), subscribe(topic t)*
  - The topic namespace may be hierarchical
  - Wildcards: *subscribe("nasdaq.stockvalue.a\*")*
- Type-based pub-sub
  - Generalization of topic hierarchy
  - Uses the fact that events of the same type have the same structure (fields)

INF5040, Roman Vitenberg

15

## Subscription semantics

- Content-based pub-sub
  - Universally known list of event attributes
  - Event represented as a set of attribute values
    - A point in the multi-dimensional event space
  - Subscription is a cuboid in the event space

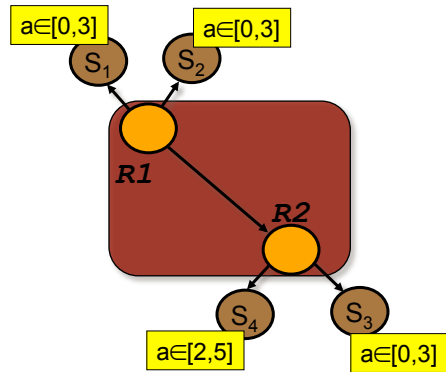


INF5040, Roman Vitenberg

16



## Content-based routing



The routing table of R1

<b>Interface</b>	<b>Filter</b>
To node 1	$a \in [0, 3]$
To node 2	$a \in [0, 3]$
Toward R2	$a \in [0, 5]$

## Communication paradigms (summary)

<i>Abstraction</i>	<i>Space decoupling</i>	<i>Time decoupling</i>	<i>Persistence</i>
<i>Raw sockets</i>	no	no	no
<i>RPC</i>	no	no	no
<i>MOM</i>	partial	yes	yes
<i>Pub-sub</i>	full	yes	possible

## Multicast and its effect on communication abstraction

- Can appear as an element in many paradigms or be considered as a paradigm by itself
- Makes complicated communication abstractions even more complicated
  - Addressing scheme becomes even more important
    - Stronger case for space decoupling
  - Reliability issues become more involved
  - Message orderings
  - Atomicity

*Not transparent for apps,  
Affects the paradigm*

INF5040, Roman Vitenberg

20

## The challenges of supporting multicast communication

- No standardized transport protocols to rely upon
  - What about IP-multicast?
    - Not always available
    - Historical trend: shift of the solutions from the network to application level
- Different approaches
  - Emulate multicast by unicast
  - Overlay-based multicast
  - Epidemic or gossip-based dissemination
  - Result in different paradigms

INF5040, Roman Vitenberg

21

## Overlay-based multicast

- Organize the destination nodes in a logical application-level network graph (*overlay*)
- Disseminate messages using overlay links
- Monitor links and nodes: failures, link quality, communication load
- Incrementally reconstruct upon joins, leaves, overload, link and node failures

INF5040, Roman Vitenberg

23

## Overlay-based multicast (the underlying principles)

- It is possible to achieve both good scalability and low latency at the same time
  - Logarithmic or better fan-out for scalability
  - Short routing paths (logarithmic # of hops)
- The **small-world phenomenon**
  - Overlay topology induced by the physical one
    - (e.g., a rectangular grid of sensors)
  - Adding a single link from each node to a random destination node is enough to create short routing paths

INF5040, Roman Vitenberg

24

## Multicast overlay types

- Multicast tree
  - The most efficient dissemination
  - Simple routing scheme (flooding)
  - The load is distributed non-evenly
  - Highly vulnerable to failures
- Other overlays (regular hypercube, regular random graph, rectangular grid)
  - Better load distribution & resilience to failures
  - More complicated routing scheme

INF5040, Roman Vitenberg

26

## Epidemic dissemination

- Observe how fast epidemics propagate in the absence of treatment
- Use the same principles for the positive purpose of message dissemination
- **Infected**, **susceptible**, and **removed** nodes
- Based on membership: every node maintains a (possibly partial) membership of other nodes it can communicate with

INF5040, Roman Vitenberg

27

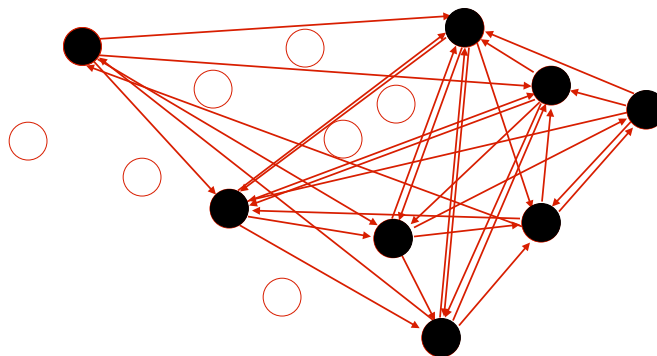
## Epidemic Dissemination (Push)

- The protocol is parameterized by *infection period*  $t$  and *fan-out*  $f$ :
  - When a node becomes infected, it executes  $t$  rounds and then becomes removed
  - At each round, it sends the message to  $f$  random nodes from its membership list
- Global round  $k$ : every node has executed at least  $k$  rounds and at least one node has executed exactly  $k$  rounds

INF5040, Roman Vitenberg

28

## Push Epidemic Dissemination Example ( $t=2, f=2$ )



INF5040, Roman Vitenberg

29

## Epidemic Dissemination (Pull)

- Each susceptible node executes an unlimited number of rounds until it becomes infected
- At each round, it contacts  $f$  random nodes from its membership list, checks if one of them is infected, and pulls the message
- Can be combined with push dissemination to form a push-pull approach

INF5040, Roman Vitenberg

30

## Epidemic dissemination (properties)

- Fault-tolerance: no need to detect message losses due to link and node failures, no message retransmissions
- Probabilistic atomicity (bimodal behavior): depending on  $t$  and  $f$ , the message is likely to be delivered
  - either to almost all nodes
  - or to a negligible portion of nodes
- The propagation is fast: if it reaches almost all nodes, it does so in  $O(\log N)$  global rounds

INF5040, Roman Vitenberg

31

## Push vs pull gossiping

### ➤ Push approach:

- Fast & efficient when few nodes are infected
- When just a few nodes are susceptible
  - Takes a lot of time to reach susceptible nodes
  - A lot of unnecessary messages are sent

### ➤ Pull approach:

- Fast & efficient when most nodes are infected
- Wasteful and slow if few nodes are infected

INF5040, Roman Vitenberg

32

## Push vs Pull gossiping

### ➤ Push-pull approach:

- Fast propagation to all nodes
- Wasteful whatever portion of nodes is infected

### ➤ Rumor spreading:

- Push-based
- Non-constant # of rounds: whenever a node pushes to an already infected node, it becomes removed with probability  $p$
- Communication-efficient but slower dissemination

INF5040, Roman Vitenberg

33

## Membership properties

- Membership list of size  $L$ 
  - Infeasibility of full membership in large-scale systems
  - Fundamental tradeoff: smaller membership list scales better but may limit dissemination
    - Risk of partitioning the set of nodes
- Uniformity: partial lists are uniform samples
- Adaptivity: ideally,  $L$  should be adapted to  $N$ 
  - Nodes may have difficulty of estimating  $N$
- Bootstrapping: membership initialization

INF5040, Roman Vitenberg

34

## Applications of gossiping

- Failure detection
- Data aggregation
- Resource discovery and monitoring
  - Access to replicated web pages
- Update propagation for data caching
- Experimental: content search, file sharing

INF5040, Roman Vitenberg

35



## Comparison: overlay- vs gossip-based multicast

- Overlay-based multicast
  - Efficient propagation
  - 100% delivery guarantee in the absence of churn
  - Costly and complex reconfiguration upon churn
- Gossip-based multicast
  - Many unnecessary messages may be sent
  - May not reach 100% of nodes even in a completely stable environment
  - Very resilient to all kind of churn

INF5040, Roman Vitenberg

37

## Reading material

- TvS Sections 4.1.2, 4.3, 4.5, 13.4.1
- Coulouris et al. Sections 6.1, 6.3 and 6.4
- “The Many Faces of Publish/Subscribe” by Eugster, Felber, Guerraoui, Kermarrec
  - Can be found in the teaching plan on the web
- “Epidemic Information Dissemination in Distributed Systems” by Eugster, Guerraoui, Kermarrec, Massoulie

INF5040, Roman Vitenberg

38