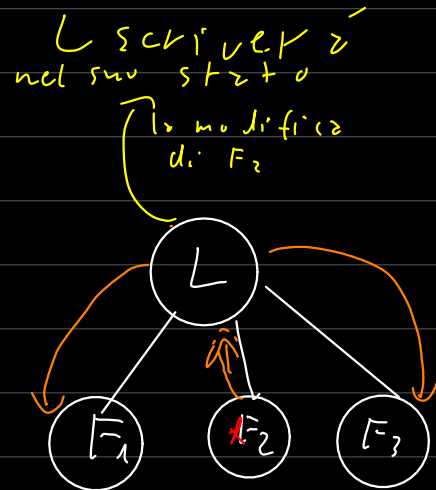
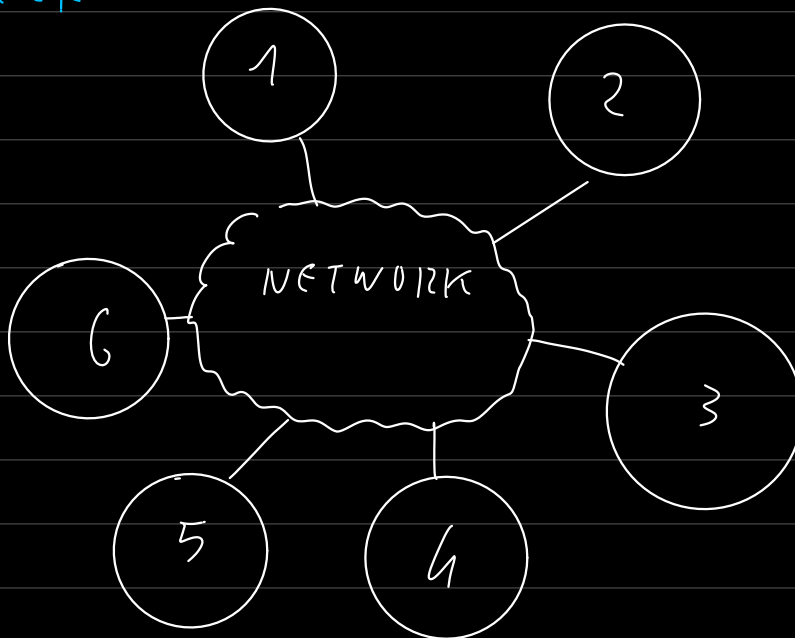


RAFT - ARCHIVIAZIONE

OBBIETTIVI:

- 1- CREAZIONE SISTEMA DI ARCHIVIAZIONE
DISTRIBUITO SENZA CENTRALIZZAZIONE (CLUSTER)
 - 2- UPLOAD FILES (write)
 - 3- DOWNLOAD FILES (read)
 - 4- DELETE FILES (delete)
-

1) CLUSTER

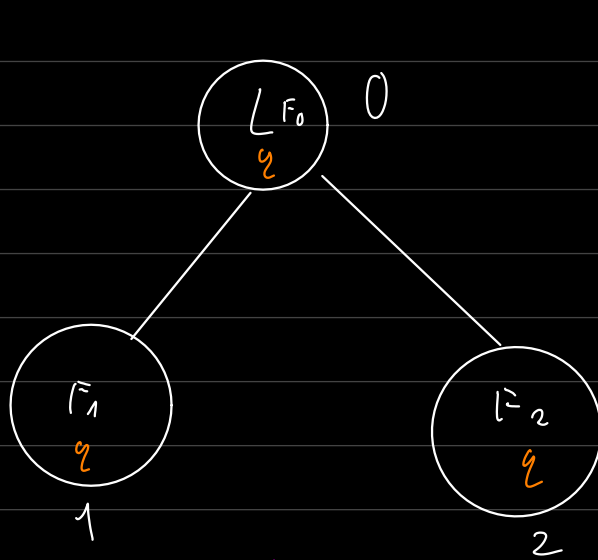


INVARIANTI:

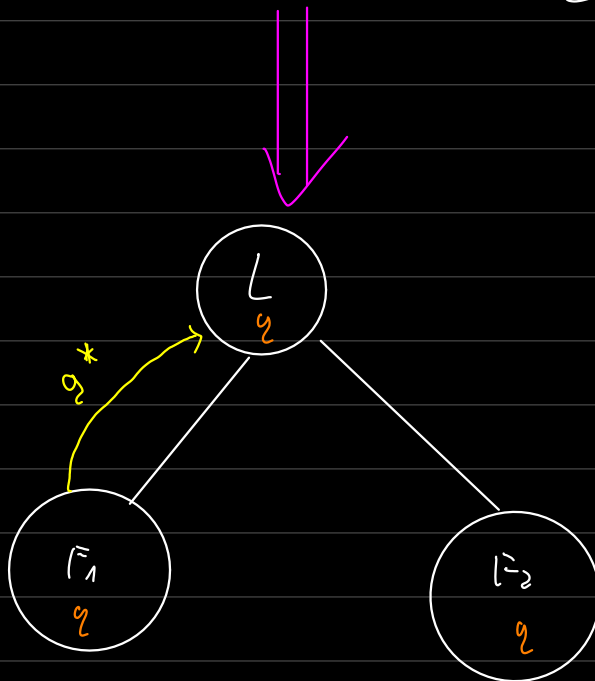
- 1- Tutti i nodi convergono allo stesso stato (a un certo punto avranno gli stessi dati)

⇒ RAFT

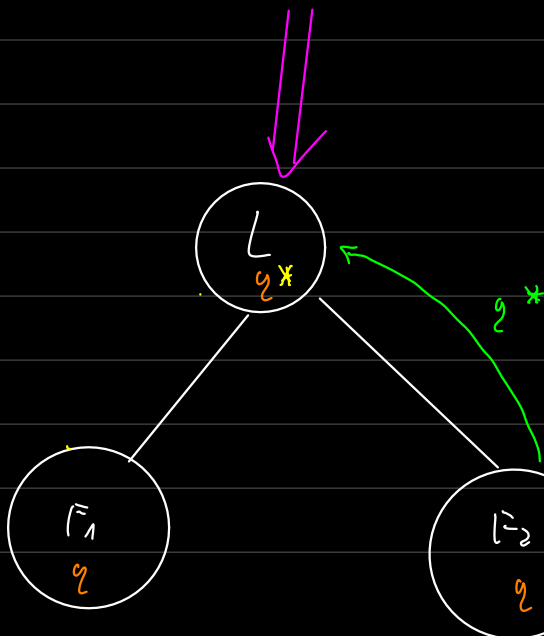
CRITICITY



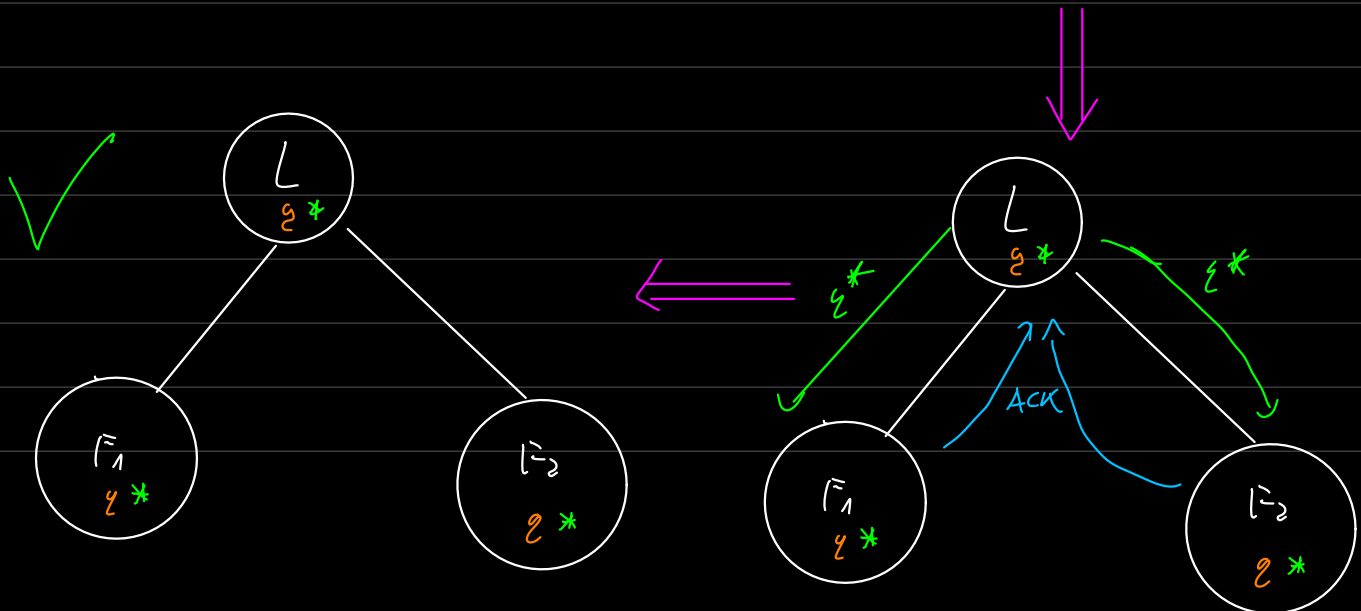
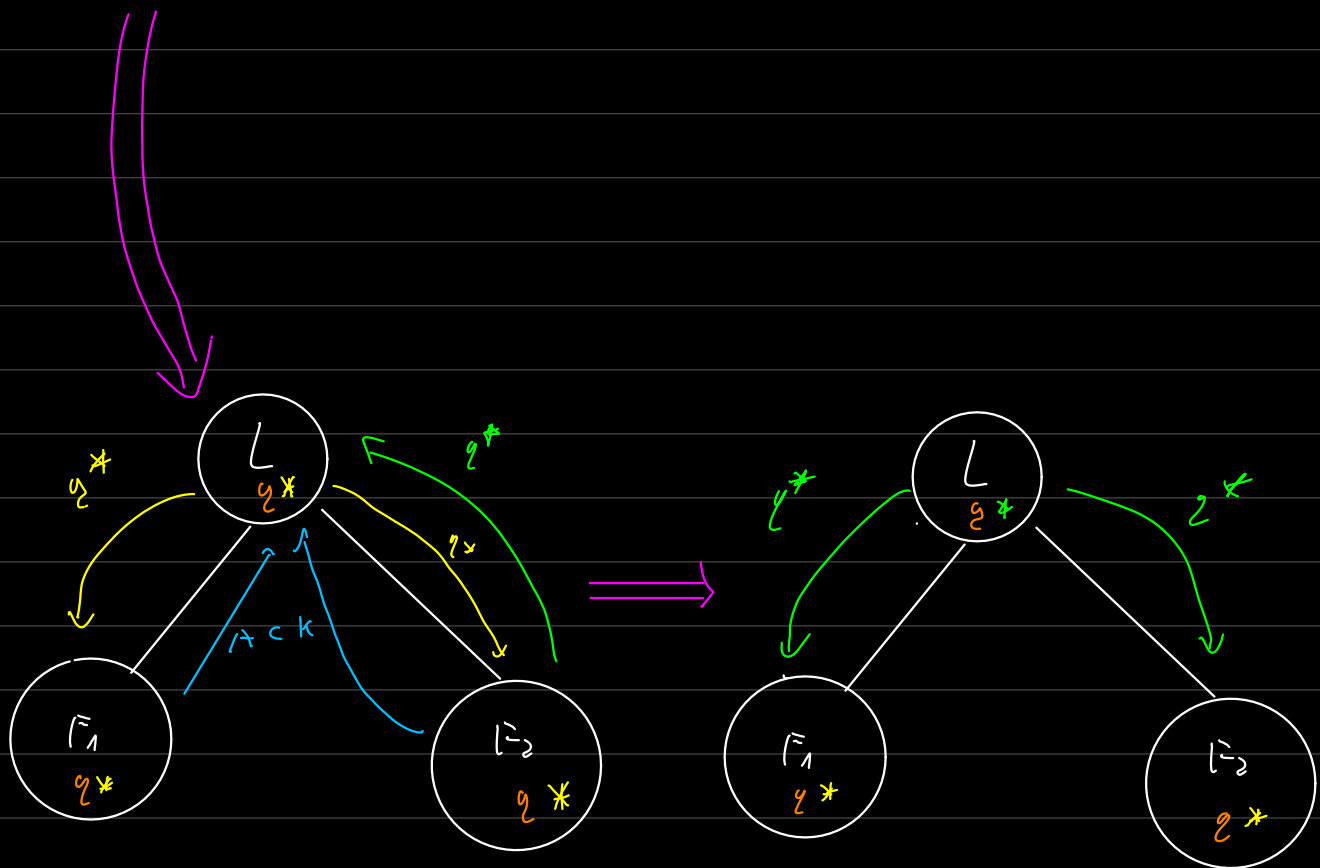
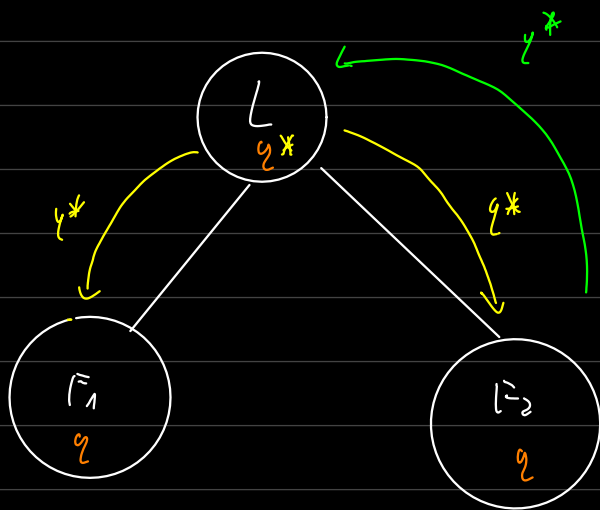
$\models L \in 1 \approx \text{?}$
 $\approx \text{"hello world"}$



$g_2^* = \text{"hello pluto"}$



$g_2^* = \text{"hello propetino"}$



Definizione dei nodi:

- VIRTUAL MACHINE:

Mantengo forte isolamento dei nodi, mantengo indipendenza dei nodi, mantengo potenziali grosse differenze tra i nodi, non ho, possibilmente, dipendenze comuni tra i nodi.

Comunicazione:

- Tramite HTTP:

- formato invio dati: **JSON**

STORAGE:

- NO FILESYSTEM DISTRIBUTO

- ogni nodo, internamente, deve avere un filesystem per funzionare

- lo STATO interno di ogni nodo

è una directory con all'interno

tutti i file dove ogni file

avrà un **ID** che verrà generato

a partire dalla hash di:

virtual directory + nome del
di storage file

LEADER ELECTION:

Ogni nodo ha un ID

⇒ evoluzione di Bully algorithm:

ogni nodo ha due timeout:

- 1- timeout heartbeat leader: unico per tutti
- 2- timeout invio richiesta elezione: random su ogni nodo.
↳ indica quanto tempo un nodo deve attendere prima di inviare una richiesta di elezione

Poi ogni nodo:

- se non ha indetto un'elezione accetta e invia a tutti l'ID della persona che vota. Ogni nodo conta chi ha più voti e quando c'è una maggioranza il nuovo leader è eletto

