

PROGETTAZIONE ED IMPLEMENTAZIONE DI UN AMBIENTE DI EMULAZIONE DI SISTEMI DI EDGE COMPUTING E RETI MOBILI 5G

Riccardo Carissimi

Relatore: CHRISTIAN QUADRI



**onnets
LAB**

UNIVERSITÀ DEGLI STUDI DI
MILANO



Contesto

LE TECNOLOGIE DI EDGE COMPUTING

L'evoluzione delle tecnologie di **edge computing** e delle reti mobili 5G sta rivoluzionando il panorama dell'elaborazione distribuita.

Per **studiare** meglio queste tecnologie è necessario:

- comprenderne il **funzionamento** e le **prestazioni**
- porre particolare attenzione agli **algoritmi di scheduling**, responsabili della gestione delle risorse e del bilanciamento del carico

Obiettivi della tesi

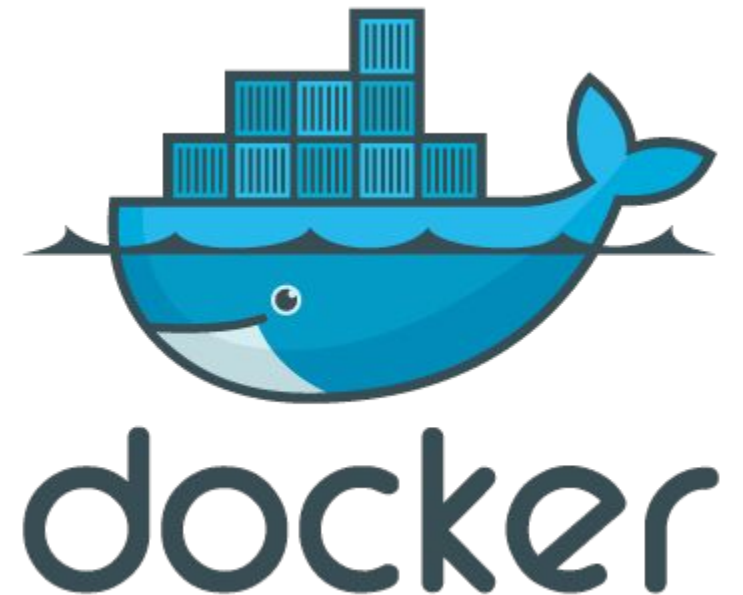
- progettare e implementare un **ambiente di emulazione** per reti 5G e sistemi di edge computing che permetta di studiarne il comportamento
- **analizzare i vari algoritmi di scheduling** proposti per capirne le caratteristiche e valutare l'impatto sull'architettura
- **simulare la distanza geografica** tra i nodi grazie al sottosistema di rete del kernel Linux

Tecnologie usate

I *container* sono tecnologie che consentono di impacchettare e **isolare le applicazioni** col loro intero ambiente di runtime.

In questo modo è facile **spostare l'applicazione** contenuta da un ambiente all'altro mantenendo tutte le funzionalità.

La tecnologia di containerizzazione più diffusa è **Docker**.



Tecnologie usate



Kubernetes è un sistema open-source per **l'orchestrazione di container** sviluppato da Google nel 2014.

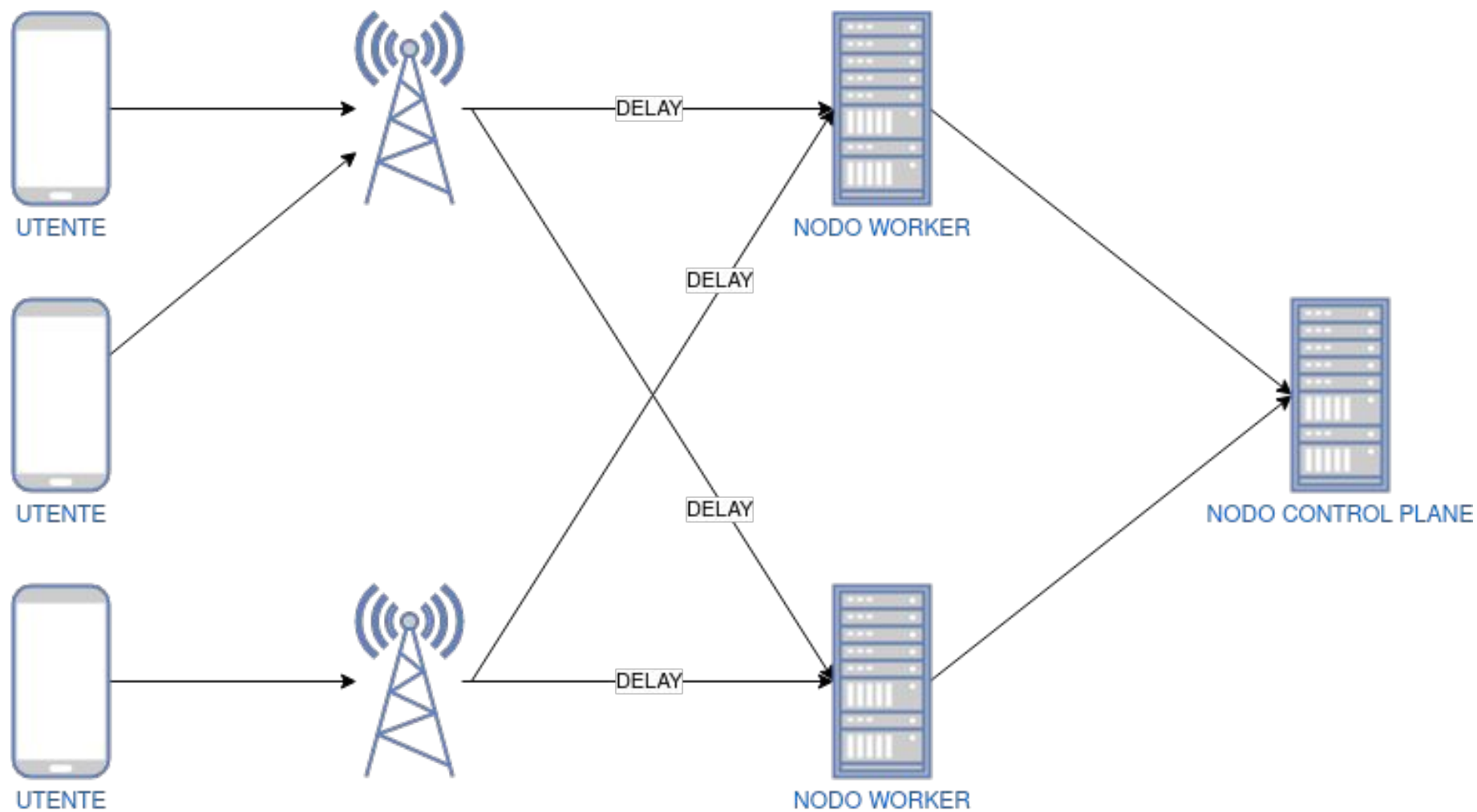
È lo standard *de facto* per la gestione dei container.

PROXMOX

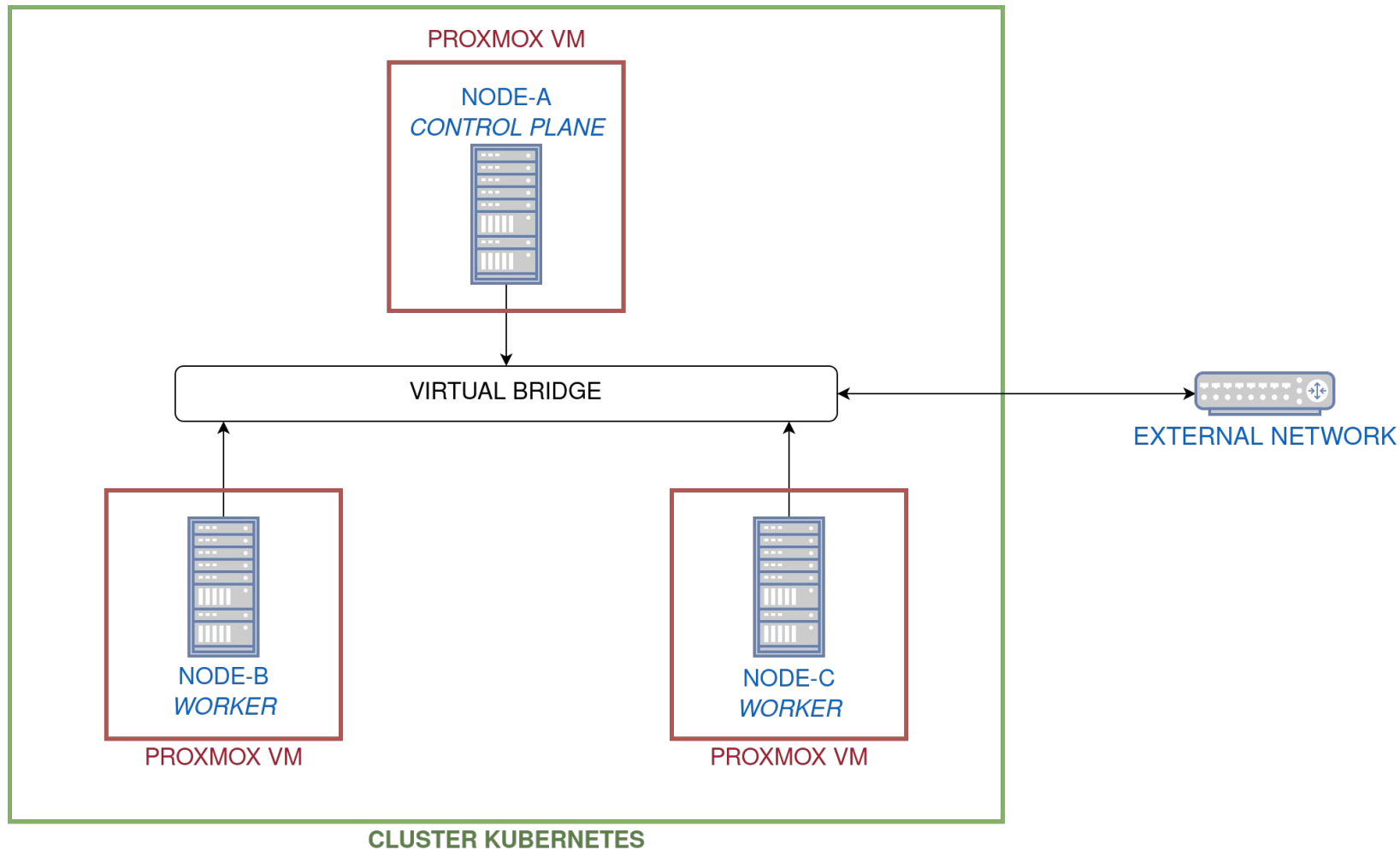
Proxmox è un **hypervisor** open source che sfrutta la funzionalità **KVM** del kernel Linux per la virtualizzazione.

Permette la gestione semplice delle VM in ambito cloud.

Progettazione



Implementazione



- L'insieme dei nodi è un cluster Kubernetes
- Ogni nodo è una macchina virtuale
- I nodi comunicano tramite un bridge virtuale
- Ogni VM ha a disposizione 4 core e 4 GB di memoria RAM

Gli Algoritmi di Scheduling

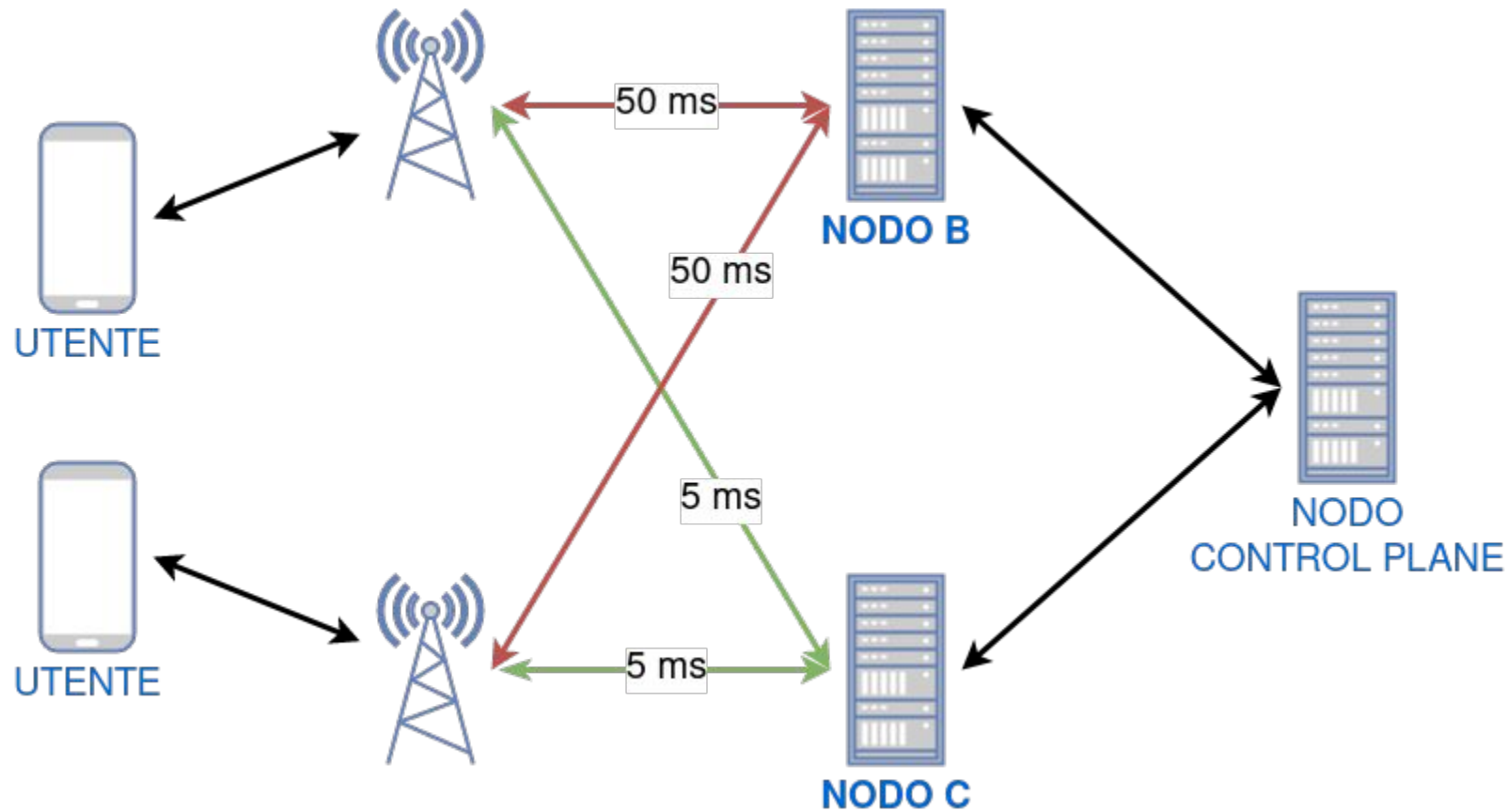
IMPLEMENTATI DA KUBERNETES TRAMITE IPVS

- *Round-Robin* (rr)
distribuisce le richieste equamente tra i nodi
- *Least Connections* (lc)
assegna le richieste al nodo con numero minore di richieste attive
- *Shortest Expected Delay* (sed)
assegna le richieste al nodo con la latenza attesa minore

$$L_s = \frac{C_i + 1}{U_i}$$

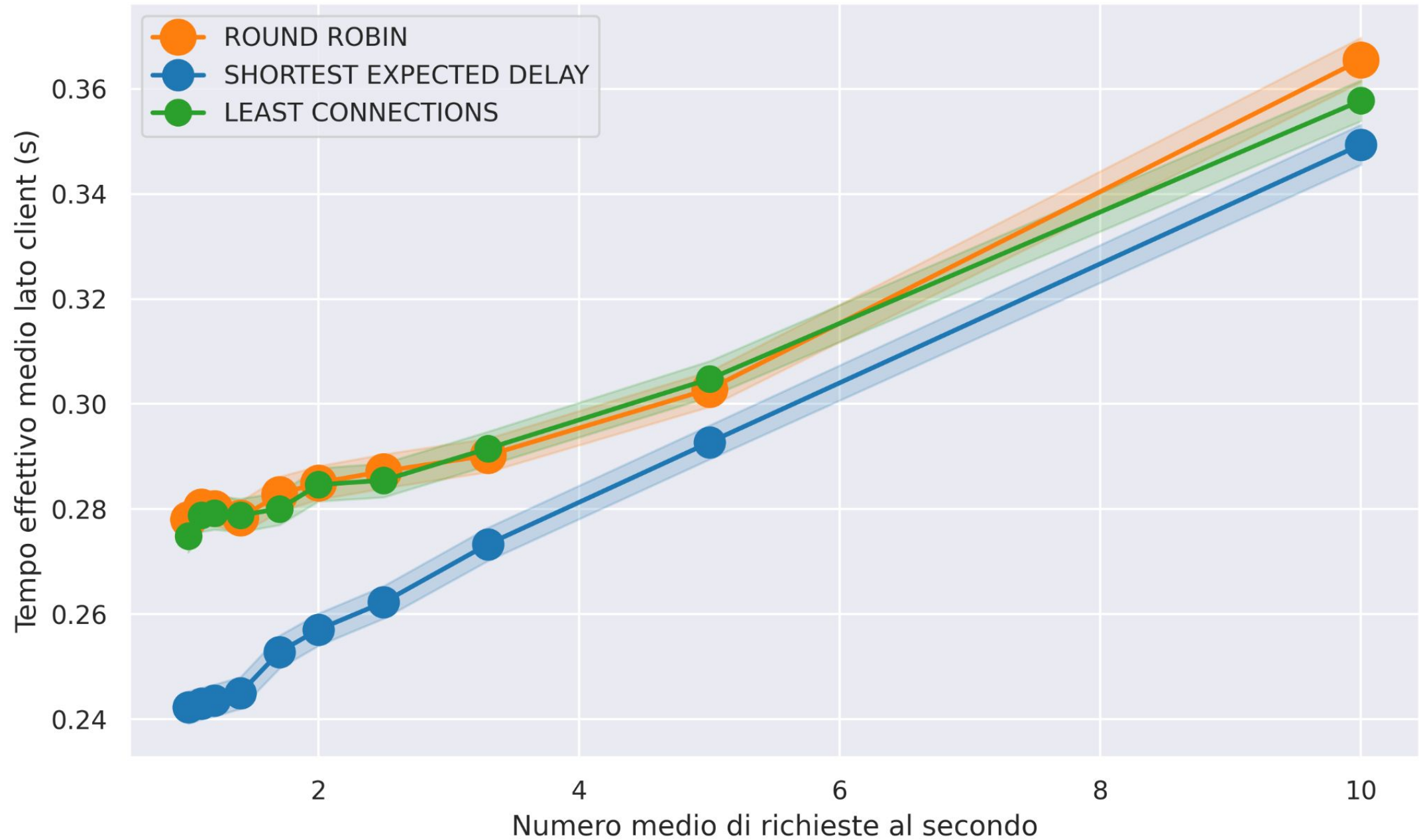
Dato C il numero di richieste attive e U il peso assegnato a ciascun server, la latenza è stimata come

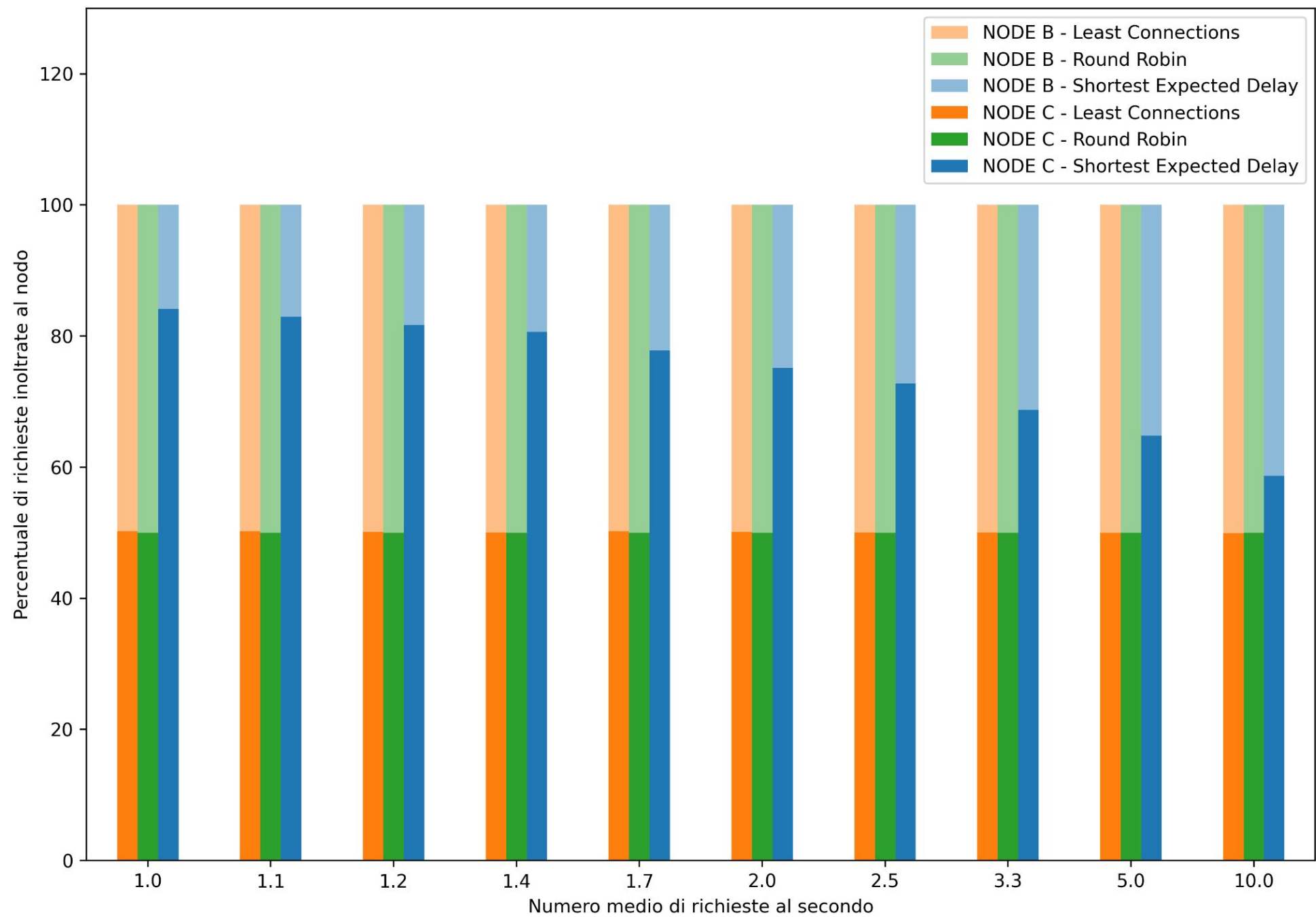
Scenario di emulazione



Algoritmi di scheduling	RR, LC, SED
Richieste per scenario	100 per run 50 run
Frequenza delle richieste	da 1 a 10 Hz

Confronto tra tempo effettivo e numero di richieste al secondo





Conclusioni

Grazie alle tecnologie di Kubernetes e Proxmox abbiamo creato un **ambiente di emulazione configurabile** e flessibile che consente l'analisi di sistemi di edge computing eterogenei.

L'introduzione del ritardo artificiale sui link della rete ci ha permesso di simulare la distanza geografica tra i nodi e di studiare gli **algoritmi di scheduling**:

- *shortest expected delay* → adatto ad ambienti **geograficamente distribuiti**
- *round-robin* e *least connections* → hanno prestazioni migliori con nodi dal delay paragonabile e li possiamo usare in quelle situazioni in cui la **sensibilità alla latenza è minore** o si vuole distribuire equamente il carico tra i nodi

È possibile espandere il progetto usando il sistema di emulazione OMNeT++ col framework *Simu5G*, che permettono di considerare la **simulazione dei punti di accesso** e della rete 5G.



Grazie per l'attenzione