Theory Exercise 6

Tarek Auel, Tiange Hu, Benedikt Christoph Naumann, Markus Schanz

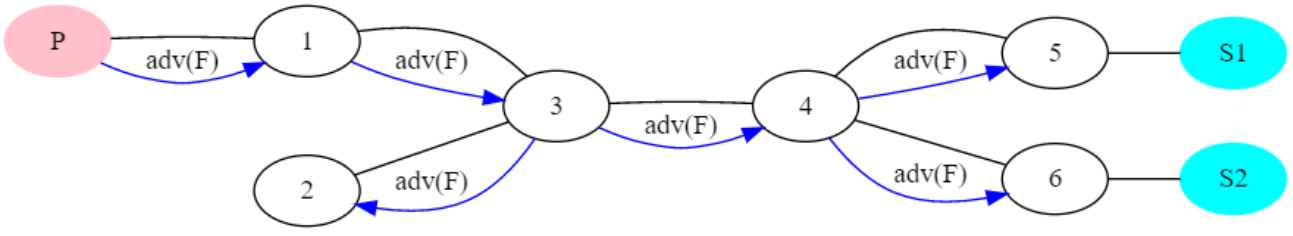
Task 1: Routing with Advertisements

Task 1.1: Publisher P sends an advertisement to router 1

- Step 1
  - (Advertisment of P is stored internally in router 1)
  - Router 1 forwards the advertisement to all neighbors
    - Forward advertisement to Router 3
- Step 2
  - Router 3 adds the advertisement to its T_A table.
  - Router 3 forwards the advertisement to all neighbors,except the origin
    - Forward advertisement to Router 2
    - Forward advertisement to Router 4
- Step 3
  - Router 2 adds the advertisement to its T_A table.
  - Router 2 forwards the advertisement to all neighbors, except the origin
    - No forwarding
- Step 4
  - Router 4 adds the advertisement to its T_A table.
  - Router 4 forwards the advertisement to all neighbors, except the origin
    - Forward advertisement to Router 5
    - Forward advertisement to Router 6
- Step 5
  - Router 5 adds the advertisement to its T_A table.
  - Router 5 forwards the advertisement to all neighbors, except the origin
    - No forwarding
- Step 6
  - Router 6 adds the advertisement to its T_A table.
  - Router 6 forwards the advertisement to all neighbors, except the origin
    - No forwarding

Steps **3 and 4**, as well as steps **5 and 6** may occur in arbitrary order.

Flow of advertisement forwardings:



T_A tables of routers after advertisement was forwarded through the network:

T_A (Router 1)	
Destination	**Filter**
-	-

T_A (Router 2)	
Destination	**Filter**
3	F

T_A (Router 3)	
Destination	**Filter**
1	F

T_A (Router 4)	
Destination	**Filter**
3	F

T_A (Router 5)	
Destination	**Filter**
4	F

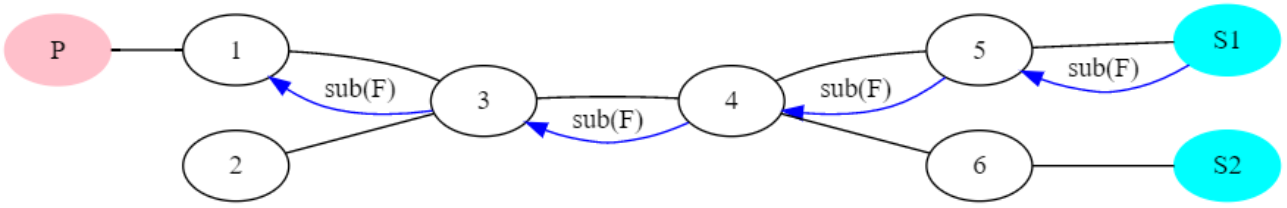
T_A (Router 6)	
Destination	**Filter**
4	F

Task 1.2: Subscriptions

Subscriber S1 sends a subscription to router 5.

- Step 1.1:
  - (Subscription of S1 is stored internally in router 5)
  - Lookup matching (D, F) pairs in T_A: Found pair (4, F)
    - Forward subscription to 4.
- Step 1.2:
  - Router 4 adds the subscription to its T_S table.
  - Lookup matching (D, F) pairs in T_A: Found pair (3, F)
    - Forward subscription to 3.
- Step 1.3:
  - Router 3 adds the subscription to its T_S table.
  - Lookup matching (D, F) pairs in T_A: Found pair (1, F)
    - Forward subscription to 1.
- Step 1.4:
  - Router 1 adds the subscription to its T_S table.
  - Lookup matching (D, F) pairs in T_A: None found
    - No further forwarding.

Flow of S1 subscription forwardings:



T_S tables of routers after subscription of S1 was forwarded through the network:

T_S (Router 1)	
Destination	**Filter**
3	F

T_S (Router 2)	
Destination	**Filter**
-	-

T_S (Router 3)	
Destination	**Filter**
4	F

T_S (Router 4)	
Destination	**Filter**
5	F

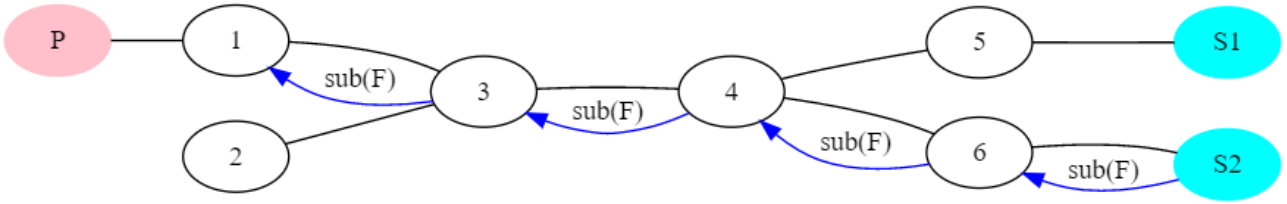
T_S (Router 5)	
Destination	**Filter**
-	-

T_S (Router 6)	
Destination	**Filter**
-	-

S2 sends a subscription to router 6

- Step 2.1:
  - (Subscription of S2 is stored internally in router 6)
  - Lookup (D, F) pairs in T_A: Found pair (4, F)
    - Forward subscription to 4.
- Step 2.2:
  - Router 4 adds the subscription to its T_S table.
  - Lookup (D, F) pairs in T_A: Found pair (3, F)
    - Forward subscription to 3.
- Step 2.3:
  - Router 3 already has the subscription in its T_S table
    - Algorithm *could* stop here - but on the slides there is no such condition.
  - Lookup (D, F) pairs in T_A: Found pair (1, F)
    - Forward subscription to 1.
- Step 2.4:
  - Router 1 already has the subscription in its T_S table
  - Lookup matching (D, F) pairs in T_A: None found
    - No further forwarding.

Flow of S2 subscription forwardings:



T_S tables of routers after subscription of S2 was forwarded through the network:

T_S (Router 1)

Destination	Filter
3	F

T_S (Router 2)

Destination	Filter
-	-

T_S (Router 3)

Destination	Filter
4	F

T_S (Router 4)

Destination	Filter
5	F
6	F

T_S (Router 5)

Destination	Filter
-	-

T_S (Router 6)

Destination	Filter
-	-

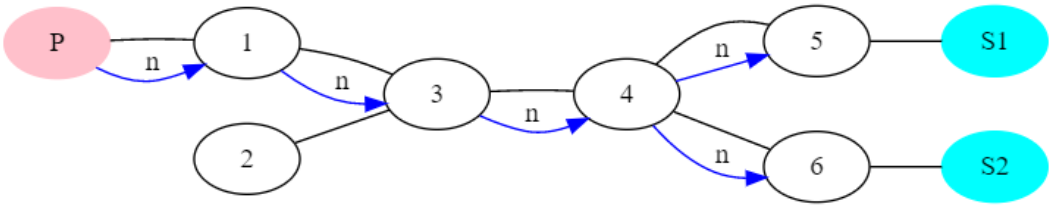
Task 1.3: Publisher P sends a notification to router 1

- Step 1
  - Router 1: Lookup matching (D, F) pairs in T_S table: Found pair (3, F)
    - Forward notification to router 3
- Step 2
  - Router 3: Lookup matching (D, F) pairs in T_S table: Found pair (4, F)
    - Forward notification to router 4
- Step 3
  - Router 4: Lookup matching (D, F) pairs in T_S table: Found pairs (5, F), (6, F)
    - Forward notification to router 5
    - Forward notification to router 6
- Step 4
  - Router 5: Lookup matching (D, F) pairs in T_S table: Found none

        - No further forward to brokers
        - Forward notification to client S1
- Step 5
    - Router 6: Lookup matching (D, F) pairs in T_S table: Found none
        - No further forward to brokers
        - Forward notification to client S2

Steps **4 and 5** may occur in arbitrary order.

Flow of notification forwardings:



Task 2: Routing in Publish/Subscribe Systems

Task 2.1

General considerations

	Pros	Cons
Routing with Subscriptions	• Brokers must not maintain a seperate table for advertisements • Publisher can send arbitary notifications	• Subscriptions are forwarded througout the whole network (Flooding) • Routing tabes for subscriptions are complex to manage as they are used for potentially a lot of subscriptions.
Routing with Advertisements	• More efficient routing of subscriptions. They are only forwarded to destinations where those type of notifications are produced.	• Publishers should know in advance what type of notifications they are producing • Overhead big if a producer sends a message only once

Matchmaking

We assume messages about the currently running game for this scenario (such as kill-death-ratio). A producer can not announce the messages that it will sent (no scores or similar figures that might be used as filter). Therefore a subscription based routing is superior. Routing with advertisements would require that the produce sends an advertisement for approximately each message because the game scores will change for every game. All in all advertisement is only useful if a meaningful filter is advertised.

Task 2.2

In general, "Routing with Advertisements" is more suiteable in applications where the producers know the content/type of the notifications they are producing in advance. This mechanism makes sense in appliations that have a wide variety of message types.

In turn "Routing with Subscriptions" is more suiteable in networks where publishers, in general, dont know the type of messages that they are producing in advance. Another scenario where this type of routing makes sense is when pubishers producing a lot of distinct messages, i.e. the messages are usally covered by a distinct filters/subscriptions.