

Benchmarking Libreria ZeroMQ

Analisi delle prestazioni di un nodo ZeroMQ vs ROS2

Martina Carella

Libreria ZeroMQ

<https://zeromq.org/>

- Libreria open source di messaggistica asincrona, veloce e scalabile
- **Multi-transport** -> in-process, inter-process, TCP, UDP, multicast, WebSocket
- **Messaging Patterns** -> pub-sub, push-pull, client-server, request-reply
- **Brokerless** -> non richiede un message broker dedicato
- **Socket**
 - A. Gestiscono *automaticamente* le connessioni in modo trasparente all'utente
 - B. Inizia *automaticamente* ad accettare le connessioni se collegata a un endpoint
 - C. Si riconnette *automaticamente* nel caso in cui la connessione vengano interrotte

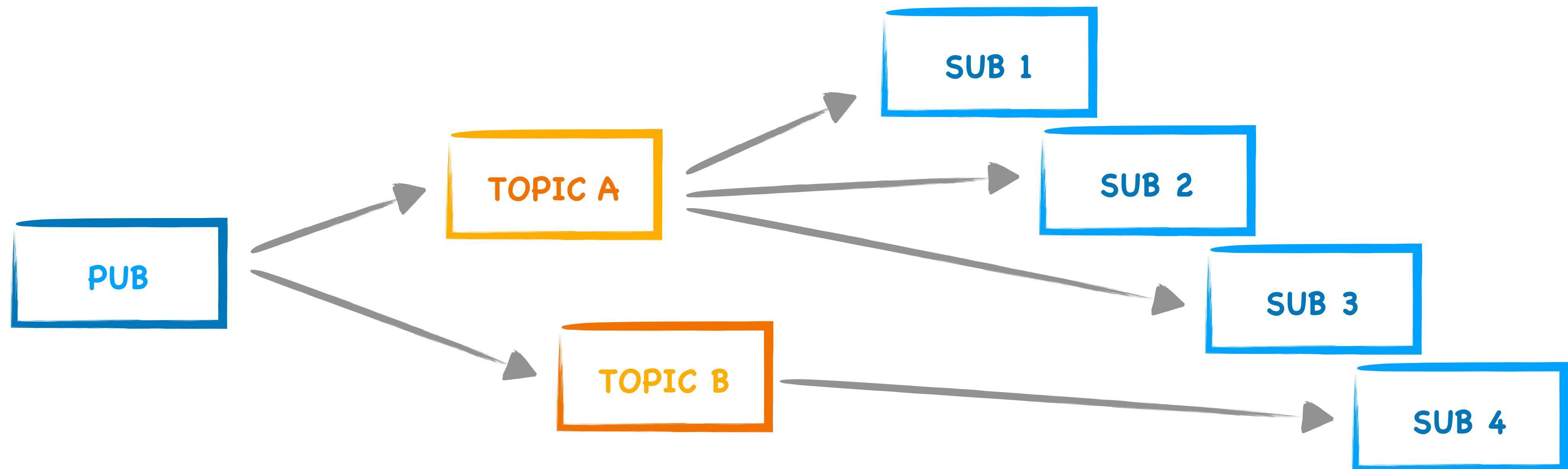
Robot Operating System (ROS2)

<https://www.ros.org/>

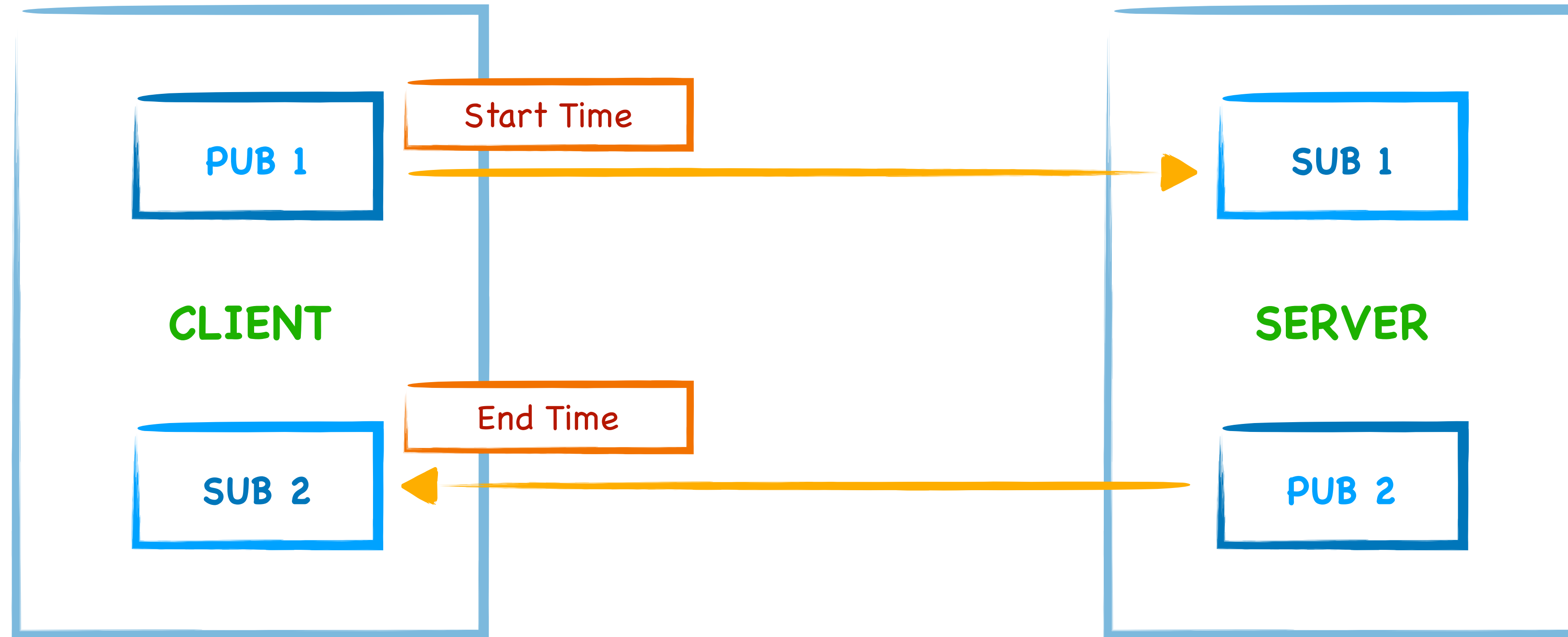
- Tools e librerie open source per la creazione di applicazioni complesse in ambito robotico
- Comunicazione tra nodi (processi) e un hub centrale
- Diversi livelli di affidabilità
- **DDS (Data Distribuite System)**
 - A. Protocollo che sostituisce TCP/UDP in ROS2
 - B. Elevato livello di configurazione
 - C. La sua complessità viene nascosta esponendo un'astrazione per l'utilizzo delle funzionalità più importanti

Pattern PUB-SUB

- I nodi si scambiano direttamente dei messaggi
- Il nodo **publisher** (PUB) **pubblica** il messaggio all'interno di un **topic**
- I nodi **subscriber** (SUB) possono **isciversi** al topic per ricevere i messaggi



Benchmarking



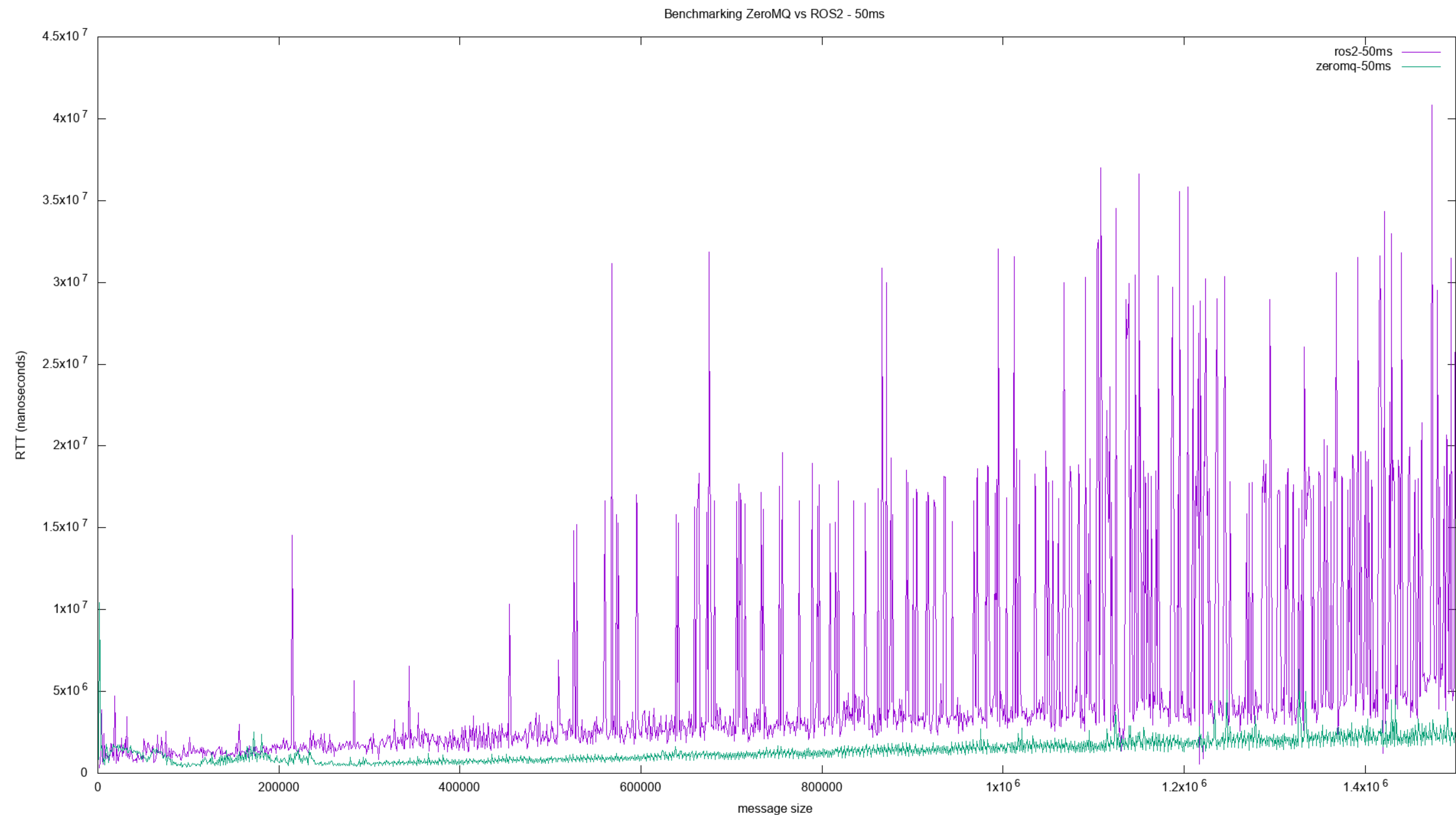
Round Trip Time = End Time - Start Time



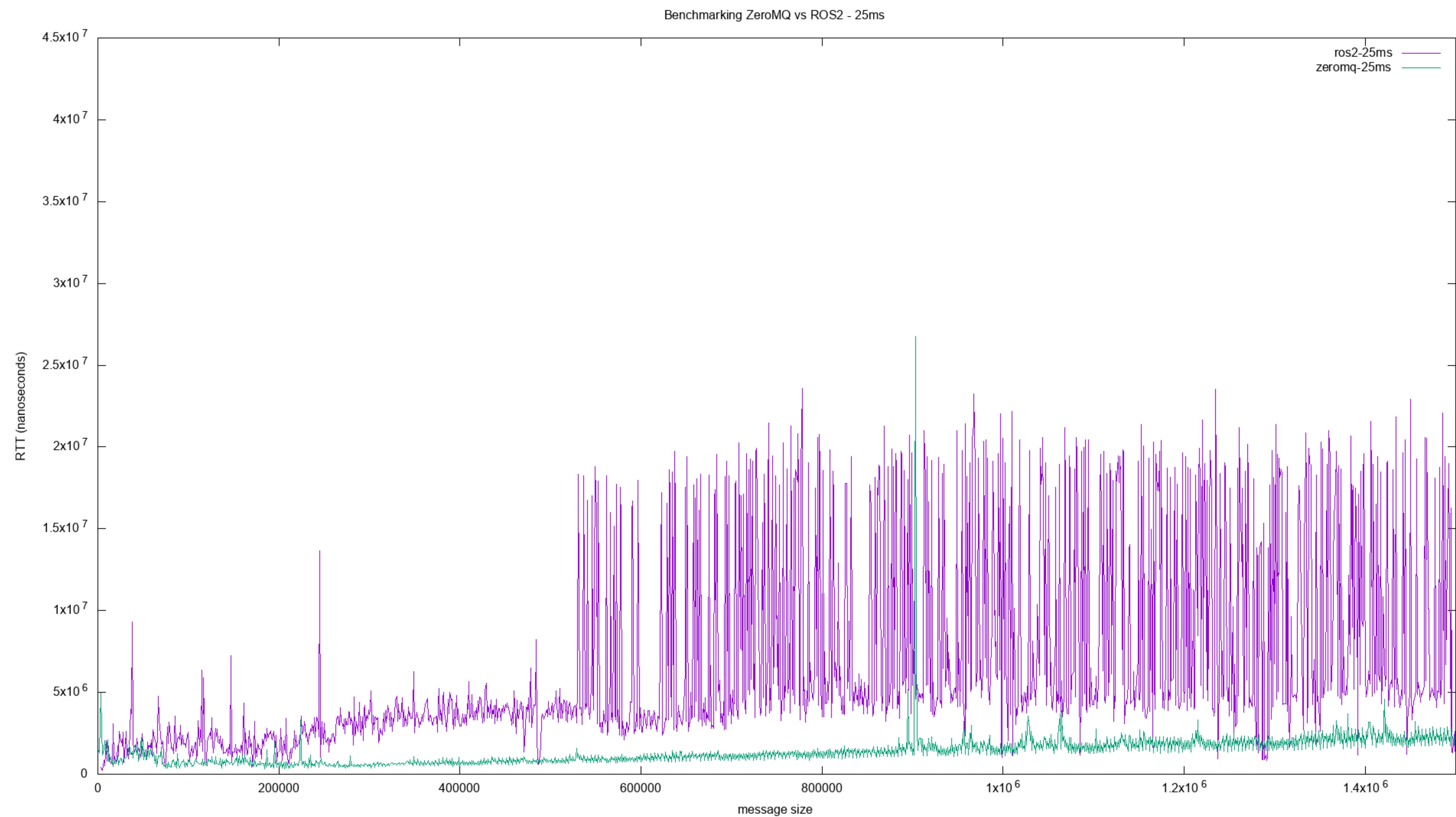
Benchmarking

- Creazione di un nodo PUB e un nodo SUB
- Ciclo eseguito per 1500 volte:
 1. Il client **invia un messaggio** al server aumentando di 1Kb la sua dimensione ad ogni invio
 2. Viene segnato il **tempo di invio** del messaggio
 3. Il server, ricevuto il messaggio, **invia una risposta** al client
 4. Il client, ricevuta la risposta, segna il **tempo di ricezione**
 5. Il tempo finale viene **salvato sul file time.txt**

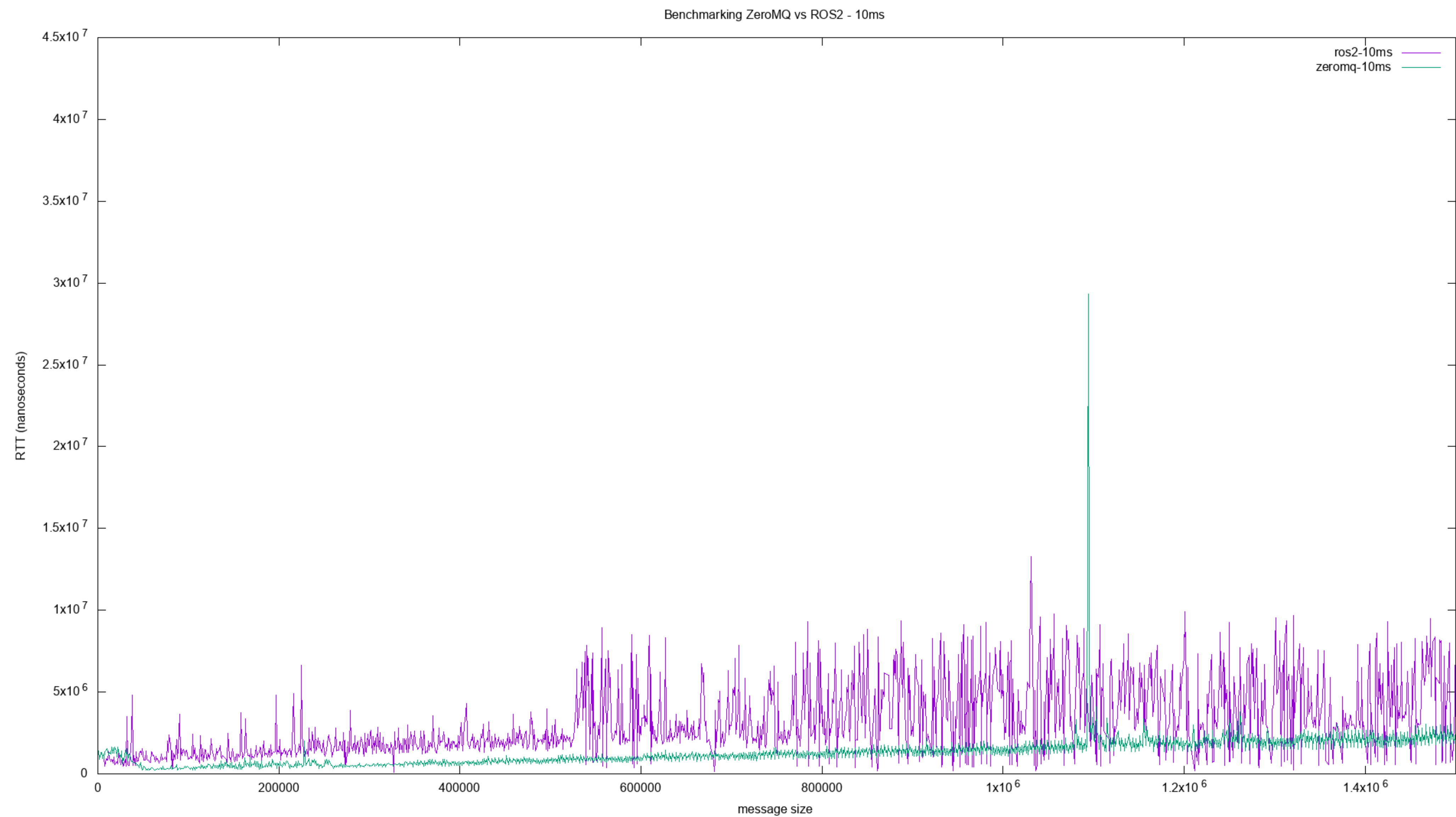
Messaggi inviati ogni 50ms



Messaggi inviati ogni 25ms

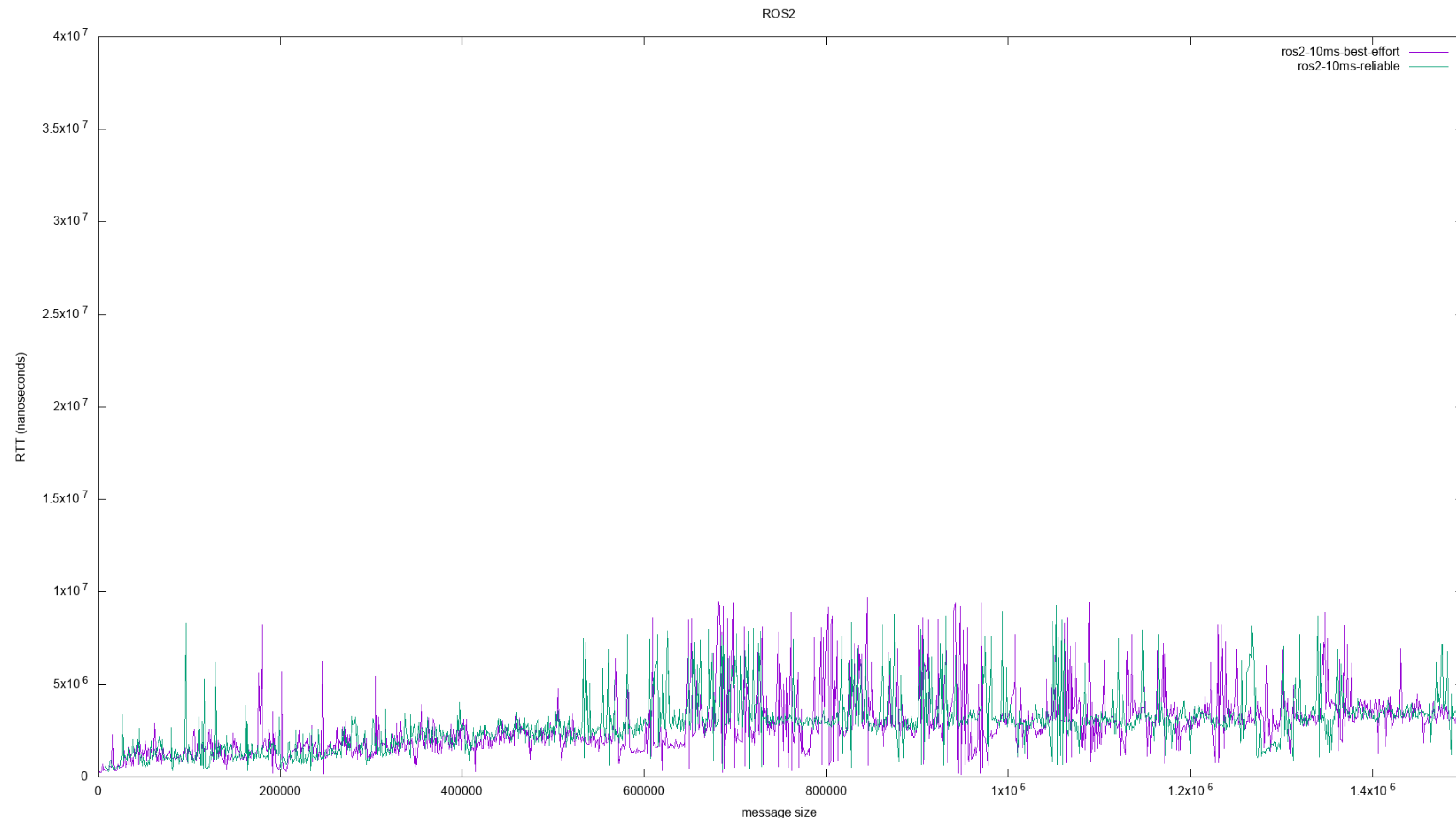


Messaggi inviati ogni 10ms



QoS ROS2

- Profilo di default per PUB-SUB
 - **History**: 'keep last' con coda = 10
 - **Reliability**: 'reliable'
- Profilo testato:
 - **History**: 'keep last' con coda = 0
 - **Reliability**: 'best-effort'



Conclusioni

- Tendenzialmente, con l'aumento della dimensioni dei messaggi, **ZeroMQ** risulta avere un **RTT minore** rispetto a ROS2
- Nella **fase iniziale** della comunicazione, ROS2 risulta avere prestazioni migliori, ma dopo aver instaurato correttamente la connessione, ZeroMQ diventa più veloce
- ROS2 risulta essere penalizzato, probabilmente, a causa del carico di lavoro sui nodi, essendo un framework e non una libreria come ZeroMQ
- La variazione delle QoS non ha portato a visibili differenze tra i risultati dei due test.