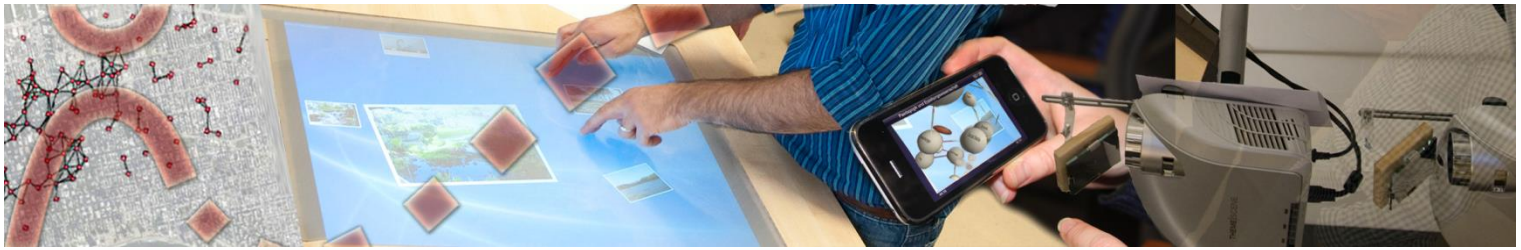
Telecooperation Lab
Prof. Dr. Max Mühlhäuser

# Telekooperation 1: Exercise WS15/16

Michael Stein, MSc.
michael.stein@tk.informatik.tu-darmstadt.de
Jens Heuschkel, MSc.
jens.heuschkel@tk.informatik.tu-darmstadt.de



# TK1 – EXERCISE

- Solution 5th Exercise
- 6th Theory Exercise

# Task 1: Routing with Subscriptions (4,5 P.)

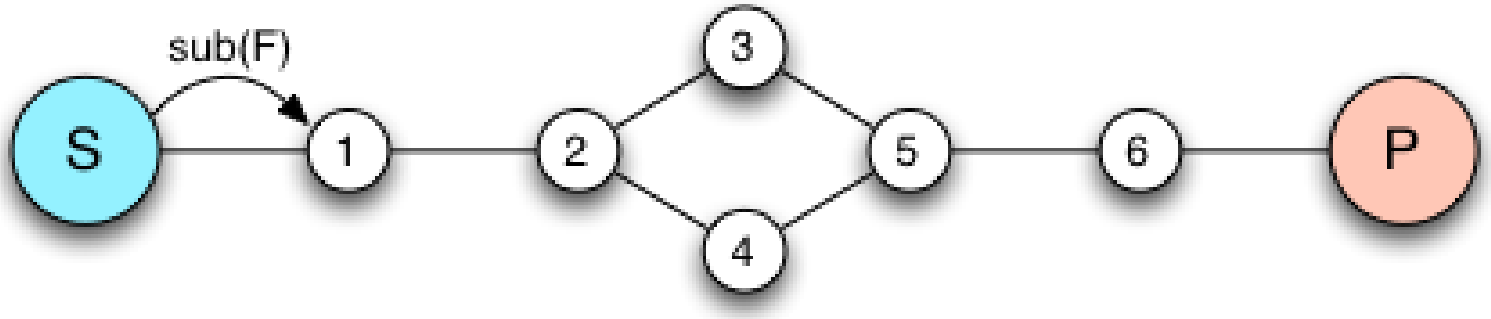
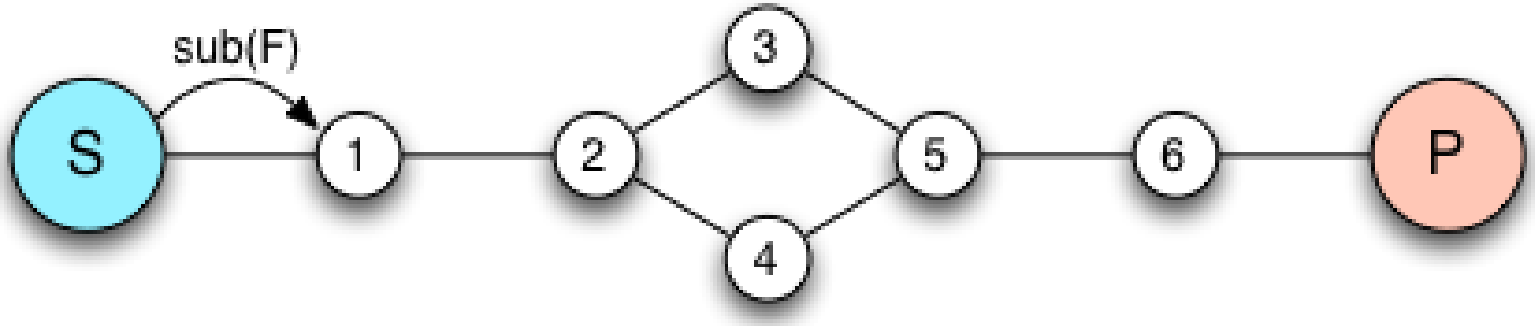



Illustration 1: Routing network

Consider the routing network in illustration 1. Apply the algorithm "Routing with Subscriptions".

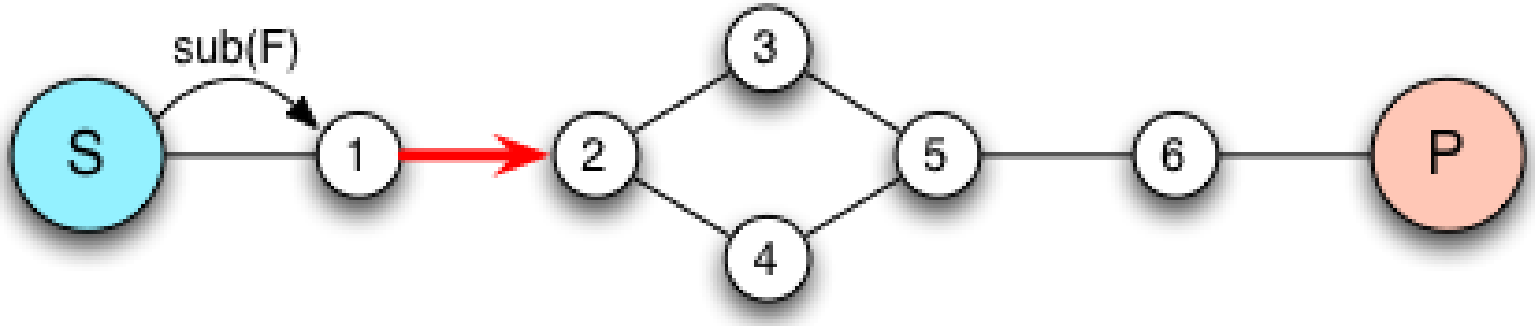
1) Subscriber S sends a subscription to router 1 with filter F. Describe the flow of the subscriptions between the routers, i.e. write down all routing tables in each step and paint the graph with the current flowing subscription also in each step.

# Task 1: Routing with Subscriptions (4,5 P.)



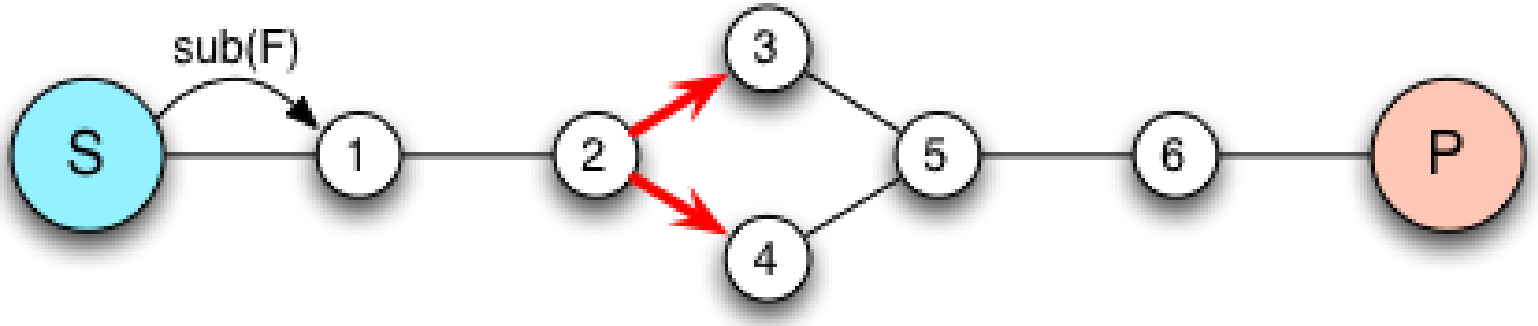

| Router 1 | | Router 2 | | Router 3 | | Router 4 | | Router 5 | | Router 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D | Filter | D | Filter | D | Filter | D | Filter | D | Filter | D | Filter |
| | | | | | | | | | | | |

# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | | Router 2 | | Router 3 | | Router 4 | | Router 5 | | Router 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D | Filter | D | Filter | D | Filter | D | Filter | D | Filter | D | Filter |
| | | 1 | F | | | | | | | | |

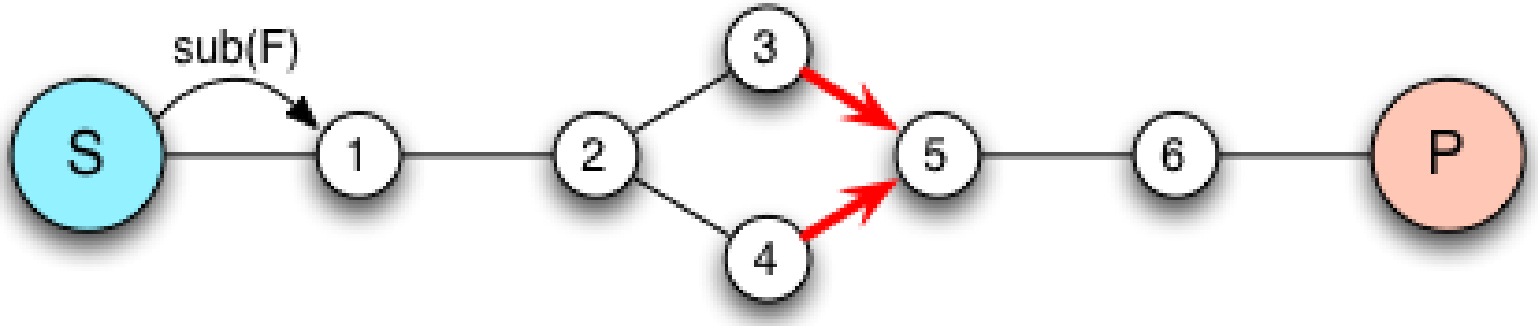
# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | | Router 2 | | Router 3 | | Router 4 | | Router 5 | | Router 6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D | Filter | D | Filter | D | Filter | D | Filter | D | Filter | D | Filter |
| | | 1 | F | 2 | F | 2 | F | | | | |

Forwarding of router 2 to router 3 and 4 were combined

# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | |
|---|---|
| D | Filter |
| | |

| Router 2 | |
|---|---|
| D | Filter |
| 1 | F |

| Router 3 | |
|---|---|
| D | Filter |
| 2 | F |

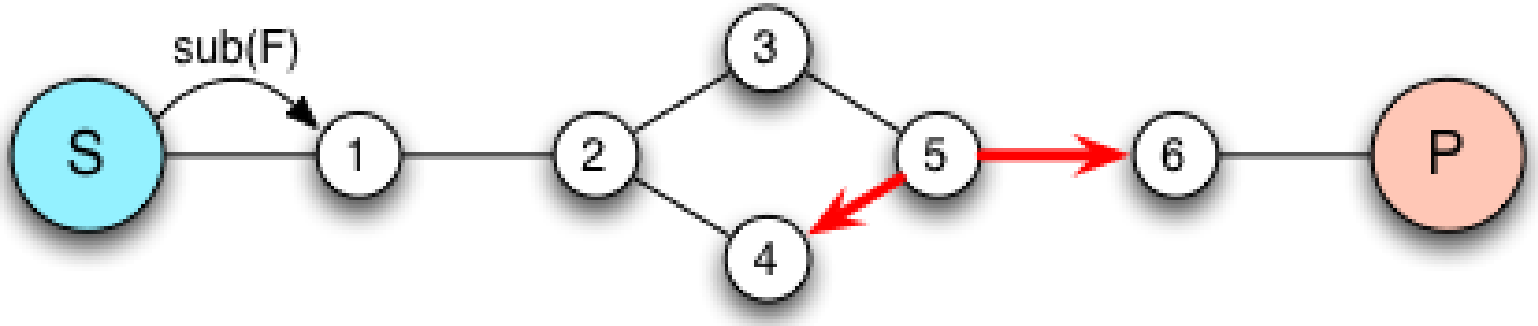
| Router 4 | |
|---|---|
| D | Filter |
| 2 | F |

| Router 5 | |
|---|---|
| D | Filter |
| 3 | F |
| 4 | F |

| Router 6 | |
|---|---|
| D | Filter |
| | |

Forwarding of router 3 and 4 were combined

# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | |
|---|---|
| D | Filter |
| | |

| Router 2 | |
|---|---|
| D | Filter |
| 1 | F |

| Router 3 | |
|---|---|
| D | Filter |
| 2 | F |

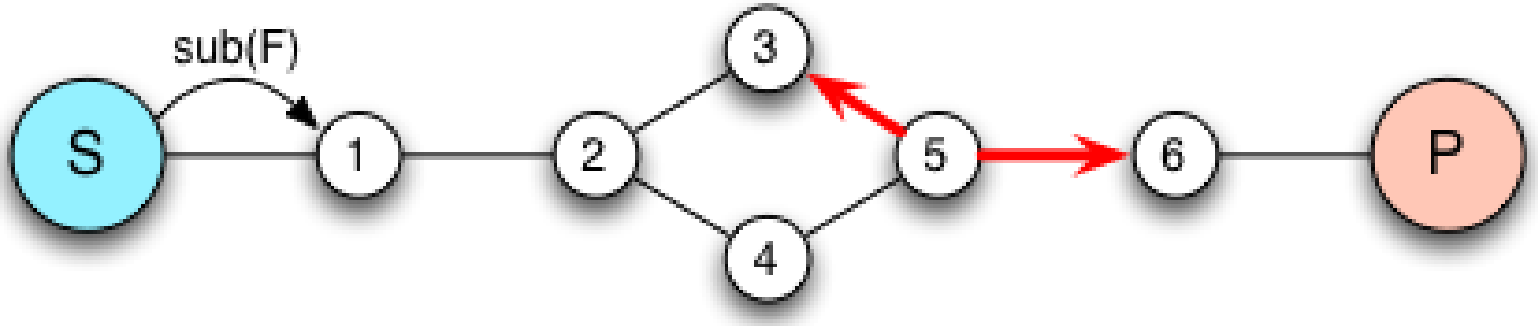
| Router 4 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

| Router 5 | |
|---|---|
| D | Filter |
| 3 | F |
| 4 | F |

| Router 6 | |
|---|---|
| D | Filter |
| 5 | F |

Router 5 processes sub(F) of router 3

# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | |
|---|---|
| D | Filter |
| | |

| Router 2 | |
|---|---|
| D | Filter |
| 1 | F |

| Router 3 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

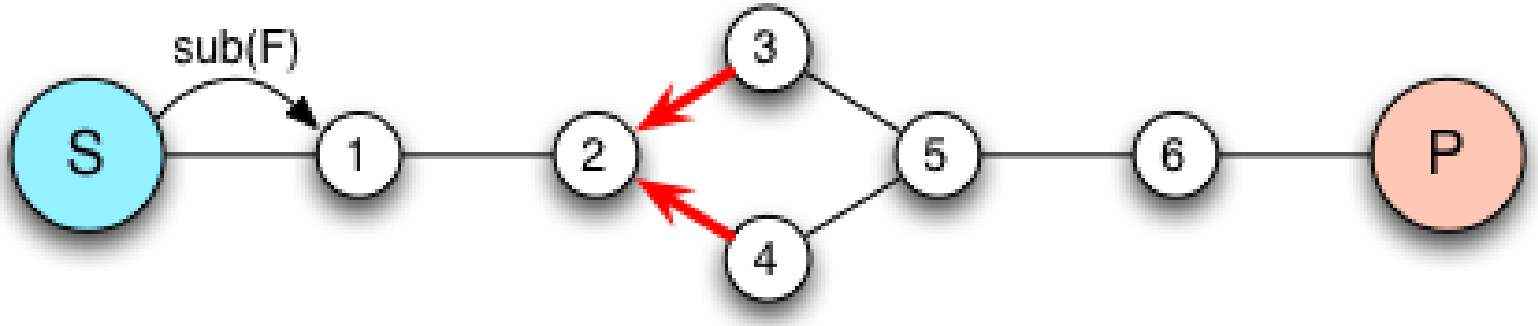
| Router 4 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

| Router 5 | |
|---|---|
| D | Filter |
| 3 | F |
| 4 | F |

| Router 6 | |
|---|---|
| D | Filter |
| 5 | F |

Router 5 processes sub(F) of router 4

# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | |
|---|---|
| D | Filter |
| | |

| Router 2 | |
|---|---|
| D | Filter |
| 1 | F |
| 3 | F |
| 4 | F |

| Router 3 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

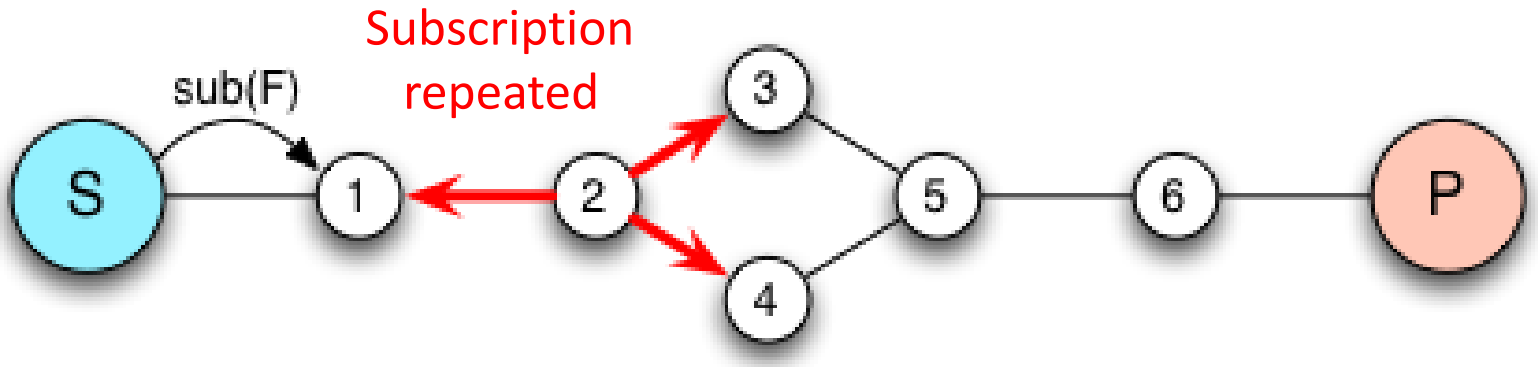
| Router 4 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

| Router 5 | |
|---|---|
| D | Filter |
| 3 | F |
| 4 | F |

| Router 6 | |
|---|---|
| D | Filter |
| 5 | F |

Forwarding of router 3 and 4 were combined

# Task 1: Routing with Subscriptions (4,5 P.)




| Router 1 | |
|---|---|
| D | Filter |
| 2 | F |

| Router 2 | |
|---|---|
| D | Filter |
| 1 | F |
| 3 | F |
| 4 | F |

| Router 3 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

| Router 4 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

| Router 5 | |
|---|---|
| D | Filter |
| 3 | F |
| 4 | F |

| Router 6 | |
|---|---|
| D | Filter |
| 5 | F |

# Task 1: Routing with Subscriptions (4,5 P.)

TECHNISCHE UNIVERSITÄT DARMSTADT

All routing tables.

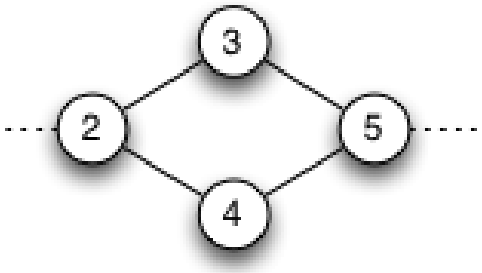
| Source | Dest | Request | Filter |
|---|---|---|---|
| 1 | 2 | sub | F |
| 2 | 3 | sub | F |
| 2 | 4 | sub | F |
| 3 | 5 | sub | F |
| 4 | 5 | sub | F |
| 5 | 6 | sub | F |
| 5 | 4 | sub | F |
| 5 | 3 | sub | F |
| 3 | 2 | sub | F |
| 4 | 2 | sub | F |
| 2 | 1 | sub | F |

and so on …

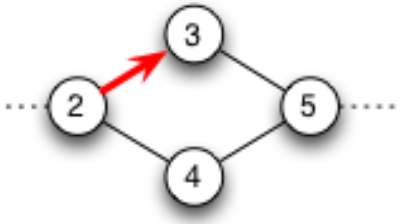
# Task 1: Routing with Subscriptions (4,5 P.)

TECHNISCHE UNIVERSITÄT DARMSTADT

2) Which problem arises and why? Give a solution to avoid this problem




- The subscriptions start to circulate
- Because the graph is not acyclic
- Solution
  - Subscriptions are only forwarded if they change the current routing table




| Router 2 | |
|---|---|
| D | Filter |
| 1 | F |
| 3 | F |
| 4 | F |

| Router 3 | |
|---|---|
| D | Filter |
| 2 | F |
| 5 | F |

Entry already exists

⇒ no updating, no forwarding

⇒ Algorithm terminates

# Task 1: Routing with Subscriptions (4,5 P.)

TECHNISCHE UNIVERSITÄT DARMSTADT

3) Arises an equivalent problem if P sends a notification? Give also a solution if this is the case.

- Yes, also the notification starts to circulate
- Solution
  - Time-To-Live-Counter in notification
  - Every router remembers which notifications he has seen

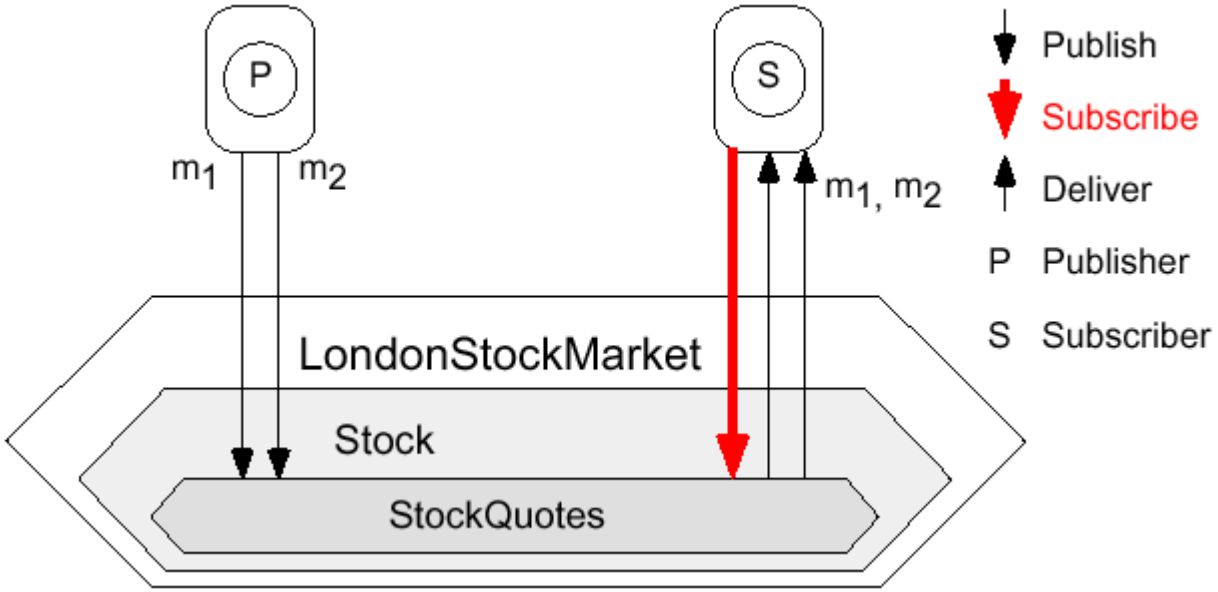
# Task 2: Addressing (4 P.)

Consider the four different types of Addressing in Publish/Subscribe Systems without Concept-Based Addressing.

1) How can a Subscriber subscribe in each type of Addressing?
2) Assume a Subscriber subscribes on a Publish/Subscribe System. How is the filtering realized in each type of Addressing?

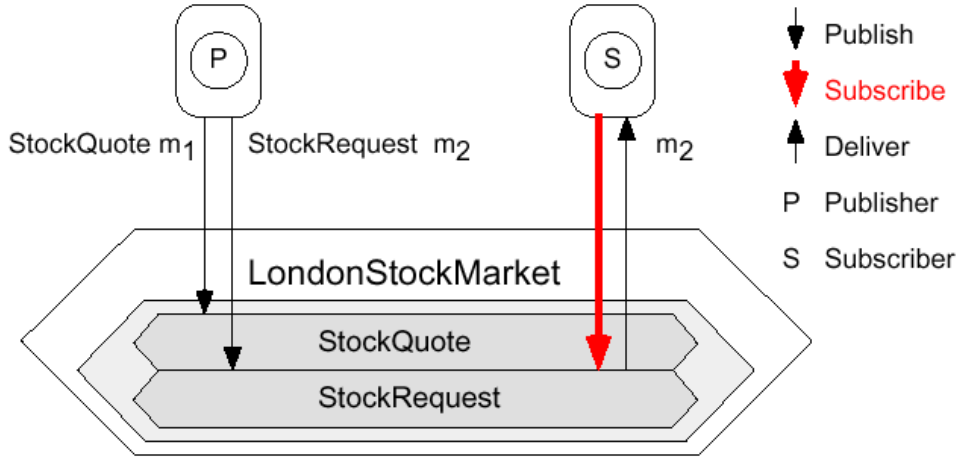
# Addressing

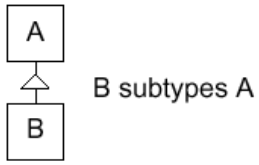
- Channel-based Addressing (=Topic-based)
  - **Interested parties can subscribe to a channel**
  - Application posts messages explicitly to a specific channel
  - **Channel Identifier is only part of message visible to event service**
  - There is no interplay between two different channels

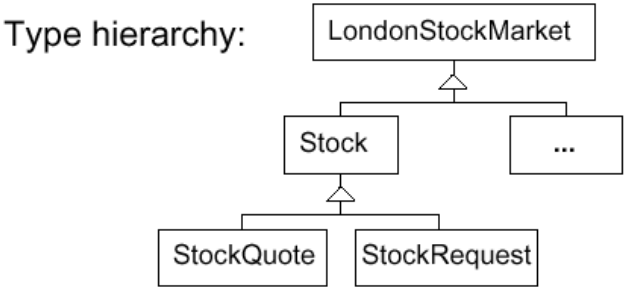


# Addressing

- Type-based Addressing
  - **Similar to channel-based Pub/Sub with hierarchies**
  - **Supports subtype tests (instanceof)**
  - Good integration of middleware & language, type safety



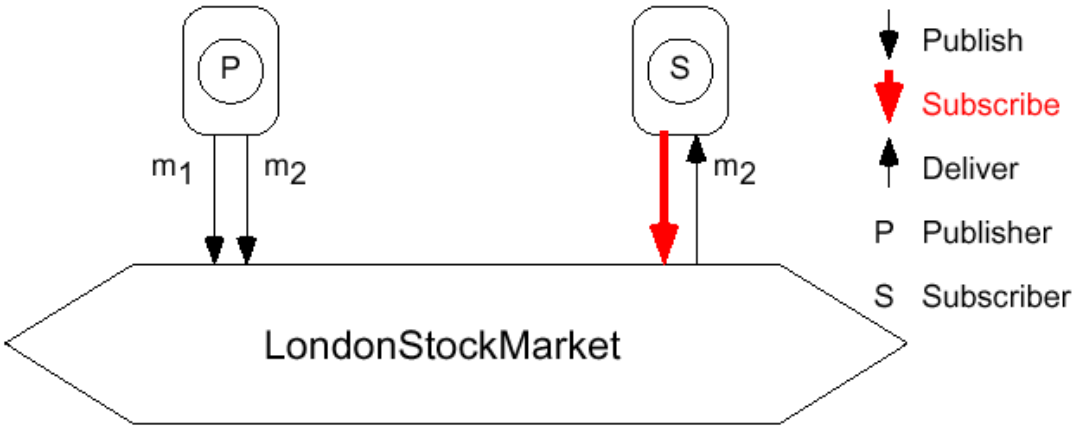
# Addressing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Subject-based Addressing
  - Notifications contain a well-known attribute – the subject – that determines their address
  - **Subscriptions express interest in subjects by some form of expressions to be evaluated against the subject**
  - Subject is
    - List of strings (TIB/Rendezvous, JEDI)
    - Properties: Typed Key/Value-Pairs (JMS)
  - **Subject (= header of notification) is visible to event service, remaining information is opaque**
  - Subscription is
    - (Limited form of) Regular Expressions over Strings (TIB, JEDI)
    - (Limited form of) SQL92 Queries (JMS)
  - Filtering is done in the Router Network!
  - Limited form of Content-based Subscription

# Addressing

- Content-based Subscription
  - **Domain of filters extended to the whole content of notification**
  - More freedom in encoding data upon which filters can be applied
  - More information for event service to set up routing information




$m_1$: { ..., company: "Telco", price: 120, ..., ... }

$m_2$: { ..., company: "Telco", price: 90 , ..., ... }

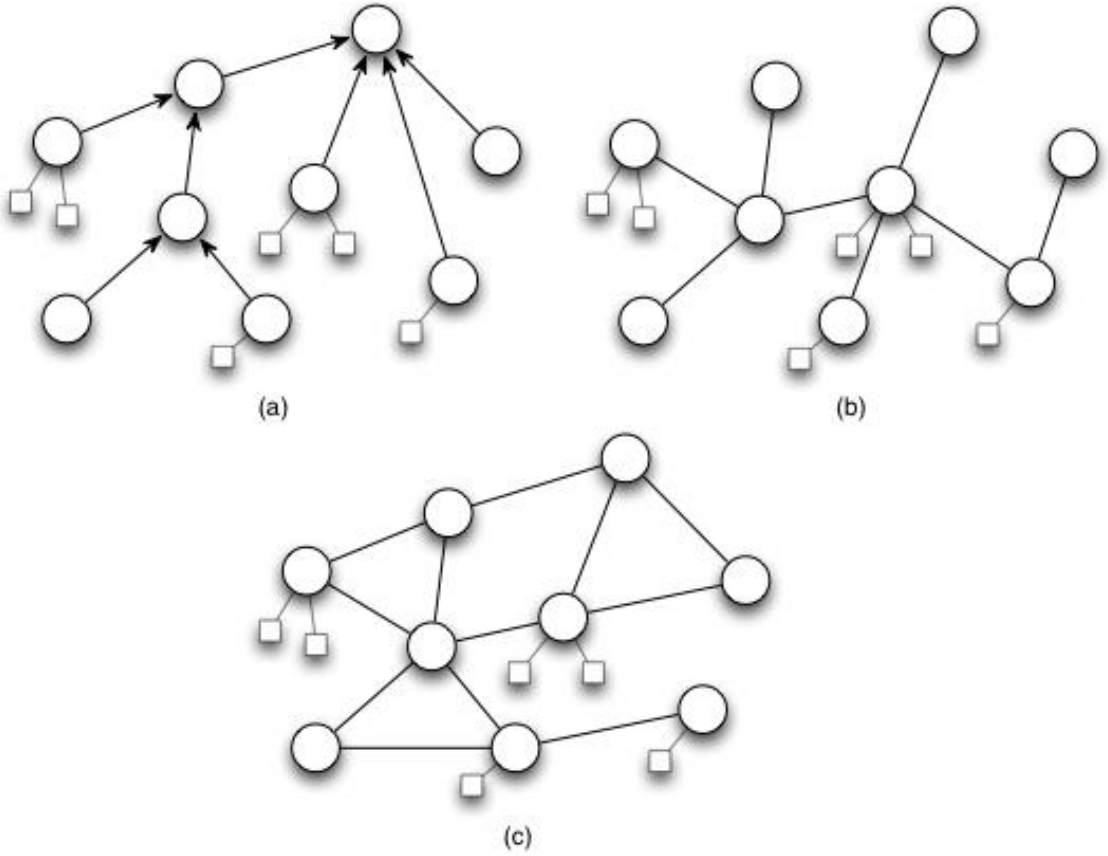
# Task 2: Addressing (4 P.)

TECHNISCHE UNIVERSITÄT DARMSTADT

| Type of Addressing | How subscribe? | How filtering? |
|---|---|---|
| Channel-based | By providing the channel name | By string comparison |
| Subject-based | By providing an expression on the entries of the message header (subject), e.g., regular expressions or SQL92 Queries | By evaluating the expression with the message header |
| Content-based | By providing an expression over the data object | By evaluating the expression with the data object |
| Type-based | By providing the expected object type | By type testing (instanceof) |

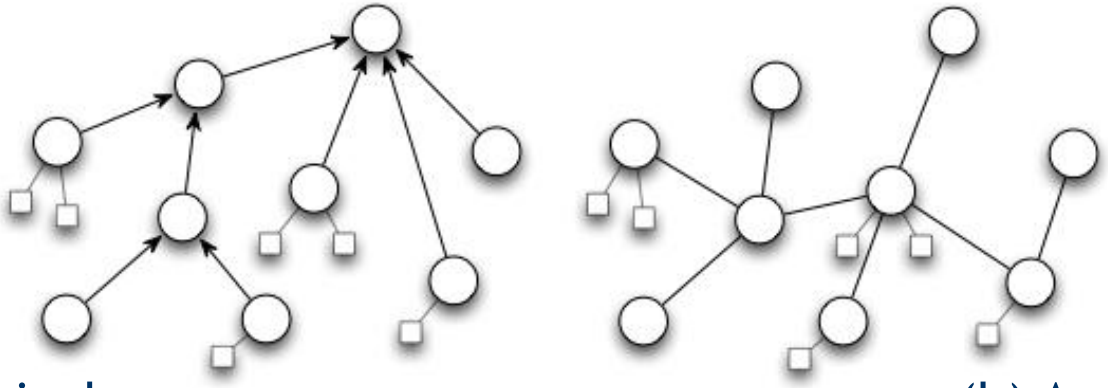
# Task 3: Routing Topologies (1,5 P.)

Consider and classify the different topologies in illustration 2. Give a reason for your decision.



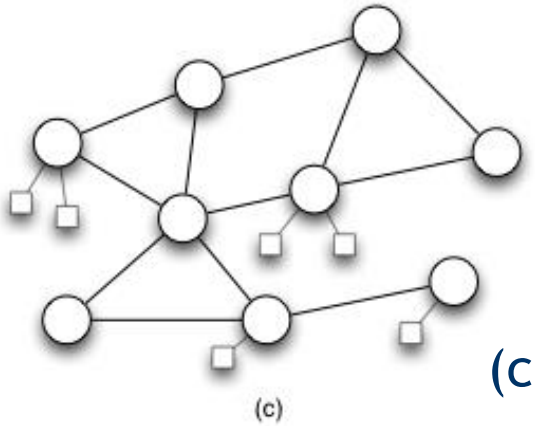
# Task 3: Routing Topologies (1,5 P.)

Consider and classify the different topologies in illustration 2. Give a reason for your decision.




(a) Hierarchical

(b) Acyclic Peer-to-Peer




(c) Generic Peer-to-Peer