

RETI PER L'AUTOMAZIONE INDUSTRIALE

Ingegneria Informatica LM-32

**Simulation
of
Wireless Sensor Network
using OMNeT++**

Autori:

Gianluca Grasso, Rino Di Paola

2020/2021

Indice

1	Introduzione	1
2	Scenario	2
3	Scelte Implementative	3
3.1	Messaggi	3
3.1.1	Ritrasmissione messaggio	3
3.2	Disposizione nodi	3
3.3	Pure-ALOHA	3
3.4	Consumo Batteria	4
3.5	Tempo negli stati di idle, tx e rx	4
3.6	Carica residua della batteria	5
3.7	Scaricamento completo nodi	5
4	Statistiche	6
5	Risultati	7
5.1	Configurazione 10 nodi	7
5.2	Configurazione 50 nodi	9
5.3	Configurazione 100 nodi	10
5.4	Configurazione 150 nodi	10
5.5	Configurazione 200 nodi	11
6	Conclusioni	12
	Elenco delle figure	14

1 Introduzione

In questo elaborato viene simulato uno scenario di una rete wireless, in cui vi sono dei nodi stazionari che scambiano dei messaggi.

La rete è costituita da n nodi trasmettitori, chiamati *source*, e da un nodo ricevitore, chiamato *sink*. I nodi sono disposti in maniera tale che ogni nodo si trovi nello stesso raggio di copertura degli altri nodi. Tutti i nodi trasmettitori, utilizzando la stessa distribuzione, inviano messaggi di tipo aperiodico. Sono state simulate diverse reti, costituite da un numero di nodi pari a 10, 50, 100, 150 e 200, escluso il sink. Ogni nodo deve riportare la durata della propria batteria fino al momento in cui si esaurisce e, inoltre, deve calcolare il tempo che trascorre negli stati di trasmissione, di ricezione e di idle.

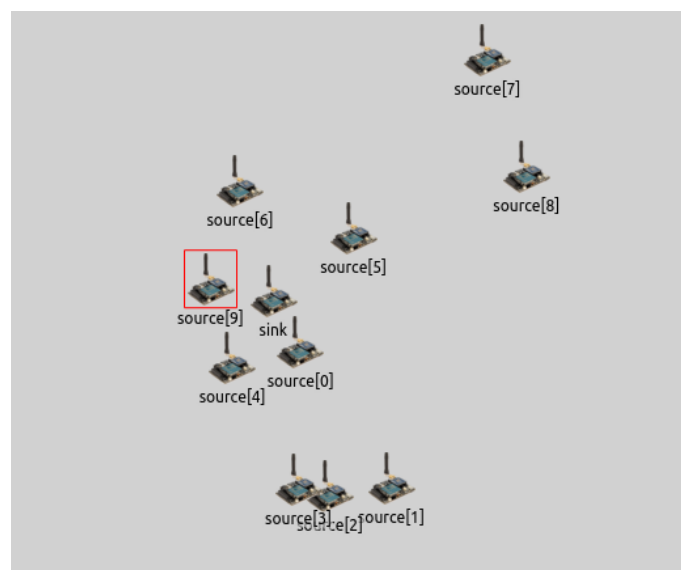


Figura 1: Configurazione rete con 10 nodi

2 Scenario

La rete descritta in questo elaborato è stata sviluppata a partire dal progetto *Simple Wireless*, utilizzando le librerie basate su INET.

Come già anticipato, tutti i nodi trasmettitori inviano messaggi di tipo aperiodico (di dimensione 60 byte) al sink utilizzando la stessa distribuzione. Nello specifico, è stata applicata una distribuzione di tipo uniforme, la quale attribuisce la stessa probabilità a tutti i valori appartenenti ad un dato intervallo $[a,b]$. Tale distribuzione è parametrizzabile.

È stata implementata una strategia di ritrasmissione, impostando inoltre il numero massimo di ritrasmissioni che un nodo trasmettitore deve effettuare nel caso in cui non venga ricevuto dal sink un ack relativo ad uno specifico messaggio entro un intervallo di tempo configurato (100 ms).

Il modello di canale (path loss) è stato settato a *LogNormalShadowing*, fissando i valori di $\alpha = 4.03$ e $\sigma = 4.98$. Il modello di canale viene utilizzato per calcolare la diminuzione della potenza del segnale radio man mano che si propaga, allontanandosi dal trasmettitore. In particolare, LogNormalShadowing è una configurazione in cui i livelli di potenza seguono una distribuzione lognormale; è utile per modellare i fenomeni di shadowing, causati da oggetti come gli alberi.

Ogni nodo utilizza un modulo apposito per mappare i consumi di energia; in particolare, viene utilizzato il modulo *SensorStateBasedCcEnergyConsumer*. Quest'ultimo determina i consumi dell'energia in base allo stato assunto dal modulo radio del nodo, combinando diversi stati: stato della radio, stato del trasmettitore, stato del ricevitore. Il consumo viene fornito in Ampere.

Il protocollo di accesso al canale sviluppato è Pure-ALOHA, in cui ogni nodo trasmette appena ha necessità di trasmettere, senza ascoltare il canale. Come mostrato in figura, se l'intervallo di tempo durante il quale due o più stazioni trasmettono si sovrappone, allora si avrà una collisione.

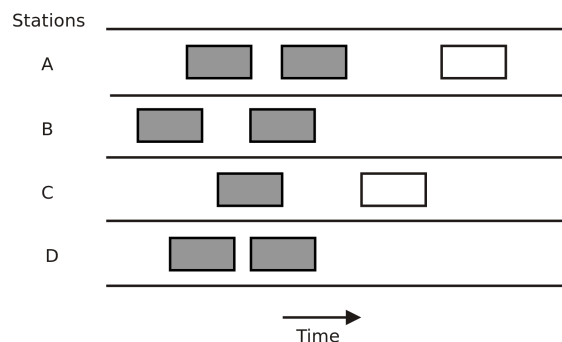


Figura 2: Protocollo Aloha: in grigio le frame in collisione

3 Scelte Implementative

3.1 Messaggi

Per far sì che i messaggi abbiano dimensione 60 byte è sufficiente impostare il campo *payload*, contenuto nell'applicazione di tutti i nodi, a 60B.

Per simulare la trasmissione di messaggi aperiodici, è stata dichiarata la variabile *period* come *volatile* ed è stata settata ad un valore random, all'interno di uno specifico intervallo, tramite la funzione *uniform*. Il modificatore *volatile* indica che un parametro viene ricalcolato ogni volta che viene richiesto un valore durante la simulazione.

3.1.1 Ritrasmissione messaggio

Per implementare una strategia che ritrasmetta il messaggio nel caso in cui non è stato ricevuto un ack dopo 100 ms, è stato necessario modificare il parametro *macAckWaitDuration* del modulo MAC. Invece per impostare il numero massimo di ritrasmissioni, viene usato il parametro *macMaxFrameRetries*.

3.2 Disposizione nodi

Per disporre i nodi in maniera uniforme e in modo tale che siano nello stesso raggio di copertura è stato utilizzato il sotto modulo mobility del nodo, modificandolo opportunamente tramite il file *omnetpp.ini*. I nodi sono stati disposti sufficientemente vicini fra di loro, poiché il modello di canale e i relativi parametri utilizzati rendono il canale particolarmente rumoroso.

3.3 Pure-ALOHA

L'implementazione del protocollo di accesso al mezzo basato su Pure-ALOHA è stata effettuata sostituendo il modulo MAC di default (*Ieee802154Mac*, che simula il protocollo di accesso *CSMA/CA*) con un modulo apposito, che è stato chiamato *Aloha-IEEE802154Mac*. Il codice di quest'ultimo è stato sviluppato a partire dal codice del modulo di default, apportando delle modifiche (ovvero, eliminando il controllo che veniva effettuato sul canale prima di trasmettere). In particolare, il *ccaDetectionTime* (tempo di ascolto del canale da parte del nodo che deve trasmettere) e il periodo di *backoff* (tempo random che aspetta il nodo prima di trasmettere) sono stati posti uguali a 0. In questo modo, ogni nodo trasmette quando ha necessità di trasmettere, senza ascoltare il canale.

3.4 Consumo Batteria

Dopo aver impostato l'energyConsumer della radio del nodo settandolo tramite typename a *SensorStateBasedCcEnergyConsumer*, è stato impostato come energyStorage il modulo *SimpleCcBattery*. Per impostare i parametri dell'energy storage, si è fatto riferimento a una batteria di comune utilizzo. All'interno del datasheet troviamo le caratteristiche della batteria: nello specifico, la tensione nominale (3.7 V), la capacità nominale (2000 mAh, che equivale a 7200 C) e la resistenza interna (160 mΩ). Inoltre, poiché il programma segnalava un errore dovuto al fatto che non riusciva a trovare l'energyStorage, è stato necessario modificare opportunamente il modulo Node ed aggiungere un sottomodulo *status* di tipo *NodeStatus*.

Datasheet: <http://www.farnell.com/datasheets/2369105.pdf>



Figura 3: Batteria utilizzata

3.5 Tempo negli stati di idle, tx e rx

Per calcolare il tempo trascorso da un nodo negli stati di idle, tx e rx, è stato modificato appositamente il file *Aloha_IEEE802154Mac.cc* e più nello specifico la funzione *receiveSignal*. Infatti, non appena viene emesso un segnale di cui è già stato fatto il *subscribe*, viene chiamata questa funzione e si entra in uno specifico if, a seconda che il segnale ricevuto sia di tipo *TrasmissionStateChangedSignal* o *ReceptionStateChangedSignal*. Nel caso in cui arrivi un segnale di tipo *TrasmissionStateChangedSignal*, se il nodo sta iniziando in questo momento la trasmissione, viene salvato l'istante in cui il nodo inizia a trasmettere *lastTxTime* tramite la funzione *simTime()*; se, invece, il nodo ha appena finito di trasmettere, viene calcolato il tempo di trasmissione *tx* effettuando la differenza fra il tempo attuale e quello salvato precedentemente

(*lastTxTime*); *tx* viene poi sommato al tempo di trasmissione totale *txTime*. È stato adottato lo stesso procedimento per calcolare il tempo di ricezione, quando ad arrivare è un segnale di tipo *ReceptionStateChangedSignal*. Infine, il tempo di *idle* viene calcolato sottraendo al tempo attuale *txTime* e *rxTime()* (*rxTime* contiene anche il tempo in cui lo stato di ricezione è busy, ovvero il tempo in cui il modulo radio del nodo è in ricezione ma il segnale non è abbastanza potente per essere ricevuto). Ogni volta che viene calcolato il tempo trascorso in ognuno di questi tre stati, viene emesso un segnale (tramite la funzione *emit*) opportunamente definito.

3.6 Carica residua della batteria

La carica residua della batteria di un nodo viene calcolata di default nella funzione *updateResidualCapacity* del file *SimpleCcBattery.cc*.

3.7 Scaricamento completo nodi

Per risolvere un “crash” improvviso che si presentava nel momento in cui si scaricava il primo nodo, si è settato lo stato del Nodo a “*DOWN*” e lo stato della radio a “*RADIO_MODE_OFF*”. Per far questo è stato necessario creare il modulo *MySimpleCcBattery*, che estende il modulo di inet *SimpleCcBattery*. Di questo modulo è stato fatto l’override dei metodi *initialize* e *updateResidualCapacity*, in modo tale che una volta raggiunta una capacità residua nulla, lo stato della radio venga settato tramite il metodo *setRadioMode* e quello del nodo tramite *setState*. Poiché il metodo *setState*, che si trova in *NodeStatus*, è di tipo *protected*, è stato creato un nuovo metodo *mySetState* di tipo public all’interno di un nuovo modulo *MyNodeStatus*, il quale estende *NodeStatus*.

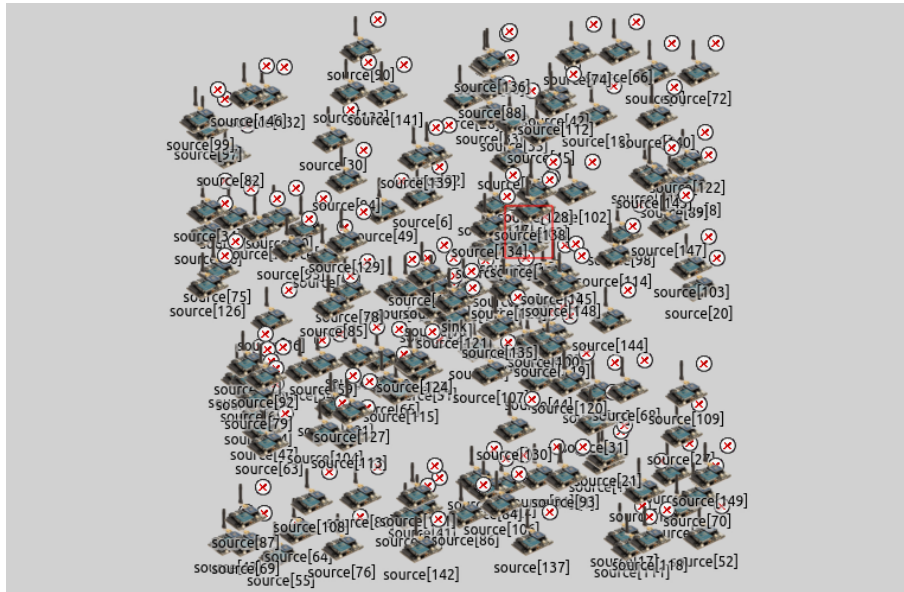


Figura 4: Configurazione 150 nodi: Scaricamento completo dei nodi

4 Statistiche

Le statistiche di interesse raccolte sono le seguenti:

- **rxTime**: somma dei tempi di ricezione; misura il tempo in cui il nodo sta ricevendo.
- **txTime**: somma dei tempi di trasmissione; misura il tempo in cui il nodo sta trasmettendo.
- **idleTime**: misura il tempo in cui il nodo è si trova in idle.
- **residualChargeCapacity**: rappresenta la capacità residua della batteria;
- **dischargeTime**: indica l'istante di tempo in cui la batteria si scarica.

5 Risultati

Poiché utilizzando la capacità nominale effettiva della batteria i tempi di simulazione (affinché tutti i nodi si scarichino) risultano essere estremamente lunghi (alcune decine di giorni), per semplicità è stata utilizzata una capacità nominale 1000 volte inferiore (da 7200 C a 7.2 C). Utilizzando questo valore di capacità, viene comunque accumulato un numero di statistiche sufficiente affinché l'errore standard rimanga basso.

Per creare le cinque configurazioni (10, 50, 100, 150, 200 nodi), è stata utilizzata la keyword *Config* nel file *omnetpp.ini*. Ad esempio, la prima configurazione, costituita da 10 nodi, è stata definita chiamandola “**tenNodes**” e impostando il numero di nodi trasmettitori tramite “****NumHost = 10**”.

5.1 Configurazione 10 nodi

Tempo di scaricamento dei nodi:

- Il tempo medio di scaricamento dei nodi è di 4604.43 s.
- Il primo nodo che si scarica è il nodo trasmettitore 9: il suo tempo di scaricamento è di 4048.05 s.
- L'ultimo nodo che si scarica è il nodo trasmettitore 3: il suo tempo di scaricamento è di 4983.55 s.

Tempo trascorso dai nodi negli stati di idle, trasmissione e ricezione:

	Max	Min	Media
idleTime	4493.75 s (nodo 3)	3607.81 s (nodo 9)	4130.70 s
txTime	89.45 s (nodo 9)	26.03 s (nodo 3)	47.96 s
rxTime	463.77 s (nodo 3)	350.79 s (nodo 9)	425.77 s

Al tempo di ricezione del nodo, rxTime, è stato accorpato il tempo di *busy*, che rappresenta il tempo in cui si è in ricezione, ma il segnale rilevato non ha abbastanza potenza per essere ricevuto.

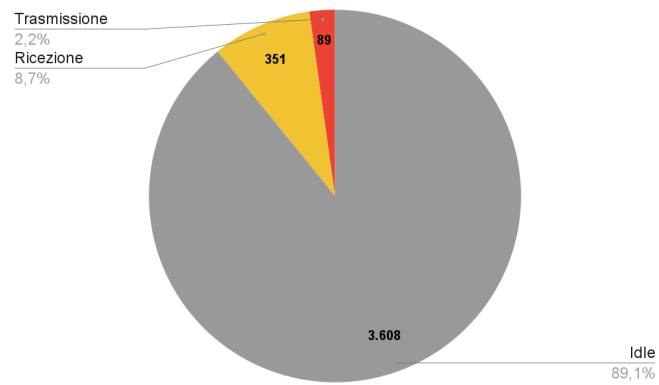


Figura 5: Nodo trasmettitore 9: Tempo trascorso negli stati di idle, rx e tx

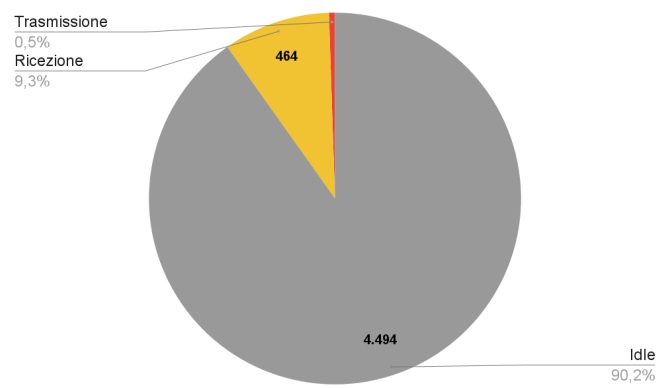


Figura 6: Nodo trasmettitore 3: Tempo trascorso negli stati di idle, rx e tx

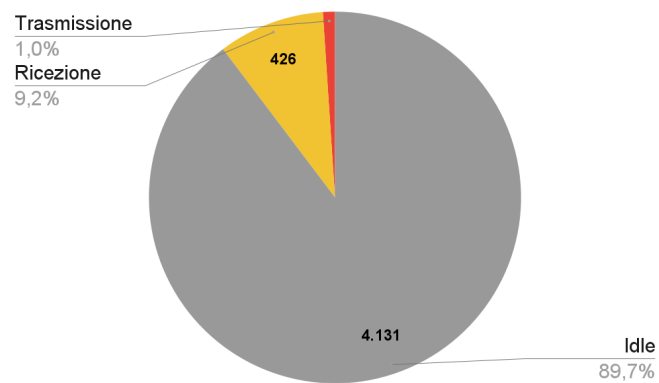


Figura 7: Configurazione 10 nodi: Tempo medio trascorso negli stati di idle, rx e tx

Dall'analisi dei dati precedenti, si evince che il nodo che si scarica per primo è quello che ha trasmesso di più; dai grafici a torta, infatti, si nota che nel nodo trasmettitore 9 la trasmissione rappresenta il 2.2% (89 secondi), mentre nel nodo trasmettitore 3 soltanto lo 0.5% (26 secondi).

Nella cartella *results* è possibile trovare tutte le statistiche estratte.

Stringa usata per filtrare le statistiche relative al tempo trascorso dai nodi in idle, rx e tx:

module(.mac) AND(name(txTime:sum) OR name(rxTime:sum) OR name(idleTime:last))*

5.2 Configurazione 50 nodi

Tempo di scaricamento dei nodi:

- Il tempo medio di scaricamento dei nodi è di 1230.67 s.
- Il primo nodo che si scarica è il nodo trasmettitore 30: il suo tempo di scaricamento è di 1197.47 s.
- L'ultimo nodo che si scarica è il nodo trasmettitore 37: il suo tempo di scaricamento è di 1295.62 s.

Tempo trascorso dai nodi negli stati di idle, trasmissione e ricezione:

	Max	Min	Media
idleTime	717.53 s (nodo 37)	638.02 s (nodo 30)	659.07 s
txTime	37.65 s (sink)	8.90 s (nodo 37)	16.39 s
rxTime	569.20 s (nodo 37)	520.66 s (sink)	555.20 s

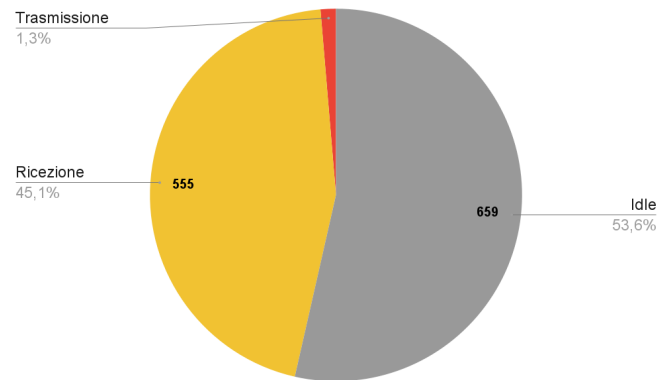


Figura 8: Configurazione 50 nodi: Tempo medio trascorso negli stati di idle, rx e tx

Si nota che all'aumentare del numero dei nodi non vi è più una netta preponderanza del tempo di idle. Infatti, i nodi, rispetto alla configurazione precedente, passano più tempo nello stato di ricezione. Questo avviene poiché si sta utilizzando una rete di tipo wireless, in cui ogni nodo riceve tutti i pacchetti, a prescindere se sono destinati a lui o meno; pertanto, aumentando il numero di nodi, aumenterà il tempo che ognuno dei nodi passa in ricezione. Tuttavia, in questa configurazione il tempo di idle è ancora prevalente (53.6%), seguito dal tempo di ricezione (45.1%) e dal tempo di trasmissione (1.3%). Inoltre, si nota che il sink è il nodo con il tempo di ricezione minore: questo avviene in quanto il sink è l'unico nodo che non riceve gli ack (inviati dal sink stesso). È importante evidenziare che, all'aumentare dei numero

dei nodi, aumenta la probabilità che si verifichino delle collisioni, anche perchè si sta utilizzando un protocollo mac di tipo Pure ALOHA, in cui ogni nodo trasmette non appena ha necessità di trasmettere, senza ascoltare il canale.

5.3 Configurazione 100 nodi

- Il tempo medio di scaricamento dei nodi è di 816.23 s.
- Il primo nodo che si scarica è il nodo trasmettitore 49: il suo tempo di scaricamento è di 804.24 s.
- L'ultimo nodo che si scarica è il nodo trasmettitore 29: il suo tempo di scaricamento è di 845.03 s.

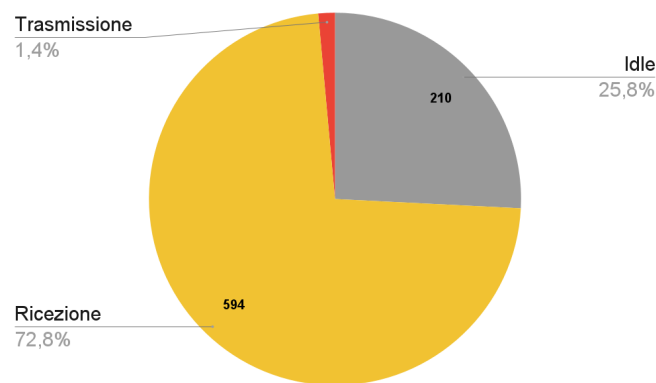


Figura 9: Configurazione 100 nodi: Tempo medio trascorso negli stati di idle, rx e tx

5.4 Configurazione 150 nodi

- Il tempo medio di scaricamento dei nodi è di 683,99 s.
- Il primo nodo che si scarica è il nodo trasmettitore 29: il suo tempo di scaricamento è di 675.24 s.
- L'ultimo nodo che si scarica è il nodo trasmettitore 25: il suo tempo di scaricamento è di 712.15 s.

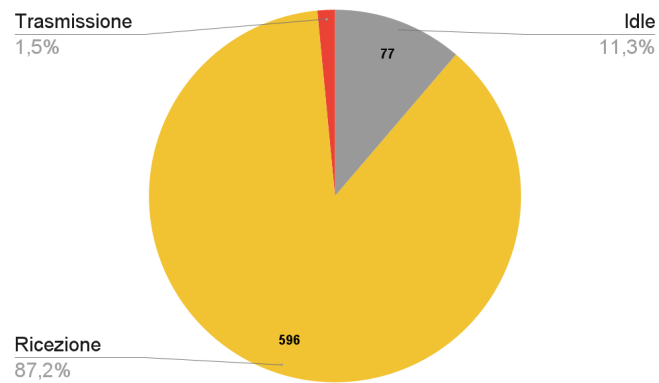


Figura 10: Configurazione 150 nodi: Tempo medio trascorso negli stati di idle, rx e tx

5.5 Configurazione 200 nodi

- Il tempo medio di scaricamento dei nodi è di 630.53 s.
- Il primo nodo che si scarica è il nodo trasmettitore 121: il suo tempo di scaricamento è di 623.77 s.
- L'ultimo nodo che si scarica è il nodo trasmettitore 170: il suo tempo di scaricamento è di 666.94 s.

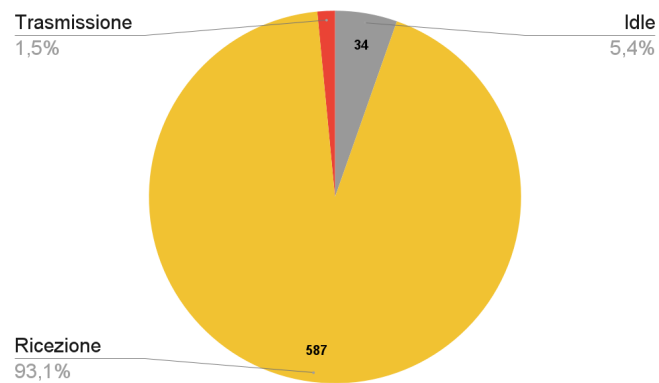


Figura 11: Configurazione 200 nodi: Tempo medio trascorso negli stati di idle, rx e tx

6 Conclusioni

Analizzando i risultati delle cinque simulazioni, si nota che il tempo di scaricamento medio dei nodi diminuisce all'aumentare del numero dei nodi.

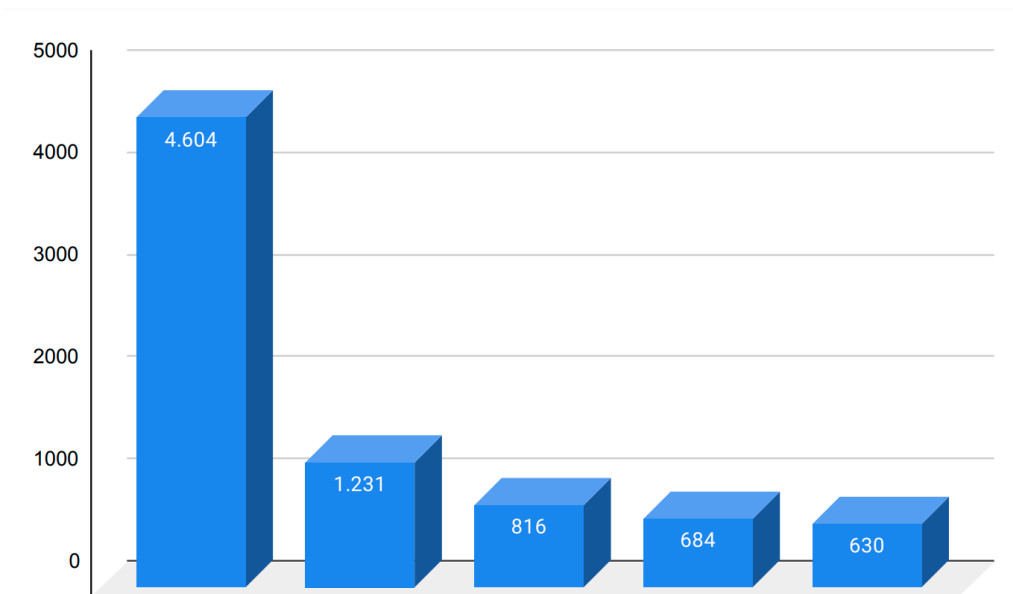


Figura 12: Tempi di scaricamento medi delle 5 simulazioni a confronto

Come si può notare dalla figura 12, nella configurazione a 10 nodi il tempo di scaricamento medio dei nodi è di 4604 s. Man mano che i nodi aumentano, il tempo di scaricamento medio diminuisce, fino ad arrivare ad un valore di 630 s nella configurazione da 200 nodi.

Questo comportamento è dovuto al fatto che, all'aumentare del numero dei nodi, il tempo medio in cui un nodo sta in idle diminuisce, mentre aumentano il tempo di ricezione (notevolmente) e il tempo di trasmissione.

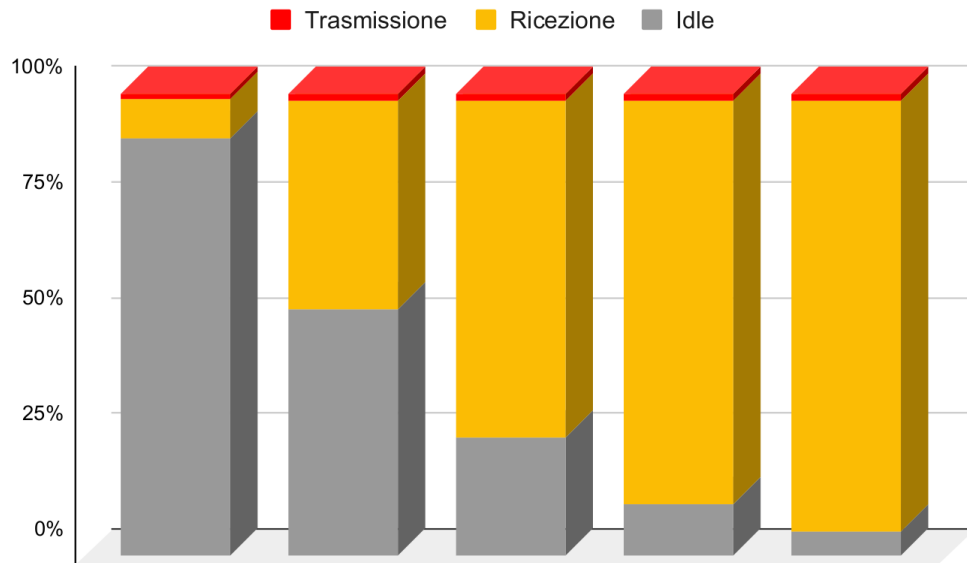


Figura 13: Tempo medio di idle, rx e tx delle 5 simulazioni a confronto

Dalla figura 13 si evince che il tempo medio trascorso da un nodo nello stato di idle nella prima configurazione (rete con 10 nodi) occupa la maggior parte del tempo di simulazione (89.7%), mentre il tempo medio trascorso negli stati di ricezione e trasmissione sono decisamente più bassi (rispettivamente 9.2% e 1%). Già nella seconda configurazione il tempo trascorso nello stato di idle si riduce drasticamente, fino ad occupare il 5.4% del tempo di simulazione nella configurazione a 200 nodi. Al contrario, il tempo di ricezione aumenta notevolmente, fino ad occupare, nella configurazione a 200 nodi, quasi per intero il tempo di simulazione (93.1%). Infine, il tempo di trasmissione aumenta leggermente all'aumentare del numero di nodi, passando da un valore dell'1% (rete con 10 nodi) ad un valore dell'1.5% (rete con 200 nodi).

L'aumento del tempo medio di ricezione è dovuto al fatto che con un numero di nodi maggiore ogni nodo riceverà un numero maggiore di pacchetti, poiché vi sono più nodi trasmettenti. Poiché per un nodo che si trova nello stato di trasmissione o ricezione la batteria ha un consumo molto maggiore rispetto a quando il nodo si trova nello stato di idle, l'aumento del tempo di ricezione e del tempo di trasmissione (a discapito del tempo di idle) all'aumentare del numero dei nodi comporta che la durata della batteria sarà inferiore. Infatti, quando un nodo si trova nello stato di idle consuma circa 1-2 mA, mentre quando si trova nello stato di ricezione e trasmissione consuma rispettivamente 5-10 mA e 100 mA.

Elenco delle figure

1	Configurazione rete con 10 nodi	1
2	Protocollo Aloha: in grigio le frame in collisione	2
3	Batteria utilizzata	4
4	Configurazione 150 nodi: Scaricamento completo dei nodi	5
5	Nodo trasmettitore 9: Tempo trascorso negli stati di idle, rx e tx . .	8
6	Nodo trasmettitore 3: Tempo trascorso negli stati di idle, rx e tx . .	8
7	Configurazione 10 nodi: Tempo medio trascorso negli stati di idle, rx e tx	8
8	Configurazione 50 nodi: Tempo medio trascorso negli stati di idle, rx e tx	9
9	Configurazione 100 nodi: Tempo medio trascorso negli stati di idle, rx e tx	10
10	Configurazione 150 nodi: Tempo medio trascorso negli stati di idle, rx e tx	11
11	Configurazione 200 nodi: Tempo medio trascorso negli stati di idle, rx e tx	11
12	Tempi di scaricamento medi delle 5 simulazioni a confronto	12
13	Tempo medio di idle, rx e tx delle 5 simulazioni a confronto	13