DHT Symphony Simulation

Andrea Aquino, Michele Consiglio Università di Bologna

11 febbraio 2013

Il protocollo DHT Symphony discusso da Manku, Bawa e Raghavan si propone come nuova frontiera per il mantenimento e l'accesso di dizionari distribuiti sulla rete globale. Sfruttando una topologia di rete virtuale ad anello sulla quale vengono instaurate connessioni tra nodi distanti, è possibile ridurre drasticamente i tempi di recupero di dati evitandone l'eccessiva replicazione. Tuttavia il numero di messaggi necessari al funzionamento del protocollo stesso non è oggetto di studio del documento prodotto dagli autori di cui sopra. Si intende quindi analizzare tale aspetto mediante la simulazione del protocollo a mezzo del simulatore Omnet++. Verrà inoltre introdotta una modifica protocollare con lo scopo di diminuire il numero di messaggi transitanti nella rete senza inficiare la correttezza e l'efficacia del protocollo.

1 Protocollo DHT Symphony

Al fine di realizzare una tabella hash distribuita affidabile è necessario che si mantengano copie dei dati in essa contenuti su svariate macchine facenti parte della rete che compone la tabella stessa. In pratica ciò è quasi sempre inattuabile. Il protocollo DHT Symphony tenta di risolvere il problema eleggendo dei candidati facenti parte della rete per la gestione e la memorizzazione di determinate

porzioni della tabella. Affinché l'accesso ai dati in essa contenuti sia efficiente i nodi vengono organizzati in una struttura circolare in modo che ogni nodo abbia un vicino ad esso immediatamente precedente ed uno successivo. Questo approccio garantisce la raggiungibilità di qualsiasi macchina della DHT a partire da qualunque altra in un numero di passi non maggiore della metà del numero di nodi facenti parte della rete. Al fine di ridurre ulteriormente questo valore il protocollo introduce delle connessioni ulteriori tra nodi non adiacenti instaurate a mezzo di una particolare distribuzione di probabilità (distribuzione armonica da cui il protocollo trae il suo nome). Per poter comprendere al meglio il suo funzionamento è consigliabile pensare alla totalità dei dati contenuti nella DHT come se fossero posti su una circonferenza di perimetro unitario. Ogni nodo facente parte della rete occupa una posizione differente su tale circonferenza e si occupa di soddisfare le richieste (in gergo, *fa da manager*) per la porzione di dati contenuta tra il nodo medesimo ed il suo immediato vicino in senso antiorario. Si noti che:

1. Sebbene un nodo sia connesso ai due nodi a lui adiacenti e a svariati altri non conosce il punto preciso che questi occupano sulla circonferenza. Tale informazione è di vitale importanza per il funzionamento di diversi punti del protocollo;

2. Inoltre essendo la rete particolarmente dinamica, il protocollo gestisce l'accesso e l'uscita di nodi in modo che i dati della DHT rimangano consistenti. In particolare quando un nodo intende accedere alla rete decide autonomamente in quale punto della circonferenza posizionarsi mediante la scelta randomica di un valore uniformemente distribuito nell'intervallo [0,1). Informa poi un nodo già connesso della sua volontà di accedere alla rete distribuita; mediante un protocollo di instradamento interno noto come *protocollo di routing* tale nodo contatta l'attuale manager di quel punto permettendo al nuovo nodo di accedere previa modifica delle connessioni opportune.

Il protocollo consta di tutta una serie di sottoprotocolli più o meno complessi che gli permettono di operare nella sua interezza. Per informazioni più dettagliate si rimanda il lettore al documento "DHT Symphony: Distributed hashing in a small world" [1].

2 Simulazione

2.1 Gli strumenti

Lo studio di simulazione è stato realizzato a mezzo del simulatore Omnet++. Nello specifico il protocollo oggetto dello studio è stato emulato mediante l'implementazione dei moduli stanti alla lista seguente:

- File NED descrivente la topologia di rete virtuale;
- Classe rappresentante i singoli nodi all'interno della rete;
- Pacchetti di rete strutturati per la trasmissione delle informazioni protocollari rilevanti;
- File di configurazione per gestire le informazioni parametriche della simulazione.

Sono stati inoltre utilizzati gli strumenti di analisi offline messi a disposizione da tale simulatore per l'analisi prestazionale del protocollo nella versione originale e soggetto a modifica. I risultati di tale studio verranno discussi nella sezione "Risultati" del presente elaborato.

2.2 Il livello di dettaglio

L'implementazione in esame consiste a tutti gli effetti in una emulazione puntuale del protocollo DHT Symphony allo scopo di analizzare con assoluta precisione le prestazioni dello stesso in termini di numero di messaggi scambiati tra i nodi della rete al passare del tempo. La simulazione di rete astrae dalla topologia di rete reale realizzando soltanto le connessioni virtuali previste dal protocollo medesimo. In particolare i nodi sono disposti su di un ipotetico cerchio di perimetro unitario che costituisce una rete ad anello. Nella fase di inizializzazione del protocollo ogni nodo è quindi connesso ai suoi due immediati vicini ed è dotato di un numero K (configurabile) di connessioni "lunghe" con i nodi i cui id differiscono dal proprio tra le 8 e le 8+K unità.

Per semplicità le connessioni con i nodi vicini (da qui in poi connessioni brevi) sono state realizzate mediante gate bidirezionali facenti parte di un vettore di gate contenente la totalità delle connessioni per il nodo corrente. In fase di sperimentazione il numero di nodi della rete è stato fissato a 16, 32 e 64, di cui la metà connessi nella struttura ad anello discussa in precedenza e la rimanente metà in principio non connessa.

Infine sono stati individuati e simulati fedelmente i seguenti aspetti del protocollo:

1. stima della lunghezza del segmento gestito da un nodo;
2. stima del numero di nodi della rete;
3. protocollo di routing per individuare il manager di un punto qualsiasi sull'anello di perimetro unitario;
4. protocollo di relink;
5. protocollo di join;

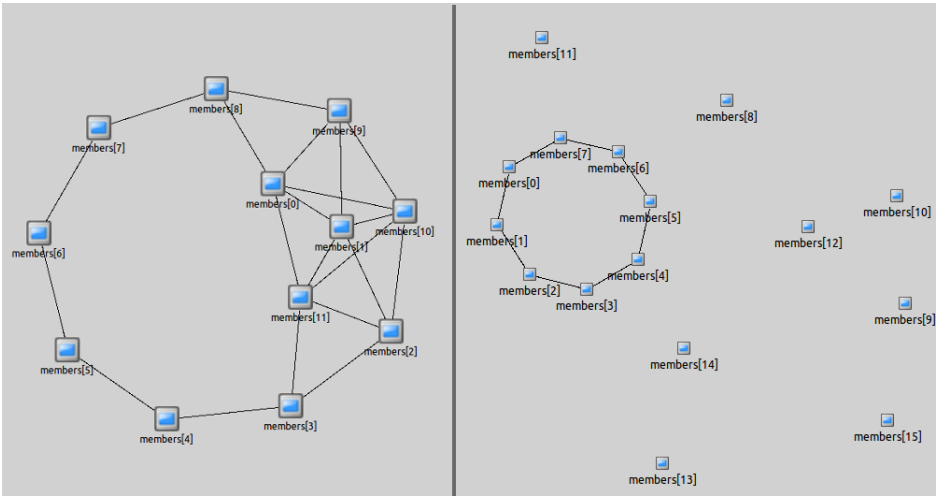

Figura 1: *Topologie di reti virtuali*

2.3 Assunzioni

Il documento "Symphony: Distributed Hashing In A Small World" non specifica una variegata serie di dettagli. Pertanto è stato necessario effettuare tutta una serie di assunzioni al fine di poter implementare il protocollo stesso.

1. La rete è supposta essere, in principio, popolata;
2. Il numero di nodi della DHT è inferiormente limitato da un valore positivo configurabile, supposto maggiore di 8;
3. Ogni nodo non ha accesso diretto al valore indicante la posizione all'interno della circonferenza di perimetro unitario dei propri vicini, deve pertanto recuperarla chiedendola esplicitamente a questi mediante l'uso di messaggi di rete. Il protocollo Symphony assume che la posizione dei nodi manager

del segmento immediatamente precedente e del segmento immediatamente successivo di un dato nodo sia direttamente accessibile in ogni momento. Ciò ha perfettamente senso in termini matematici in quanto ogni nodo altro non è che una particolare struttura dotata di collegamenti ai nodi vicini. Tuttavia i nodi descritti dal protocollo sono a tutti gli effetti macchine connesse attraverso la rete, ne deriva il non trascurabile dettaglio implementativo che consiste nel dover recuperare tali valori a mezzo messaggi mediati dai canali di comunicazione dai nodi adiacenti.

4. Il lookup dei file della DHT non è ritenuto di interesse e pertanto non è simulato;

5. Nessuna coppia di nodi facente parte della DHT ha la stessa posizione sulla circonferenza di perimetro unitario. In linea teorica pescando valori randomici uniformemente distribuiti nell'intervallo [0,1) due nodi diversi potrebbero ottenere lo stesso valore. Ciò significherebbe che entrambi i nodi sono manager della stessa porzione della DHT. Il protocollo Symphony non dichiara come affrontare questo particolare problema, che crea ambiguità in fase di esecuzione del sottoprotocollo di join.

3 Modifica protocollare

L'implementazione originale del protocollo di routing prevede, in virtù del punto 3. della sezione "Assunzioni", la necessità di effettuare un broadcast ai nodi vicini per poter definire quali di questi dista meno dal nodo destinazione e quindi instradare le successive richieste attraverso di esso. In fase di simulazione suddetto dettaglio sembra inficiare duramente la qualità del protocollo in termini prestazionali (quantità di messaggi trasmessi). Quindi, sotto l'assunzione che ogni nodo mantenga traccia della posizione dei suoi vicini sulla circonferenza di perimetro unitario, è stata effettuata una modifica protocollare che snellisce sostanzialmente il protocollo di routing (massivamente sfruttato anche da parte dei protocolli di join e di relink) sotto tale punto di vista.

3.1 Implementazione

In teoria sarebbe stato necessario tenere traccia per ogni nodo della posizione di ogni altro nodo ad esso connesso mediante un vettore locale. Si noti che la posizione dei vicini di un dato nodo può variare solamente a seguito dell'accesso o dell'uscita di un nodo dalla rete; si ritiene che tale evento occorra assai più raramente rispetto al protocollo di routing (che è cuore del meccanismo di

inserimento e ricerca di chiavi nel dizionario distribuito). Per semplicità, tale vettore è stato assunto esistere e la posizione dei nodi vicini al nodo corrente è stata recuperata mediante le direttive del simulatore Omnet++ stesso.

3.2 Correttezza

La modifica protocollare precedentemente discussa non modifica in alcun modo la semantica del protocollo. Segue una dimostrazione informale di tale evidenza:

1. la modifica in esame riguarda esclusivamente il protocollo di routing; se la semantica di tale protocollo rimane la medesima, di conseguenza la semantica del protocollo complessivo non varia.

2. per dimostrare che la semantica del protocollo di routing non cambia è sufficiente dimostrare che in ogni caso in cui la versione originale del protocollo effettua una scelta tra i nodi vicini per l'instradamento di una richiesta, il protocollo modificato effettua la medesima scelta.

3. dimostriamo il punto 2. per induzione sul numero di connessioni lunghe del nodo corrente.
   - **caso base**: il nodo corrente ha solo le connessioni brevi con i nodi che lo seguono immediatamente in senso orario e antiorario. Poiché sono supposti esistere almeno 3 nodi all'interno della rete simulata, deve esistere una relazione d'ordine totale tra le posizioni di questi sulla circonferenza di perimetro unitario. Siano

     $$0 \leq a \neq b \neq c < 1$$

     tali posizioni e sia

     $$0 \leq x < 1$$

     il valore di cui si intende individuare il nodo manager per mezzo del protocollo di routing. Ne consegue che

     $$|a - x| \neq |b - x| \neq |c - x|$$

     e quindi una sola distanza è quella minima. Non essendoci alcuna forma di non determinismo ed essendo la stima di a, b e c uguale per entrambe le versioni del protocollo la scelta di instradamento non subisce variazioni.

– **caso induttivo**: supponiamo che i protocolli si comportino allo stesso modo per un certo numero n di link lunghi. Dimostriamo che continuano a comportarsi allo stesso modo anche per n+1 link lunghi. In virtù del punto 5. della sezione "Assunzioni" il nodo connesso al nodo corrente mediante l'n+1-esimo link lungo ha una posizione diversa da tutti gli altri sulla circonferenza di perimetro unitario. I casi sono dunque due:

  * il nuovo nodo è il più vicino al nodo destinazione; ma in tal caso lo è per entrambe le versioni del protocollo e quindi i protocolli continuano ad instradare i messaggi mediante lo stesso nodo.
  * il nuovo nodo è più distante rispetto ad un altro vicino del nodo corrente dal nodo destinazione; ma in tal caso il protocollo si comporta come nel caso in cui l'n+1-esimo link lungo non esista affatto, ma in questa situazione i due protocolli hanno lo stesso comportamento per ipotesi induttiva.

Fino ad ora abbiamo supposto che la posizione dei nodi vicini al nodo che sta correntemente eseguendo il protocollo di routing sia costantemente aggiornata per la versione modificata del protocollo. Ciò è tuttavia certamente vero sotto l'ipotesi che nel momento in cui un nodo accede alla rete o immediatamente prima di abbandonarla comunichi ai propri vicini la necessità di aggiornare le posizioni dei nodi adiacenti.

4 Risultati

A mezzo del simulatore general purpose Omnet++ è stata effettuata la simulazione del protocollo Symphony sia nella versione originale che nella versione modificata. Per entrambi i protocolli è stato fissato il numero di nodi della DHT a 128, 256 e 512, di cui in principio 64 connessi tra loro nella canonica rete ad anello. Per ognuna di queste configurazioni sono state analizzate differenti politiche relative ai tempi di attesa per l'accesso dei nodi alla rete. Per ogni configurazione sono state eseguite quindici run di simulazione, ed in ogni run i generatori di numeri casuali dell'applicazione sviluppata sono stati inizializzati con dei seed diversi. Utilizzando gli strumenti di analisi offline messi a disposizione da Omnet++ sono state effettuate le seguenti analisi:

1. *scalabilità del protocollo*, utilizzando come parametro prestazionale il rapporto tra il numero di messaggi inviati da ciascun nodo ed il numero totale di messaggi inviati all'interno della rete da tutti i nodi;

2. *confronto tra la versione originale e la versione modificata del protocollo Symphony*, utilizzando come parametro prestazionale il numero di messaggi inviati dai singoli nodi.

4.1 Scalabilità del protocollo

Di seguito verranno esposti i risultati dell'analisi di scalabilità del protocollo Symphony nella sua versione originale, concentrando i confronti in prima battuta sull'eventuale variazione del parametro prestazionale al variare dell'access rate dei nodi all'interno della DHT e successivamente sull'impatto della variazione del numero di nodi che entrano nella DHT. Nelle figure che seguono sono rappresentati i grafici delle percentuali di messaggi inviati da ciascun nodo. In particolare l'asse delle ascisse rappresenta il tempo simulato, mentre quello delle ordinate la percentuale di messaggi inviati dal nodo in questione. Per semplicità ogni studio prestazionale del presente elaborato prende in esame due sole possibili politiche di access rate:

- access rate alto: attesa tra l'accesso alla rete di un nodo e quello successivo uniformemente distribuita in un intervallo rappresentante pochi secondi.
- access rate basso: attesa tra l'accesso alla rete di un nodo e quello successivo uniformemente distribuita in un intervallo rappresentante svariati minuti.

In figura 2 e 3 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 0 con access rate alto e con access rate basso all'interno di una DHT con massimo 128 nodi. Come si evince, a meno di due run di simulazione in cui la percentuale di messaggi inviati da tale nodo è più alta (quasi doppia) con access rate basso, la percentuale di messaggi inviati si attesta circa sull'1% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

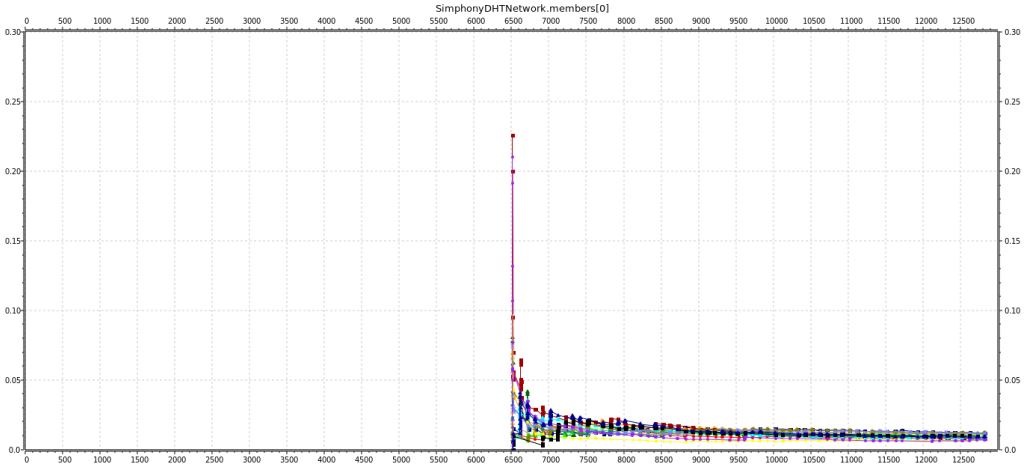

Figura 2: *Percentuale di messaggi inviati dal nodo 0 con access rate alto all'interno di una DHT con massimo 128 nodi*

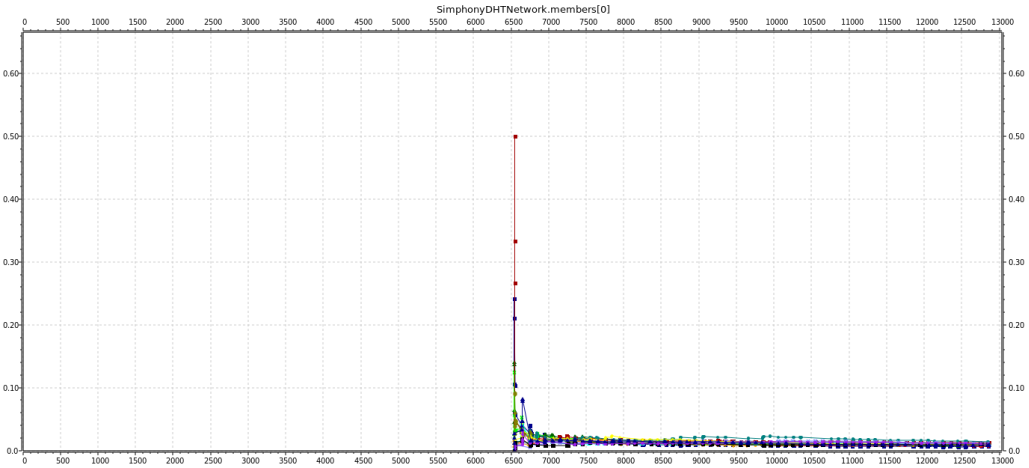

Figura 3: *Percentuale di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 128 nodi*

In figura 4 e 5 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 64 con access rate alto e con access rate basso all'interno di una DHT con massimo 128 nodi. Come si evince, a meno di una fase iniziale in cui la percentuale di messaggi inviati da tale nodo è più alta (circa quattro volte di più) con access rate basso, la percentuale di messaggi inviati si

attesta circa sul 2% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati neanche da tale nodo.

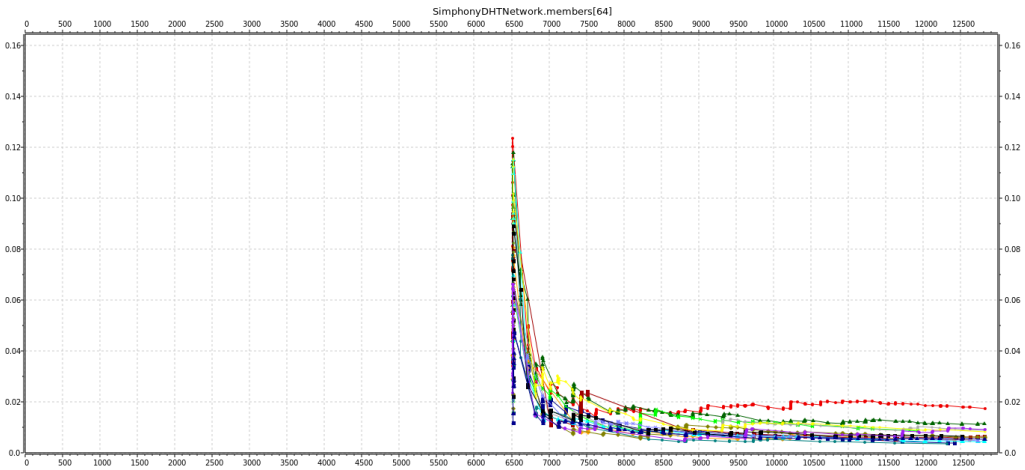

Figura 4: *Percentuale di messaggi inviati dal nodo 64 con access rate alto all'interno di una DHT con massimo 128 nodi*

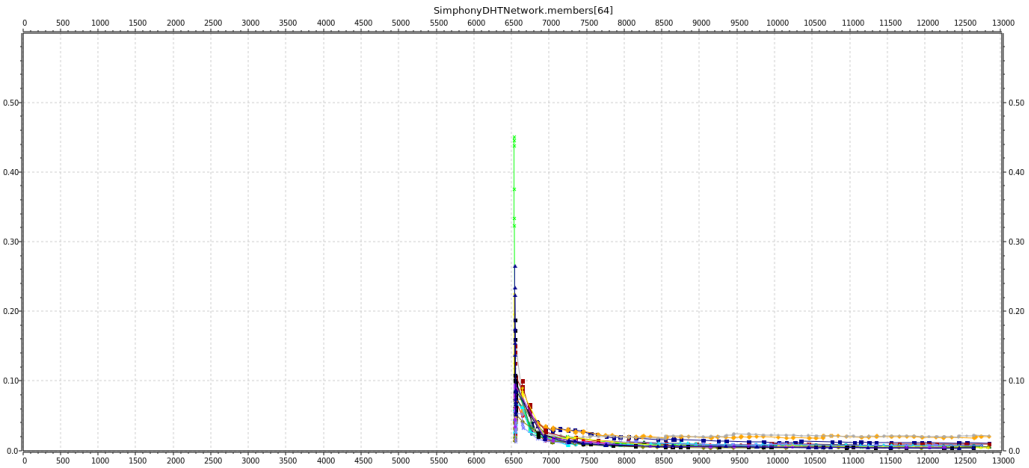

Figura 5: *Percentuale di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 128 nodi*

In figura 6 e 7 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 96 con access rate alto e con access rate basso all'interno di una DHT con massimo 128 nodi. Come si evince, a meno di

due run di simulazione in cui con access rate alto il nodo in questione ha una percentuale di messaggi inviati più alta (alla fine circa doppia) rispetto a tutte le altre run nelle stesse condizioni, la percentuale di messaggi inviati si attesta sotto lo 0,6% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati dal nodo in questione.

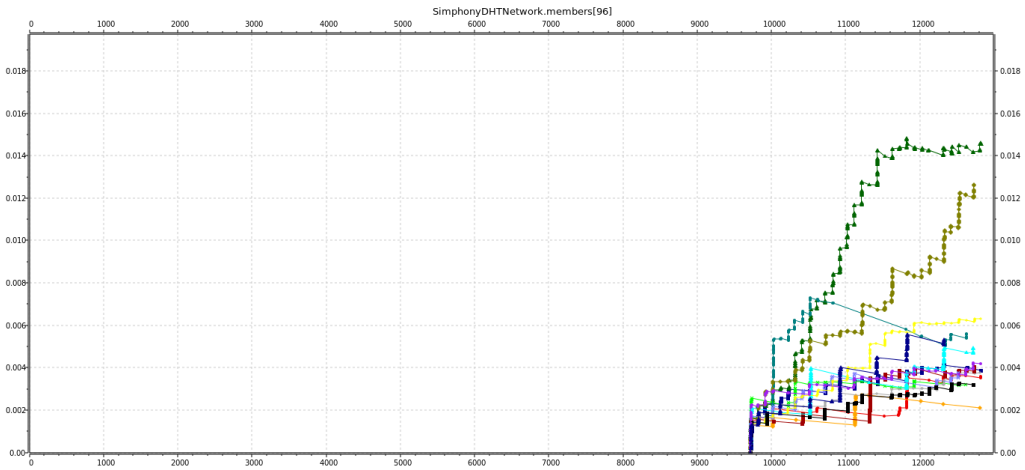

Figura 6: *Percentuale di messaggi inviati dal nodo 96 con access rate alto all'interno di una DHT con massimo 128 nodi*

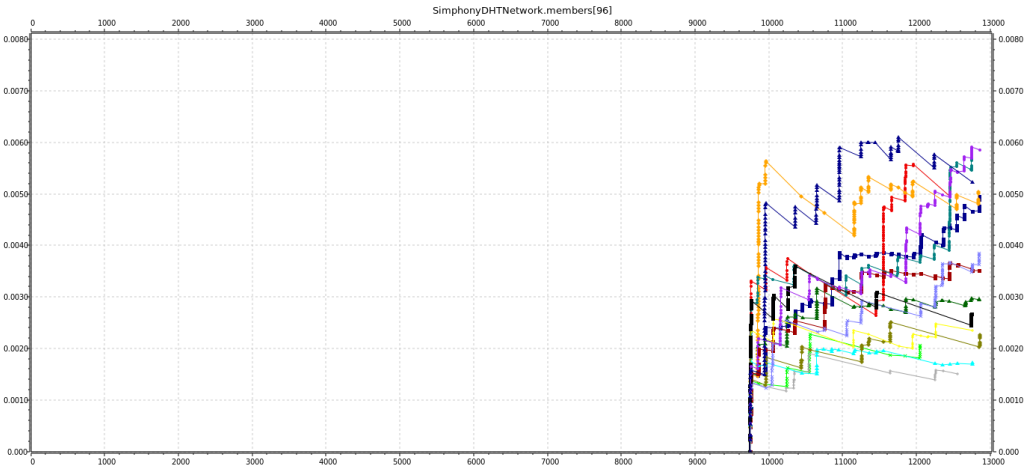

Figura 7: *Percentuale di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 128 nodi*

In figura 8 e 9 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 0 con access rate alto e con access rate basso all'interno di una DHT con massimo 256 nodi. Come si evince, a meno di una run di simulazione in cui la percentuale di messaggi inviati da tale nodo è più alta (quasi doppia) con access rate alto, la percentuale di messaggi inviati si attesta circa sull'1% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

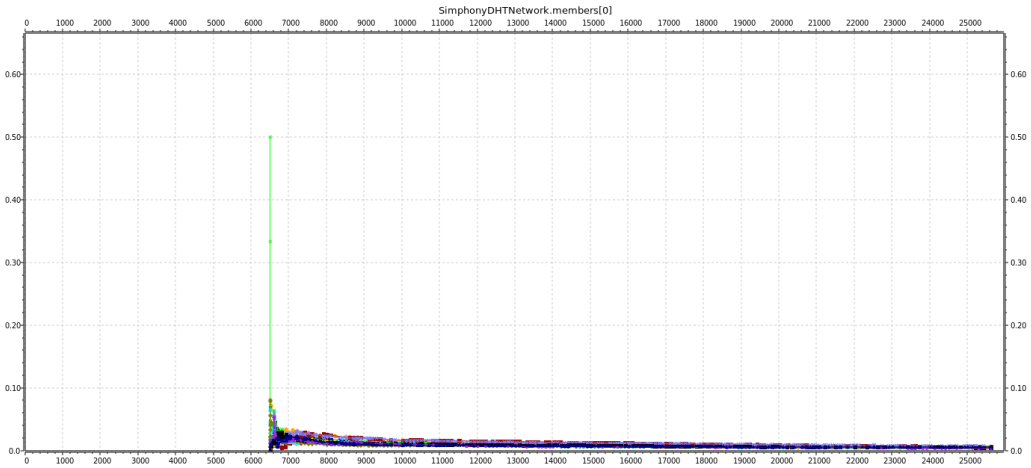

Figura 8: *Percentuale di messaggi inviati dal nodo 0 con access rate alto all'interno di una DHT con massimo 256 nodi*

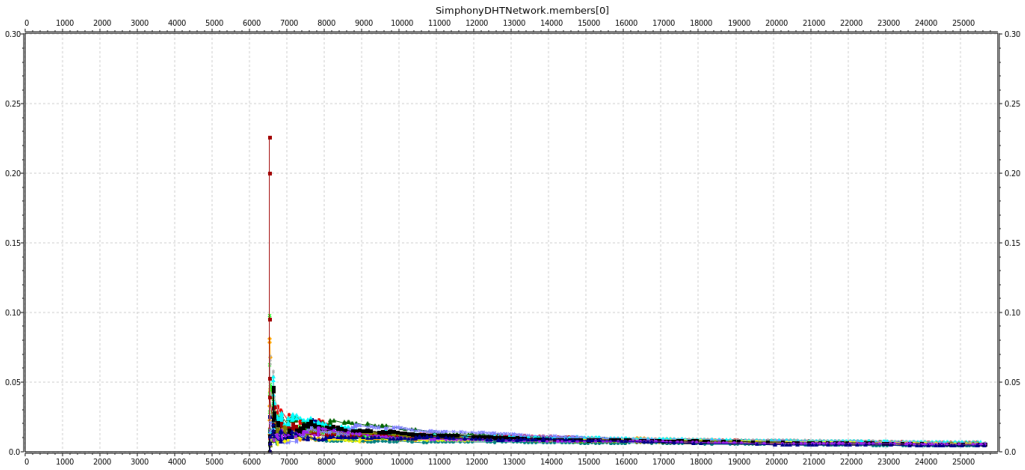

Figura 9: *Percentuale di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 256 nodi*

In figura 10 e 11 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 64 con access rate alto e con access rate basso all'interno di una DHT con massimo 256 nodi. Come si evince, a meno di una fase iniziale in cui la percentuale di messaggi inviati da tale nodo è più alta (quasi il doppio) con access rate basso, la percentuale di messaggi inviati si attesta sotto il 2% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

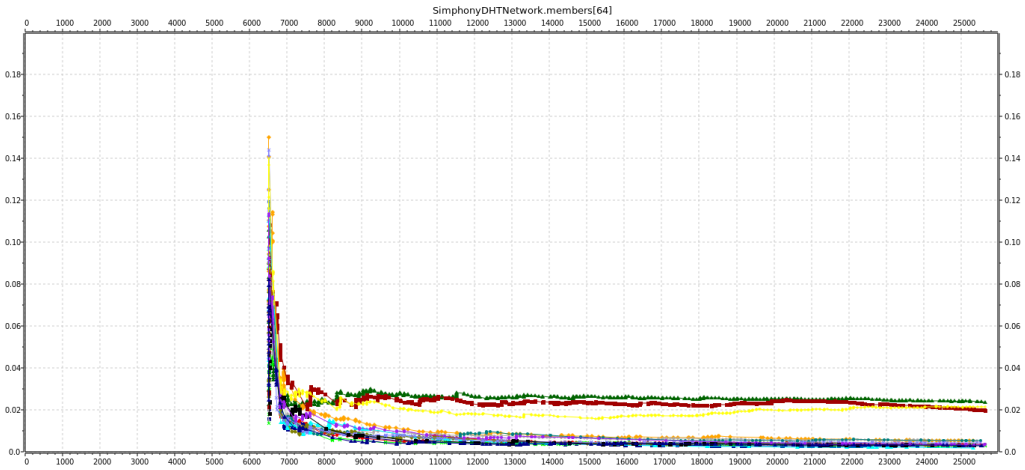

Figura 10: *Percentuale di messaggi inviati dal nodo 64 con access rate alto all'interno di una DHT con massimo 256 nodi*

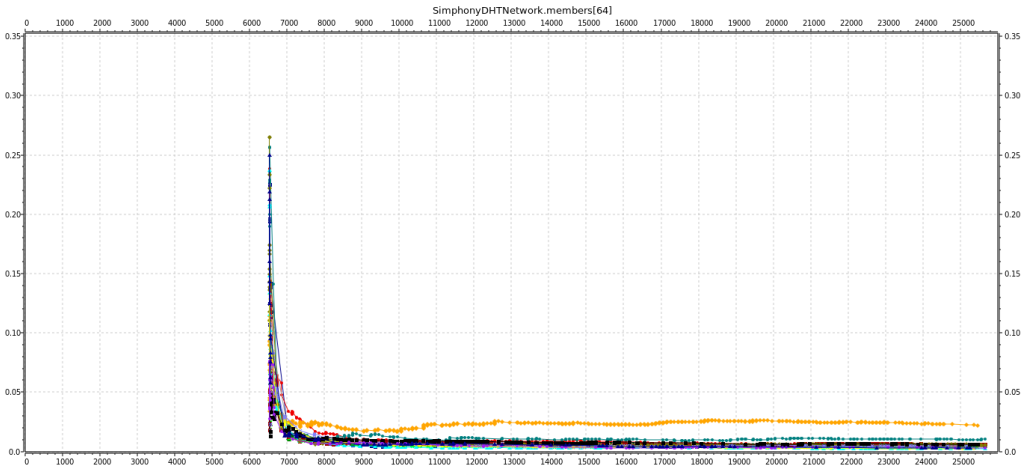

Figura 11: *Percentuale di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 256 nodi*

In figura 12 e 13 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 96 con access rate alto e con access rate basso all'interno di una DHT con massimo 256 nodi. Come si evince, a meno di due run di simulazione in cui con access rate alto il nodo in questione ha una percentuale di messaggi inviati più alta (alla fine circa doppia) rispetto a tutte le altre run nelle stesse condizioni, la percentuale di messaggi inviati si attesta sotto lo 0,9% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

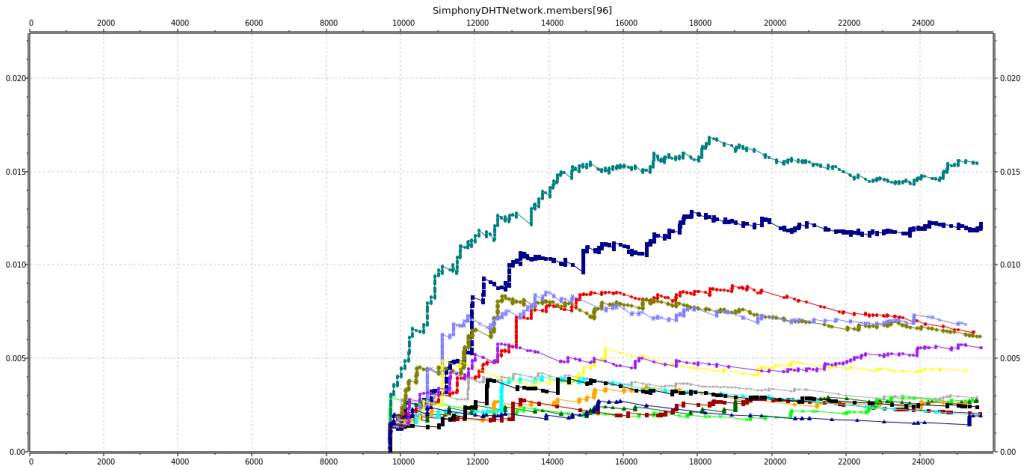

Figura 12: *Percentuale di messaggi inviati dal nodo 96 con access rate alto all'interno di una DHT con massimo 256 nodi*

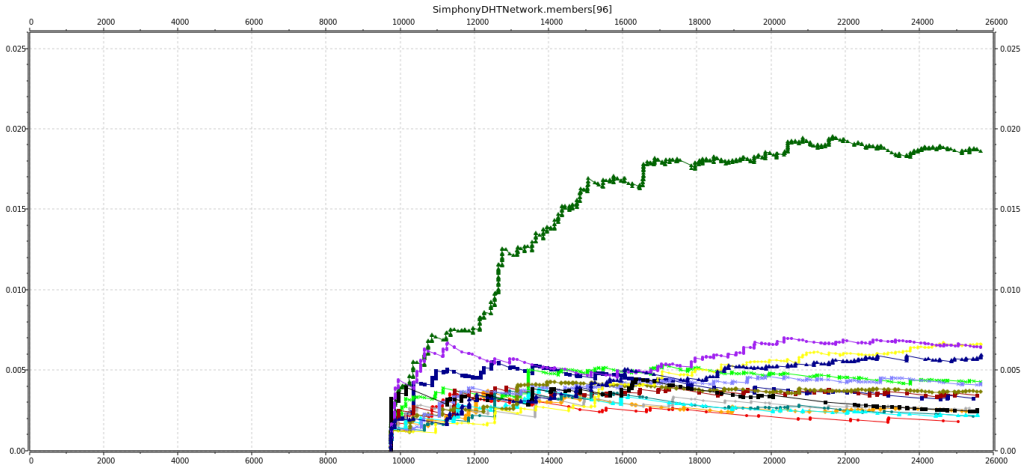

Figura 13: *Percentuale di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 256 nodi*

In figura 14 e 15 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 0 con access rate alto e con access rate basso all'interno di una DHT con massimo 512 nodi. Come si evince, a meno di una fase iniziale in cui la percentuale di messaggi inviati da tale nodo è più alta (circa doppia) con access rate basso, la percentuale di messaggi inviati si attesta circa sull'1%

in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

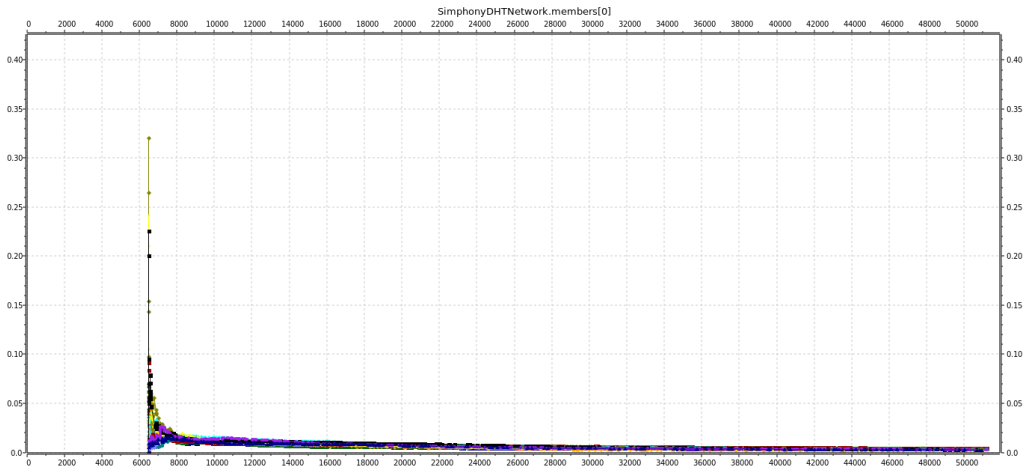

Figura 14: *Percentuale di messaggi inviati dal nodo 0 con access rate alto all'interno di una DHT con massimo 512 nodi*

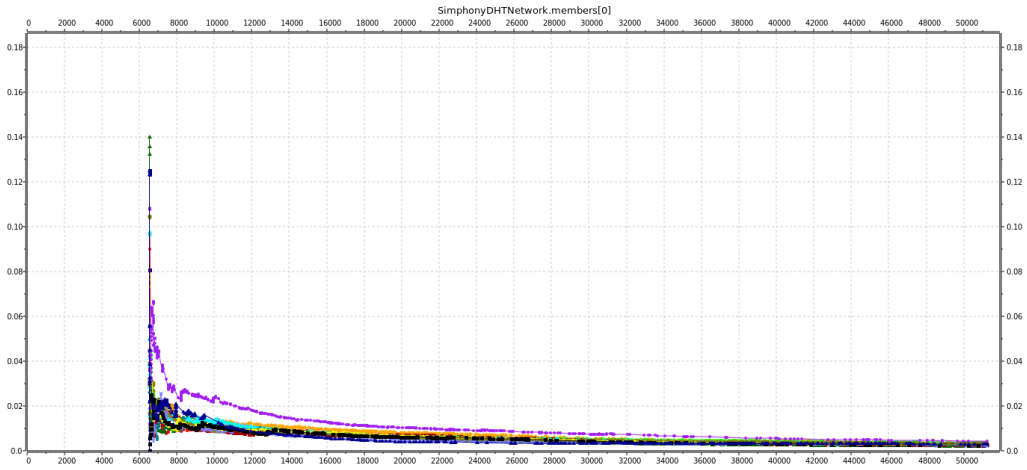

Figura 15: *Percentuale di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 512 nodi*

In figura 16 e 17 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 64 con access rate alto e con access rate basso all'interno di una DHT con massimo 512 nodi. Come si evince, a meno di una

run di simulazione in cui la percentuale di messaggi inviati da tale nodo è più alta (circa il doppio) con access rate basso, la percentuale di messaggi inviati si attesta sotto il 2% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

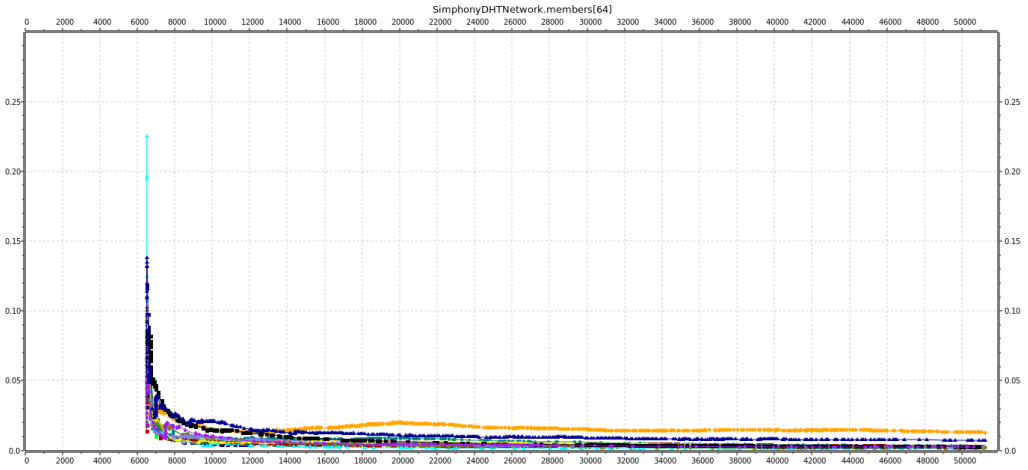

Figura 16: *Percentuale di messaggi inviati dal nodo 64 con access rate alto all'interno di una DHT con massimo 512 nodi*

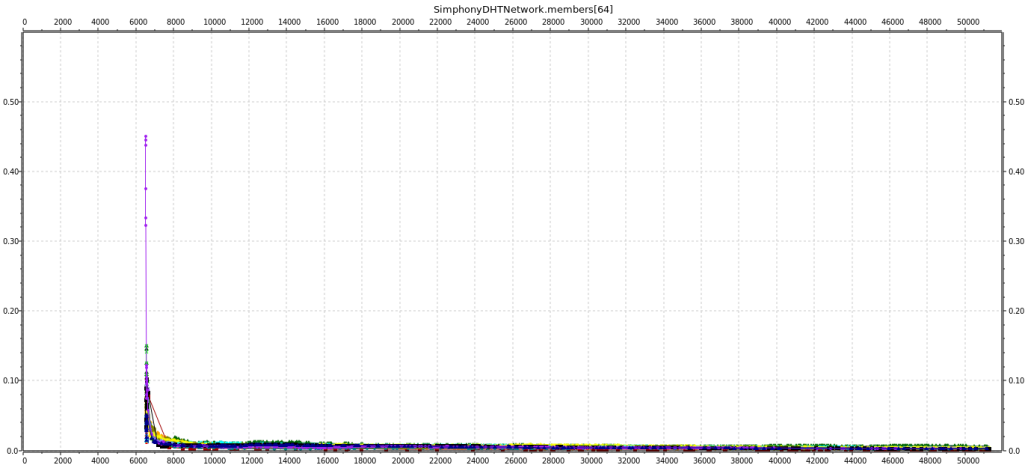

Figura 17: *Percentuale di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 512 nodi*

In figura 18 e 19 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 96 con access rate alto e con access rate basso

all'interno di una DHT con massimo 512 nodi. Come si evince, a meno di due run di simulazione in cui con access rate alto il nodo in questione ha una percentuale di messaggi inviati più alta (alla fine circa doppia) rispetto a tutte le altre run nelle stesse condizioni, la percentuale di messaggi inviati si attesta sotto lo 0,7% in entrambi i casi, segno che la variazione dell'access rate non influisce sul carico di messaggi inviati da tale nodo.

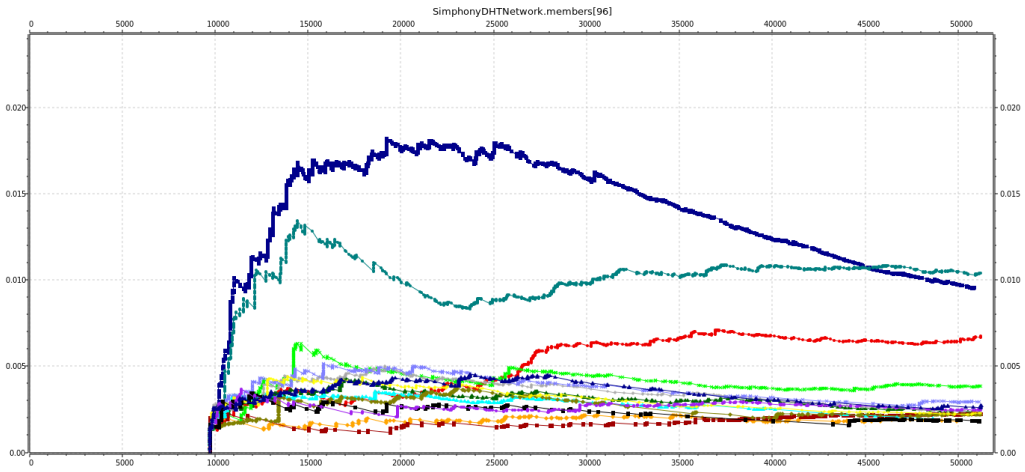

Figura 18: *Percentuale di messaggi inviati dal nodo 96 con access rate alto all'interno di una DHT con massimo 512 nodi*

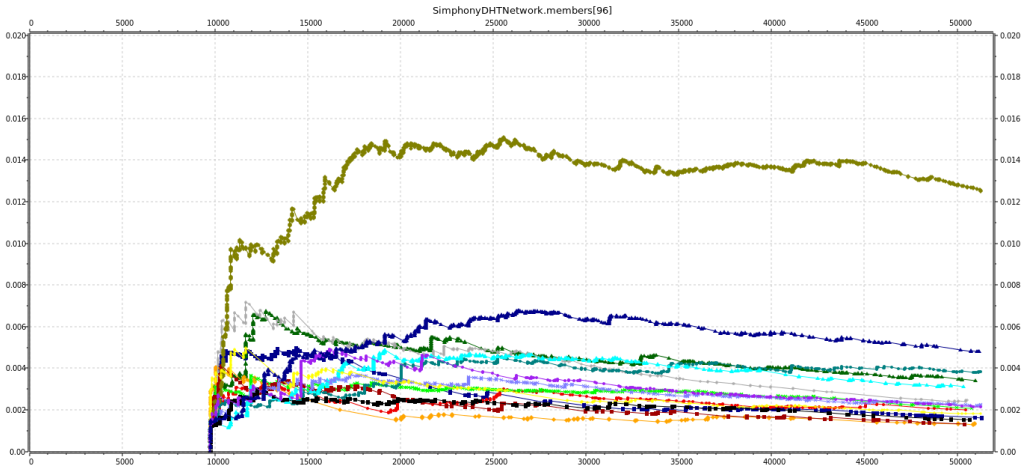

Figura 19: *Percentuale di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 512 nodi*

A fronte di tale analisi è decisamente verosimile concludere che le politiche di access rate non influiscono in alcun modo sulla percentuale di messaggi inviati da ogni nodo se si assume fissato il numero di nodi presenti nella rete.

In figura 20 e 21 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 0 con con access rate basso all'interno di una DHT con massimo 128 nodi e la percentuale di messaggi inviati dal nodo 0 con access rate alto all'interno di una DHT con massimo 512 nodi. Come si evince, il carico di messaggi inviati dal nodo 0 nel secondo caso è sempre inferiore (circa dimezzato) rispetto al primo caso, segno che il protocollo scala molto bene, distribuendo equamente il carico di richieste tra tutti i nodi coinvolti.

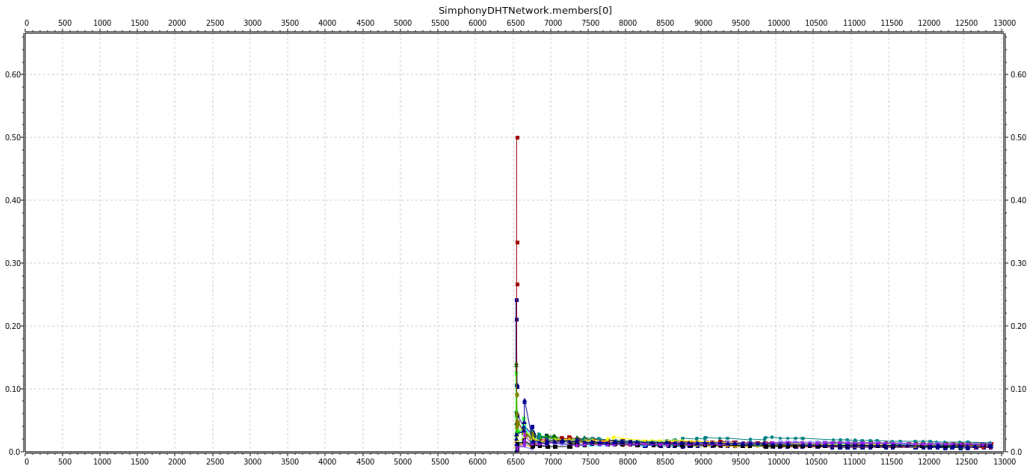

Figura 20: *Percentuale di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 128 nodi*

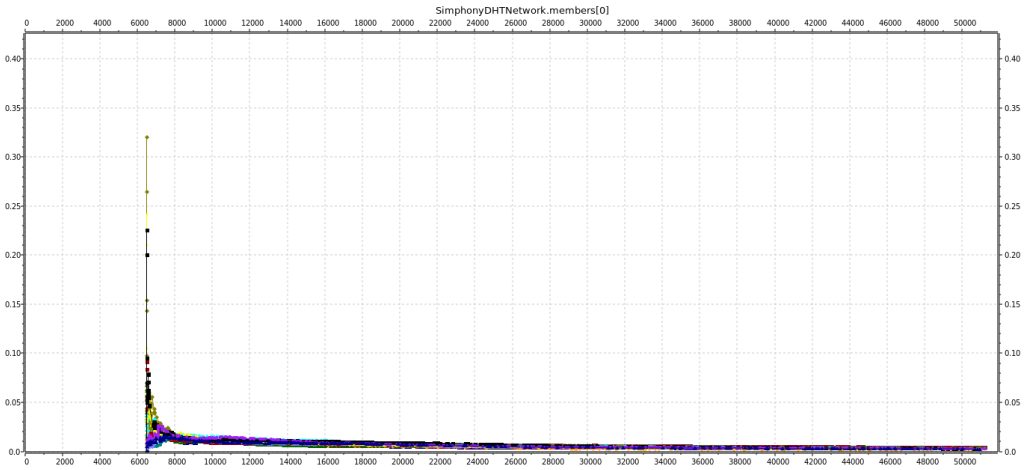

Figura 21: *Percentuale di messaggi inviati dal nodo 0 con access rate alto all'interno di una DHT con massimo 512 nodi*

In figura 22 e 23 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 64 con con access rate basso all'interno di una DHT con massimo 128 nodi e la percentuale di messaggi inviati dal nodo 64 con access rate alto all'interno di una DHT con massimo 512 nodi. Come si evince, il carico di messaggi inviati dal nodo 64 nel secondo caso è sempre inferiore (circa dimezzato) rispetto al primo caso, segno che il protocollo scala molto bene, distribuendo equamente il carico di richieste tra tutti i nodi coinvolti.

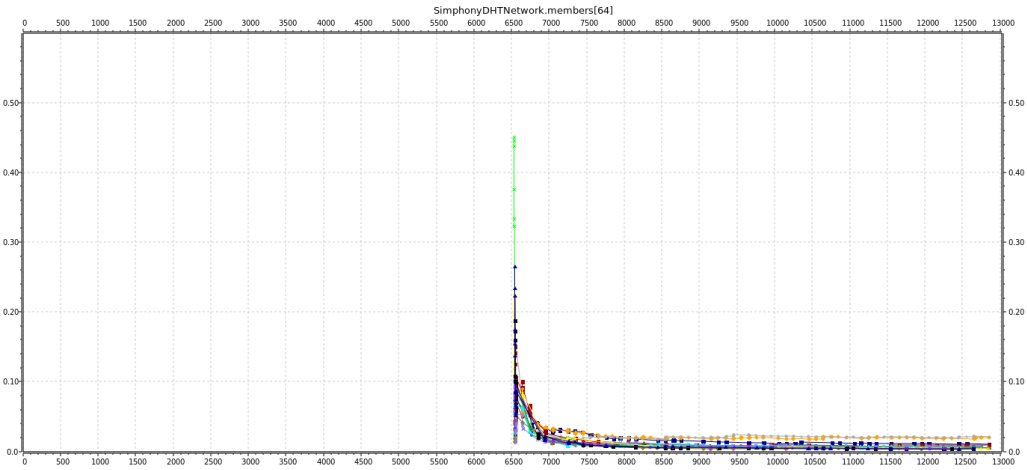

Figura 22: *Percentuale di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 128 nodi*

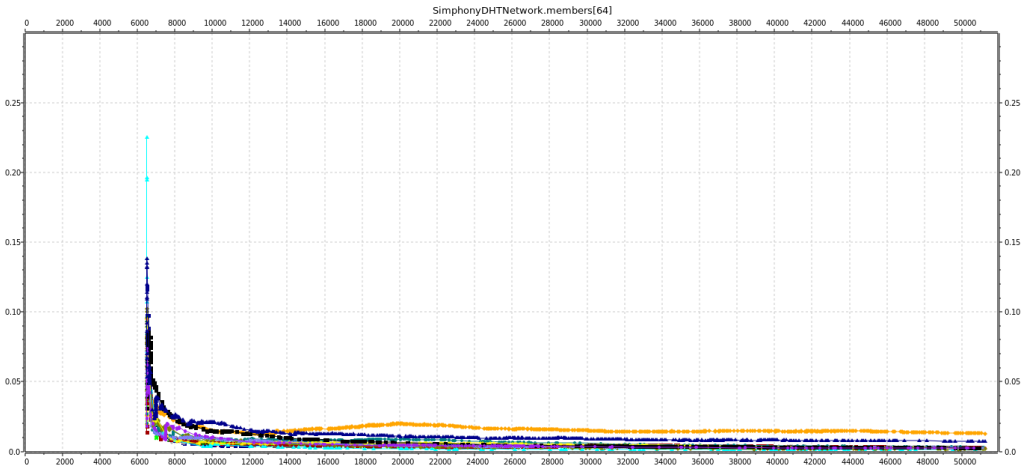

Figura 23: *Percentuale di messaggi inviati dal nodo 64 con access rate alto all'interno di una DHT con massimo 512 nodi*

In figura 24 e 25 vengono rappresentati rispettivamente la percentuale di messaggi inviati dal nodo 96 con con access rate basso all'interno di una DHT con massimo 128 nodi e la percentuale di messaggi inviati dal nodo 96 con access rate alto all'interno di una DHT con massimo 512 nodi. Come si evince, il carico di messaggi inviati dal nodo 96 nel secondo caso è di molto inferiore rispetto al primo caso, segno che il protocollo scala molto bene, distribuendo equamente il carico di richieste tra tutti i nodi coinvolti.

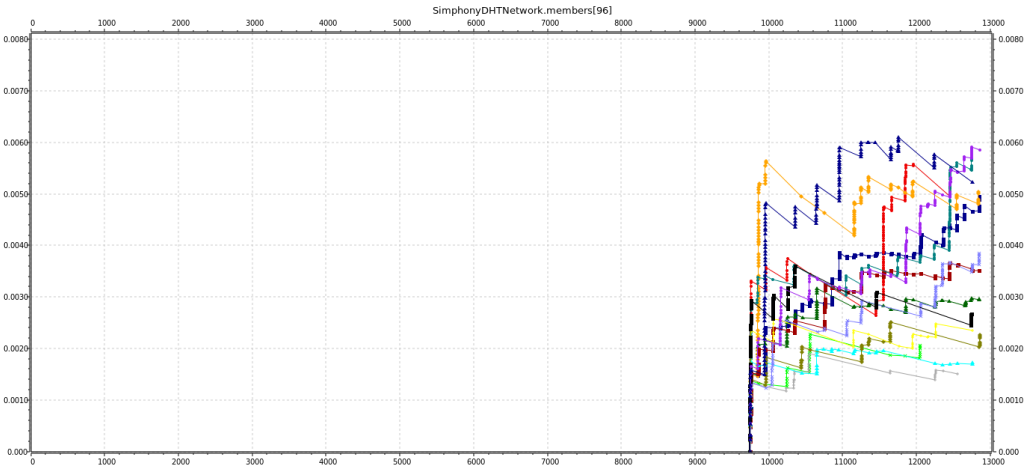

Figura 24: *Percentuale di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 128 nodi*

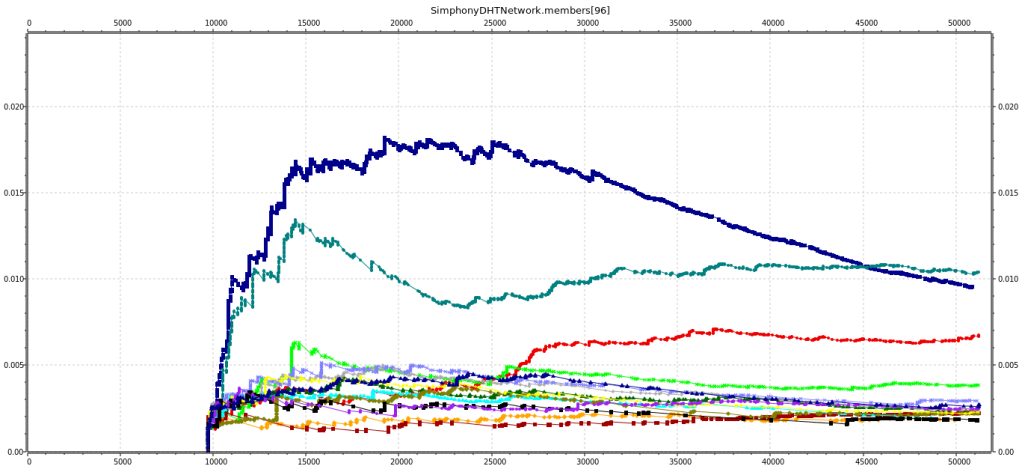

Figura 25: *Percentuale di messaggi inviati dal nodo 96 con access rate alto all'interno di una DHT con massimo 512 nodi*

Possiamo quindi concludere, alla luce dei risultati finora esposti, che il protocollo DHT Symphony scala bene sia quando il numero dei nodi facenti parte della rete aumenta sensibilmente sia quanto la velocità con cui questi accedono alla rete aumenta di molto.

4.2 Confronto tra il protocollo originale ed il protocollo modificato

Di seguito verranno esposti i risultati del confronto tra il protocollo DHT Symhpony nella versione originale e nella versione modificata precedentemente discussa. Il parametro prestazionale utilizzato per tale confronto è il numero di messaggi inviati da ciascun nodo all'interno della rete virtuale. I due protocolli hanno comportamenti estremamenti simili quindi qualsiasi misura normalizzata non sarebbe indicativa di alcuna differenza evidente. Nelle figure che seguono l'asse delle ascisse rappresenta il tempo simulato, mentre quello delle ordinate il numero di messaggi inviati da un determinato nodo.

In figura 26 e 27 vengono rappresentati il numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 128 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, in tutte le run il nodo 0 invia sempre un numero inferiore di messaggi se si utilizza il protocollo suddetto nella sua versione modificata.

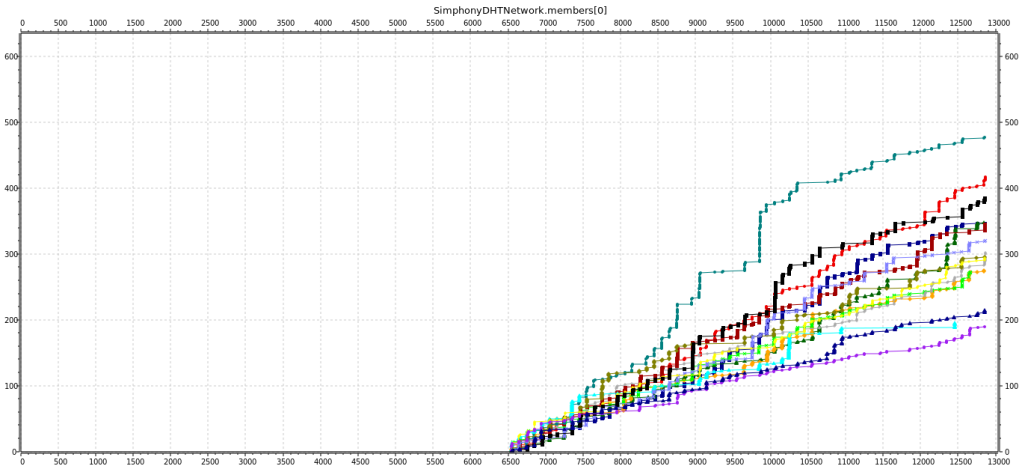

Figura 26: *Numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 128 nodi utilizzando il protocollo DHT Symphony*

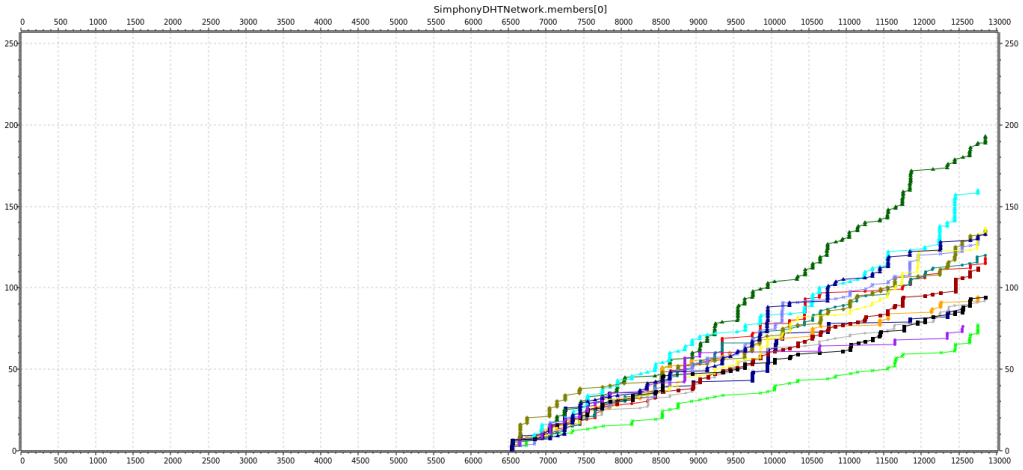

Figura 27: *Numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 128 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 28 e 29 vengono rappresentati il numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 128 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, il numero di messaggi inviati dal nodo 64 si aggira sempre intorno ai 100 (ad eccezione di alcune run) e comunque sotto

ai 250 se si utilizza la versione modificata del protocollo, mentre utilizzando la versione originale avremmo un numero massimo di messaggi inviati di circa 750 con quasi tutte le run di simulazione che si attestano su un numero di messaggi tra i 150 e i 250. Possiamo quindi affermare che, anche in questo caso, la versione modificata del protocollo ha prestazioni migliori.

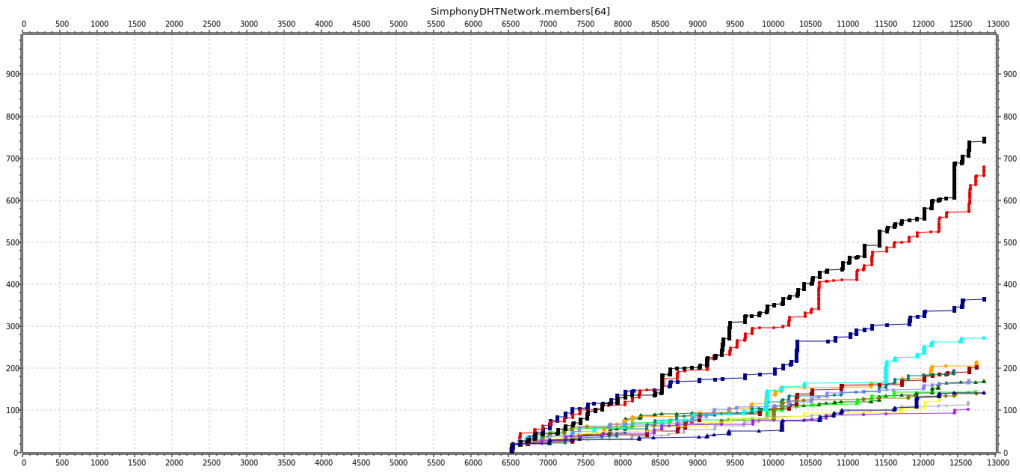

Figura 28: *Numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 128 nodi utilizzando il protocollo DHT Symphony*

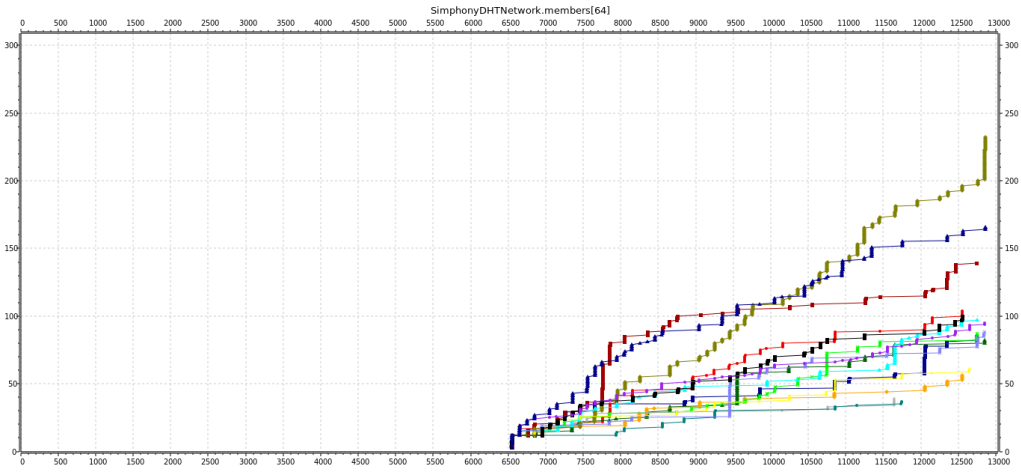

Figura 29: *Numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 128 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 30 e 31 vengono rappresentati il numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 128 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, il numero dei messaggi inviati da tale nodo nella versione modificata del protocollo si aggira (ad eccezione di due run) tra i 20 e i 60, mentre utilizzando il protocollo originale vi sono solo due run di simulazione in cui il numero di messaggi inviati è tra i 50 e i 60. Anche in questo caso quindi, la versione modificata del protocollo risulta essere migliore.

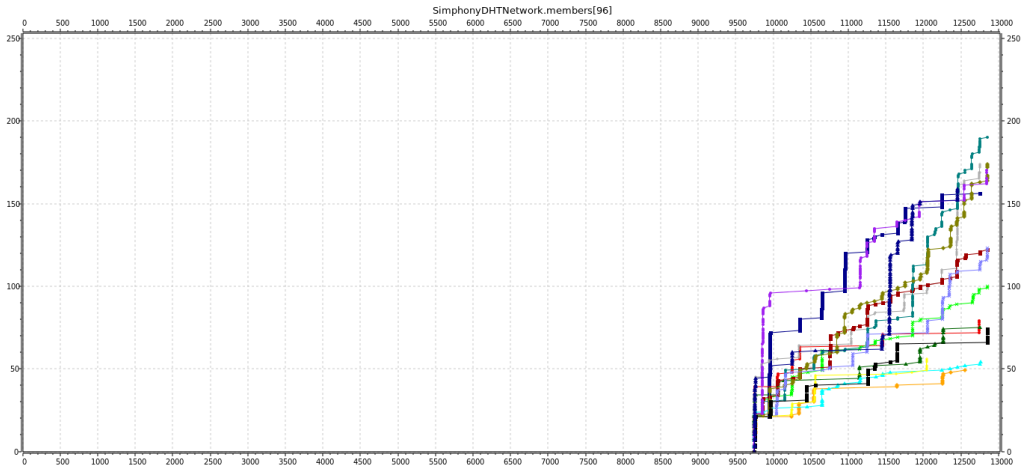

Figura 30: *Numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 128 nodi utilizzando il protocollo DHT Symphony*

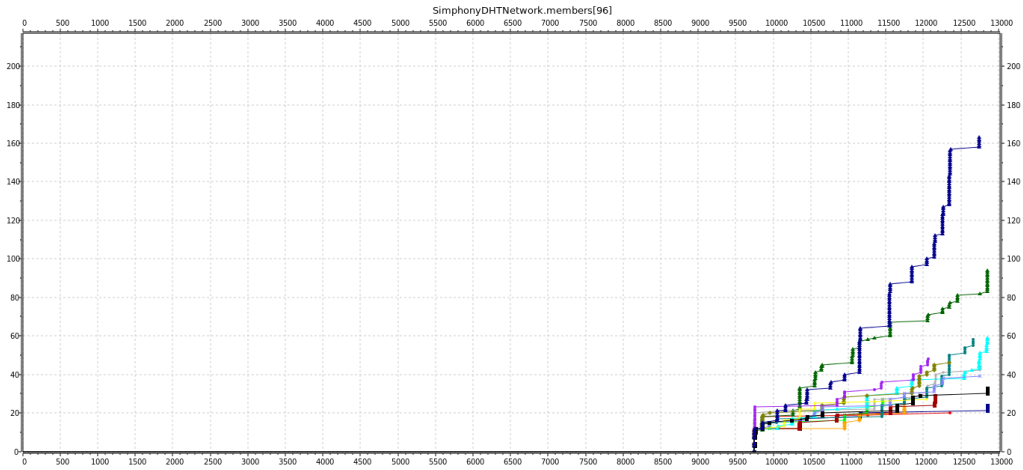

Figura 31: *Numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 128 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 32 e 33 vengono rappresentati il numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 256 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, in tutte le run il nodo 0 invia sempre un numero inferiore di messaggi se si utilizza il protocollo suddetto nella sua versione modificata. Possiamo quindi affermare che anche in questo caso la versione modificata del protocollo risulta essere migliore.

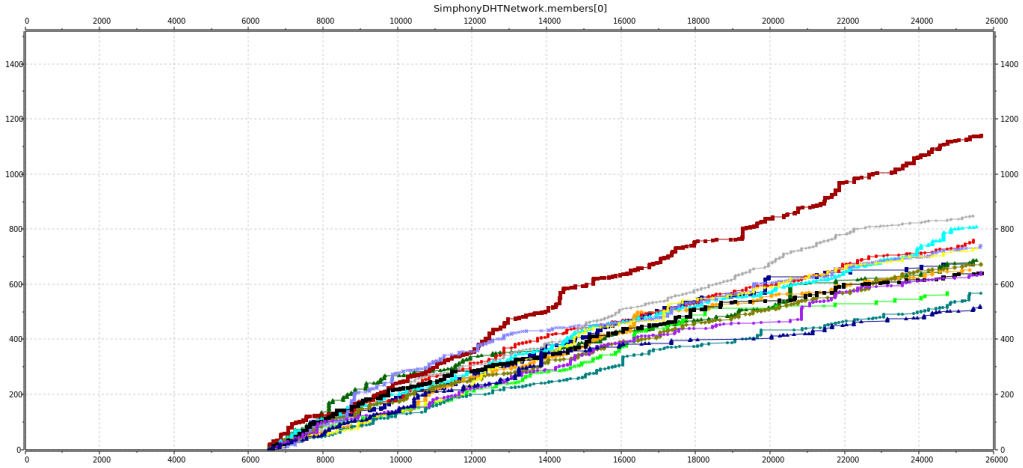

Figura 32: *Numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 256 nodi utilizzando il protocollo DHT Symphony*

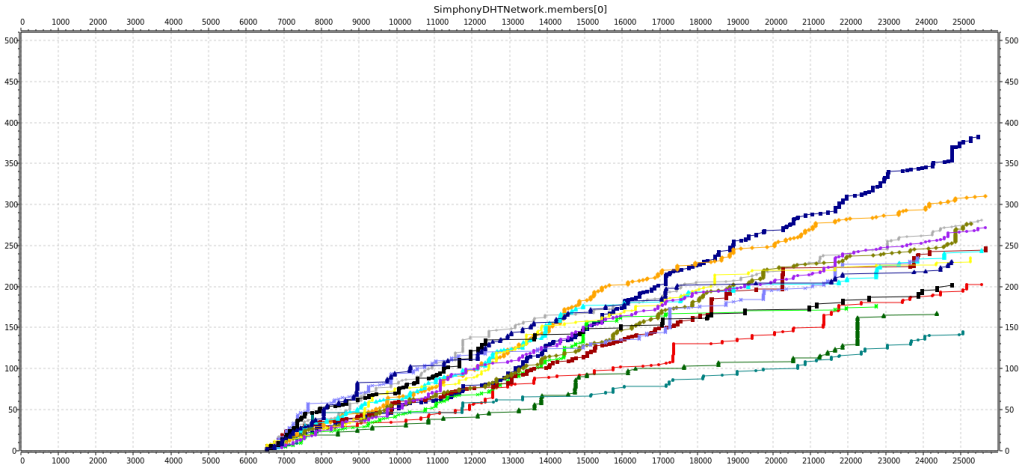

Figura 33: *Numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 256 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 34 e 35 vengono rappresentati il numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 256 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, nella versione modificata del protocollo il nodo 64 invia, nella maggior parte delle run di simulazione, un numero di

messaggi inferiore a 400, mentre nella versione originale invia un numero di messaggi superiore a 400. Anche in questo caso quindi la versione modificata risulta essere migliore.

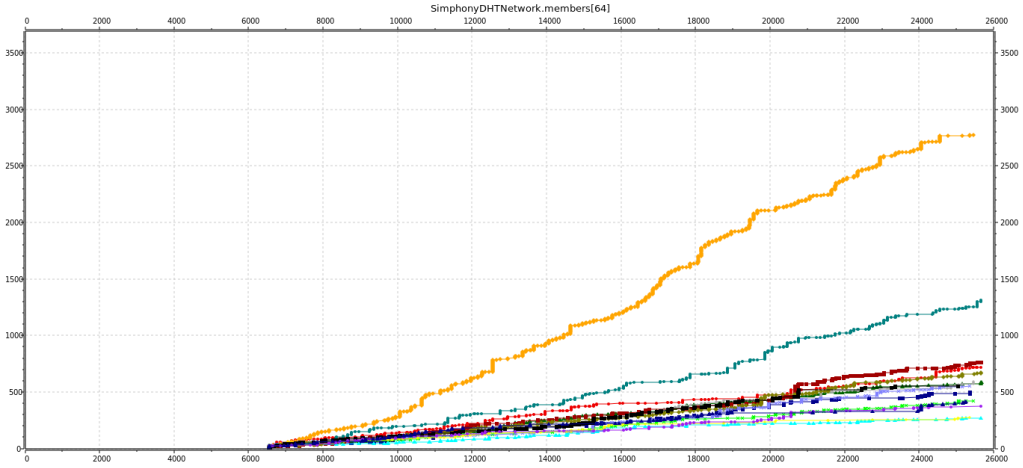

Figura 34: *Numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 256 nodi utilizzando il protocollo DHT Symphony*

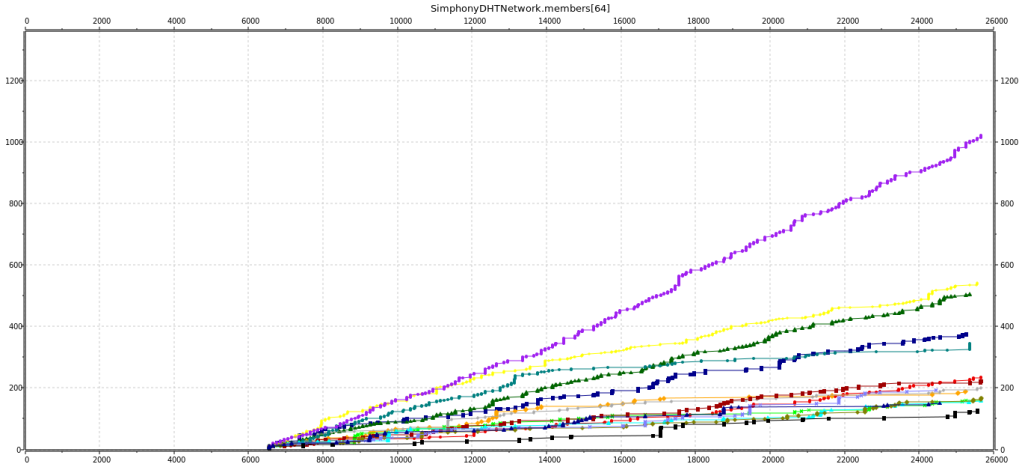

Figura 35: *Numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 256 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 36 e 37 vengono rappresentati il numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 256 nodi,

utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, nella versione modificata del protocollo il nodo 96 invia un numero di messaggi sempre inferiore a 250, mentre nella versione originale (a meno di due run di simulazione) invia un numero di messaggi sempre superiore a 300. Quindi anche in questo caso la versione modificata del protocollo risulta essere migliore.

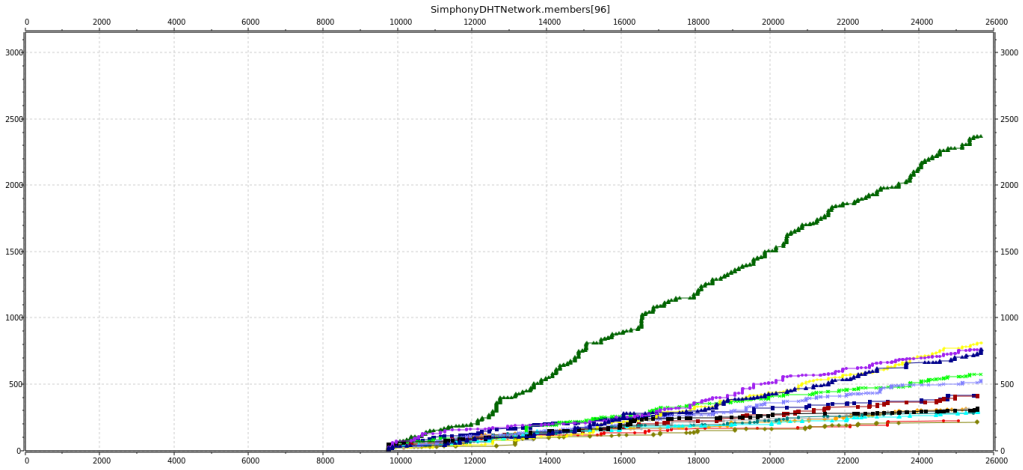

Figura 36: *Numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 256 nodi utilizzando il protocollo DHT Symphony*

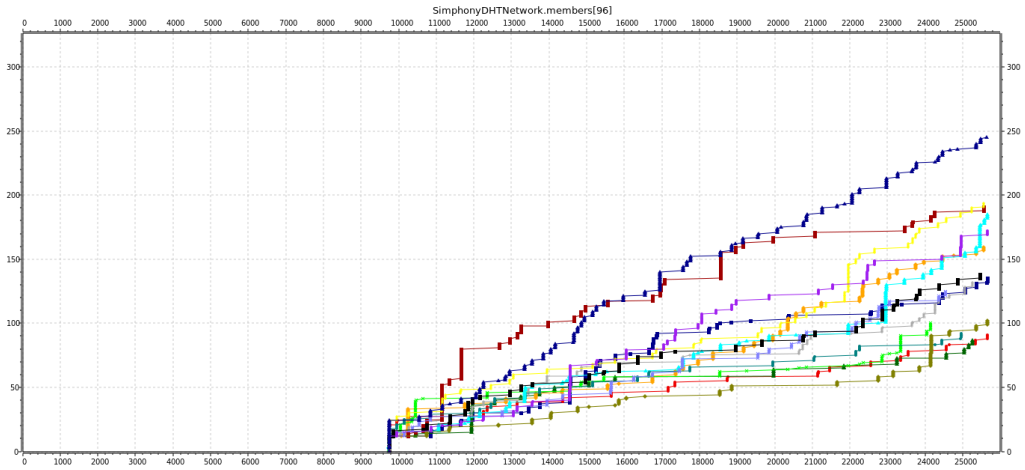

Figura 37: *Numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 256 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 38 e 39 vengono rappresentati il numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 512 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, nella versione modificata del protocollo il nodo 0 invia un numero di messaggi sempre inferiore a 800, mentre nella versione originale tale nodo invia più di 800 messaggi. Quindi anche in questo caso la versione modificata del protocollo risulta essere migliore.

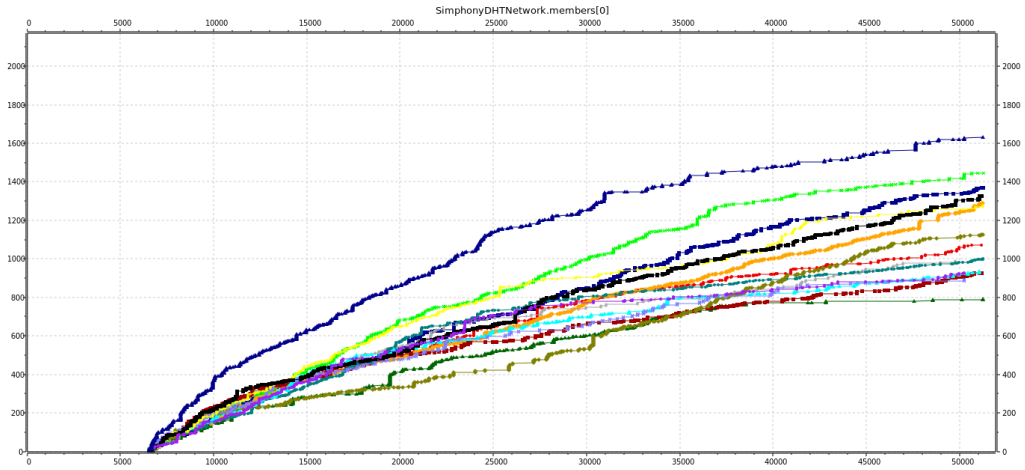

Figura 38: *Numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 512 nodi utilizzando il protocollo DHT Symphony*

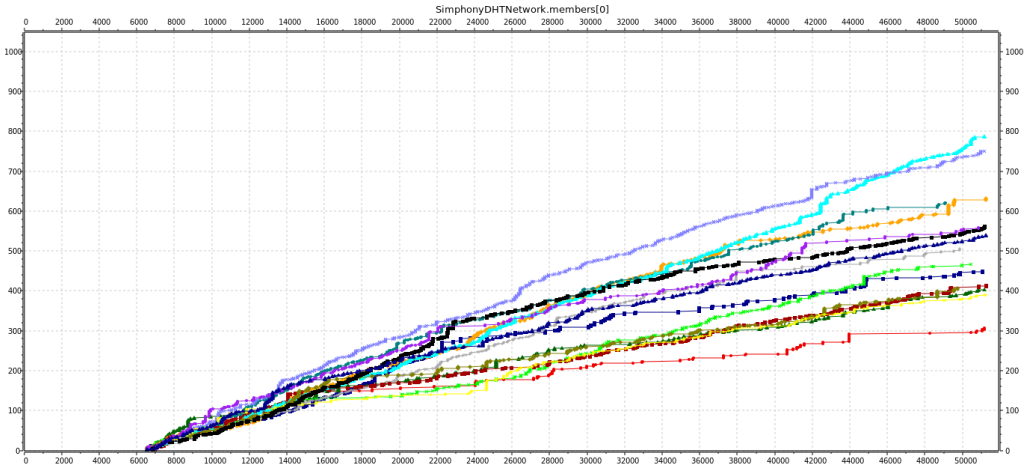

Figura 39: *Numero di messaggi inviati dal nodo 0 con access rate basso all'interno di una DHT con massimo 512 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 40 e 41 vengono rappresentati il numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 512 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione originale e modificata. Come si evince, nella versione modificata del protocollo il nodo 64 invia un numero di messaggi sempre inferiore a 700, mentre nella

versione originale tale nodo invia sempre più di 700 messaggi. Quindi anche in questo caso la versione modificata del protocollo risulta essere migliore.

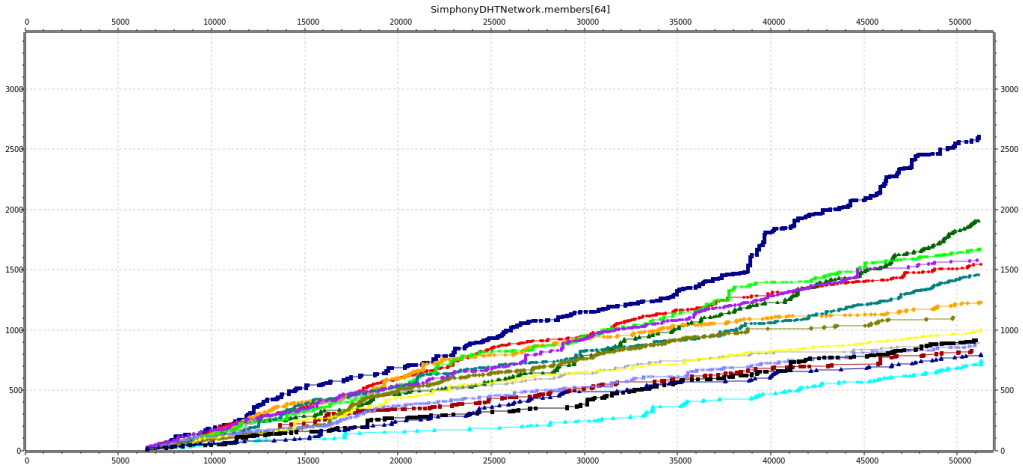

Figura 40: *Numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 512 nodi utilizzando il protocollo DHT Symphony*

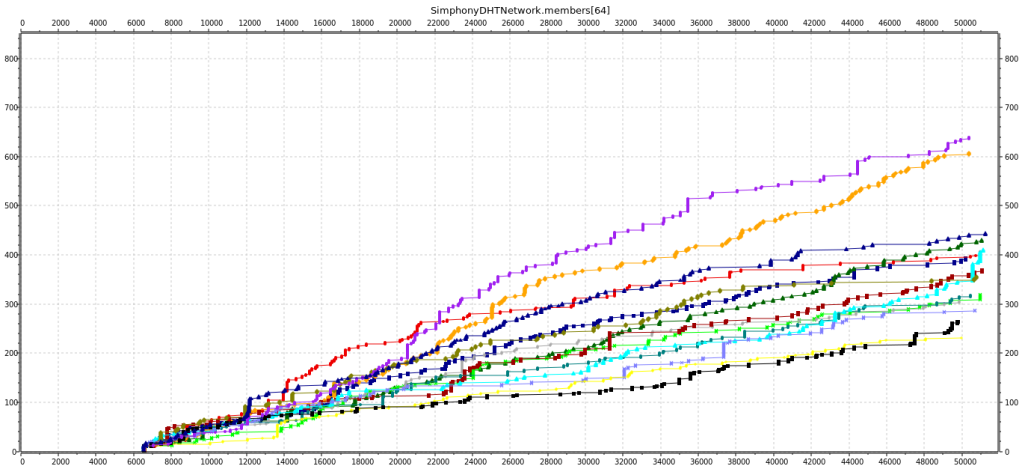

Figura 41: *Numero di messaggi inviati dal nodo 64 con access rate basso all'interno di una DHT con massimo 512 nodi utilizzando il protocollo DHT Symphony modificato*

In figura 42 e 43 vengono rappresentati il numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 512 nodi, utilizzando il protocollo DHT Symphony rispettivamente nella sua versione

originale e modificata. Come si evince, nella versione modificata del protocollo il nodo 96 invia un numero di messaggi sempre inferiore a 800, mentre nella versione originale tale nodo invia più di 800 messaggi (a meno di una run di simulazione). Quindi anche in questo caso la versione modificata del protocollo risulta essere migliore.

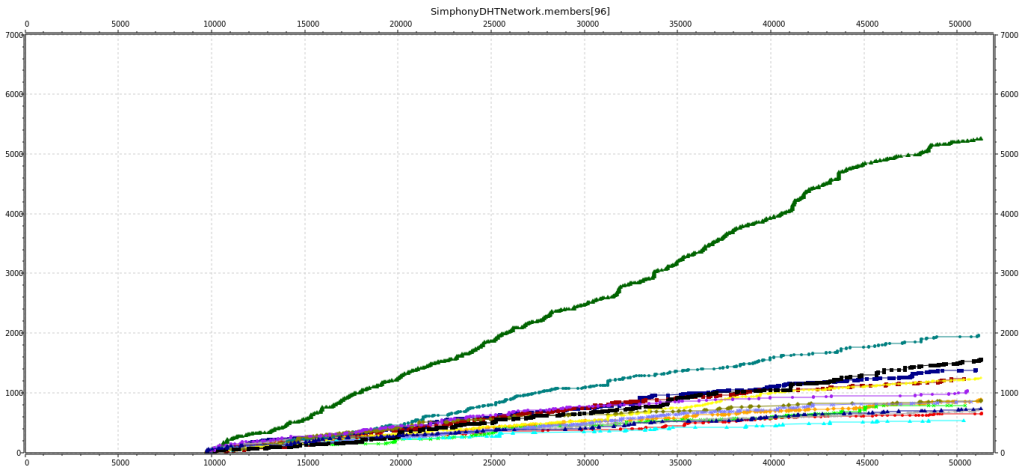

Figura 42: *Numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 512 nodi utilizzando il protocollo DHT Symphony*

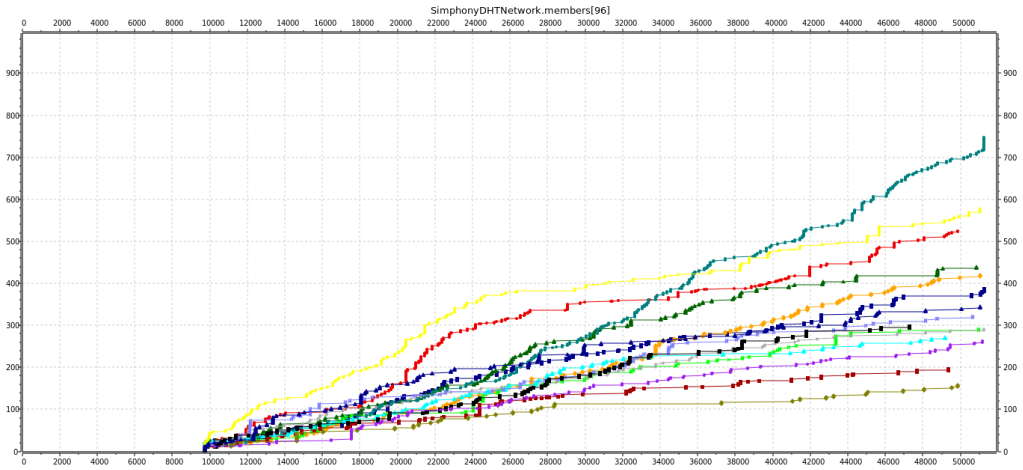

Figura 43: *Numero di messaggi inviati dal nodo 96 con access rate basso all'interno di una DHT con massimo 512 nodi utilizzando il protocollo DHT Symphony modificato*

Alla luce di tale analisi possiamo quindi affermare che la modifica protocollare proposta per il protocollo DHT Symphony comporta una diminuzione del numero medio di messaggi spediti dai nodi facenti parte della DHT analizzata.

Riferimenti bibliografici

[1] Gurmeet Singh Manku, Mayank Bawa, and Prabhakar Raghavan. 2003. Symphony: distributed hashing in a small world. In Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4 (USITS'03), Vol. 4. USENIX Association, Berkeley, CA, USA, 10-10.

[2] Omnet++ Network Simulation Framework - http://www.omnetpp.org/. Last visit 11/02/2013.

[3] Omnet++ TicToc Tutorial - http://www.omnetpp.org/doc/omnetpp/tictoc-tutorial/. Last visit 11/02/2013.