

# Distributed System

Stefano Cereda  
Leonardo Cella  
Alessandro Cimbelli

Politecnico di Milano

May 5, 2016

# Overview

## 1 Introduzione

- Traccia Scelta
- Assunzioni

## 2 Simulazione casistiche

- Client Leave/join
- Broker Leave/join
- Causal Consistency

## 3 Conclusione

# Traccia Scelta

Il progetto che abbiamo deciso di sviluppare è:

- Distributed, topic-based publish/subscribe with causal consistency

Il quale è stato realizzato tramite OmNet++.

Nel realizzare la traccia abbiamo fatto le seguenti assunzioni:

- Quando un broker smette di funzionare si comporta come un hub.
- Canali sicuri.
- Che la rete simulata sia quella di un overlay network e che sia aciclica per costruzione.

# Client Leave/join

Vediamo adesso i punti salienti della simulazione della client leave/join:

- Fase di inizializzazione della rete.
- Funzionamento *normale* della rete con scambio di messaggi.
- *Client1* performa una leave.
- La rete aggiorna le proprie tabelle di routing considerando che il *client1* si sia disconnesso.
- *Client* pubblica messaggi che sarebbero arrivati al *client1* ma che adesso, giustamente, non gli giungono più.
- Il *client1* torna a far parte della rete con una join.
- Il *client1* riceve adesso correttamente i messaggi a cui era interessato.



# Broker Leave/join

Vediamo adesso i punti salienti della simulazione della broker leave/join:

- Fase di inizializzazione della rete.
- Funzionamento *normale* della rete con scambio di messaggi correttamente inoltrati secondo le subscriptions.
- *Broker* performa una leave ed entra in *hub mode*.
- La rete continua a funzionare senza il contributo di *broker*.
- *Broker* effettua una join e, i broker vicini, gli comunicano quali sono le subscriptions a cui è interessato.
- La rete torna a funzionare come prima della leave, con un corretto inoltro dei messaggi da parte di *broker*.





# Causal Consistency

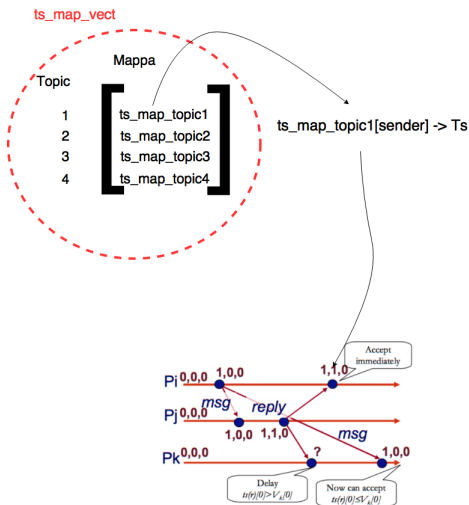


Figure: File Causal Consistency structure

# Causal Consistency Comparison

- Per la *Causal Consistency* abbiamo utilizzato dei *vector clock* la cui lunghezza varia dinamicamente con il comportamento della rete. Infatti nel nostro modello il numero di client può variare dinamicamente ed non è noto a priori. Le regole che governano i *vector clock* per ottenere la *causal consistency* restano immutate ad eccezione della regola:

$$ts(r)[i] \leq V_k \quad \forall i \neq j$$

La regola generale descritta sopra è estesa nel caso si verifichino i seguenti casi:

- ▶ Se il messaggio contiene timestamp di clients non conosciuti dal ricevente, allora esso non sarà *causal consistent*. (Potremmo dover ancora ricevere dei messaggi dai suddetti clienti)
- ▶ Se il messaggio contiene timestamp di clients conosciuti solo dal ricevente, questo non influenzerà la consistenza.

# Fine