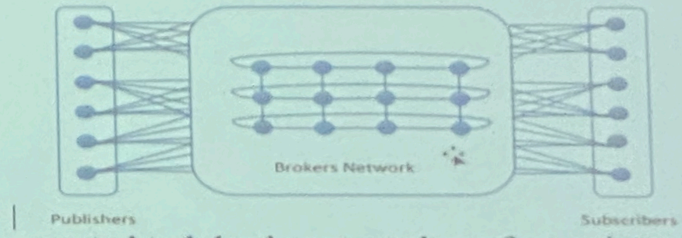
10 Dicembre 2019

Ex 1: Let us consider a distributed system composed by publishers, subscribers and brokers. Processes are arranged in a network made as follows and depicted below:



1. Each publisher is connected to k brokers trough perfect point-to-point links;
2. Each subscriber is connected to k brokers trough perfect point-to-point links;
3. Each broker is connected to k brokers trough perfect point-to-point links and the resulting broker network is k-connected [1](4-connected in the example);

Answer to the following questions:

1. Write the pseudo-code of an algorithm implementing the event-flooding dissemination scheme assuming that processes are not going to fail.
2. Discuss how many crash failures the proposed algorithm can tolerate.
3. Modify the proposed algorithm in order to tolerate f Byzantine processes in the broker network and discuss the relation between f and k.

Solution:

Punto_1

Init:

```
p_neigh = get_publisher_neighbours( )
b_neigh = get_broker_neighbours( )
s_neigh = get_subscriber_neighbours( )
subscriptions = empty

upon event publish(e)              %only for publishers
        for each pb belonging to b_neigh
                trigger pp2pSend(EVENT, e) to pb

upon event pp2pDeliver(EVENT, e)           %only for brokers
        for each pb belonging to b_neigh
                trigger pp2pSend(EVENT, e) to pb
        for sub belonging to subscriptions
                if subscriptions_matching(sub, e)
```

```
                    s = get_subscriber(sub)
                    trigger pp2pSend(NOTIFY, e) to s

upon event pp2pDeliver(NOTIFY, e)          %only for subscribers
        trigger notify(e)


upon event subscribe(subs)              %only for subscribers
        for each pb belonging to b_neigh
                trigger pp2pSend(SUBSCRIBE, subs, myID) to pb

upon event pp2pDeliver(SUBSCRIBE, subs, subId)
        subscriptions = subscriptions U {<subs, subId>}
```

Punto_2:

For sure K-1 crash failure the proposed algorithm can tolerate.