

Simulazione di una MANET con PeerSim

Stefano Piscella

Università di Bologna, Dipartimento di Scienze dell'Informazione

`stefano.piscella@studio.unibo.it`

Matricola : 0000739593

Abstract. Questo lavoro presenta i risultati ottenuti dalla simulazione di una MANET con PeerSim. L'analisi si concentra, principalmente, su come cambia la topologia della rete al variare di alcune caratteristiche della MANET.

1 Introduzione

Una MANET (Mobile Ad hoc NETwork) è una rete formata, in maniera cooperativa, da dispositivi mobili, connessi tra di loro senza fili e senza un'infrastruttura. Ogni nodo (dispositivo) in una MANET è libero di muoversi in qualsiasi direzione, questo porta ad una variazione continua delle connessioni tra dispositivi. Queste connessioni sono create sulla base di alcuni vincoli, come la vicinanza fisica (regolata da un limite del range di comunicazione) e il numero massimo di connessioni attive.

Quindi, l'obiettivo che si propone questo lavoro è quello di studiare, con l'ausilio di metriche comuni nella teoria dei grafi, come variando alcune caratteristiche della MANET, come il raggio di trasmissione o la strategia di rimozione di connessioni etc. . . , cambia la topologia della rete.

Nella prima parte di questo paper, si farà una breve introduzione a PeerSim ed all'implementazione della MANET con quest'ultimo. Si passerà poi ad analizzare e confrontare i diversi scenari utilizzati per la simulazione e i risultati ottenuti.

2 PeerSim

PeerSim [1] è un ambiente di simulazione di protocolli 'peer to peer' scritto in Java. È estremamente scalabile e supporta la dinamicità di questo tipo di sistemi. La sua struttura è basata su componenti facilmente configurabili (tramite file di configurazione) ed estendibili. PeerSim supporta sia una simulazione basata su 'cicli' e sia una basata su eventi.

Per lo sviluppo del modello di una MANET, si è scelto di utilizzare una simulazione basata su cicli in quanto, come già introdotto, andremo a studiare come cambia la topologia della rete e non, ad esempio, lo scambio di messaggi nella rete (che avrebbe potuto richiedere una simulazione ad eventi).

3 Implementazione

Vedremo ora come il modello è stato sviluppato su PeerSim, analizzando come sono stati implementati i componenti principali del simulatore e come configurarli. Verranno utilizzati concetti di PeerSim approfondibili in [1]

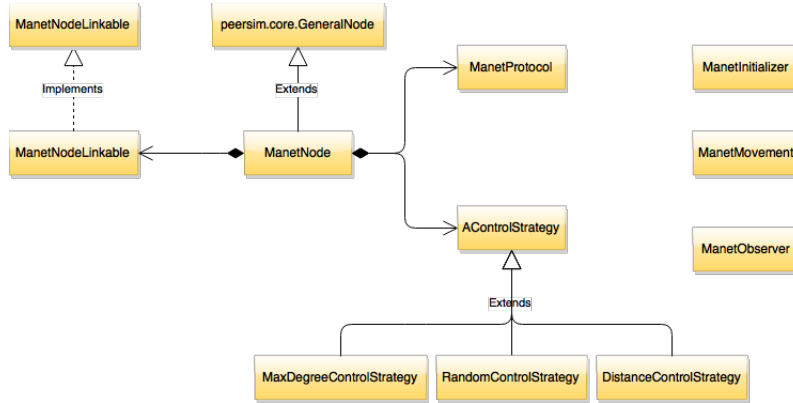


Fig. 1. Struttura del package implementato

3.1 Descrizione del package

In figura 1 possiamo osservare la struttura del modello implementato, che andremo a specificare di seguito fornendo una breve descrizione per ogni classe.

ManetNode La classe *ManetNode* estende *GeneralNode*, che rappresenta l'implementazione di un nodo generico fornita da PeerSim, aggiungendo informazioni come la lunghezza del raggio, che identifica l'area di copertura del nodo, le coordinate x,y del nodo nello spazio, il grado massimo del nodo, la grandezza dello spazio, il riferimento alla strategia di rimpiazzamento delle connessioni e il riferimento alla rete locale del nodo (i suoi vicini).

ManetProtocol La classe *ManetProtocol* implementa l'interfaccia *CDProtocol*, che identifica un protocollo di una simulazione basata su cicli. Definisce il comportamento ciclico, per ogni nodo, di rimozione delle connessioni uscite dall'area di copertura e di creazione di nuove connessioni in accordo con la strategia di rimpiazzamento.

ManetInitializer La classe *ManetInitializer* implementa l'interfaccia *Control* ed offre la funzionalità di inizializzazione della rete. In particolare configura tutti i nodi della rete in base alle configurazioni stabilite.

ManetMovement La classe *ManetMovement* implementa l'interfaccia *Control* ed offre la funzionalità ciclica di generare un movimento random su ogni nodo della rete.

ManetObserver La classe *ManetObserver* implementa l'interfaccia *Control* e si occupa di raccogliere statistiche sulla rete per ogni ciclo.

AControlStrategy La classe astratta *AControlStrategy* espone l'interfaccia per oggetti che definiscono una strategia di rimpiazzamento. Per la simulazione di questo modello sono state sviluppate quattro tipi di strategia che verranno in seguito approfondite.

ManetNodeLinkable La classe *ManetNodeLinkable* ha il compito di tenere traccia della rete locale di ogni nodo. In particolare contiene la lista dei vicini e offre metodi per accedere a questi ultimi.

Per una descrizione più specifica sulle classi implementate, si rimanda ai javadocs che accompagnano l'implementazione del modello.

3.2 Configurazione

La configurazione del modello avviene seguendo gli standard di PeerSim. In figura 2 è possibile osservare la porzione del file di configurazione, utilizzato da PeerSim, relativo alla MANET. Qui possono essere definiti in maniera chiara e semplice i parametri della nostra rete da simulare.

```
network.node cs.project.node.ManetNode
# ::::: PROTOCOL :::::
protocol.lmanet cs.project.protocol.ManetProtocol
# ::::: INITIALIZERS :::::
init.Omanet_init cs.project.initializers.ManetInitializer
#default 1
init.Omanet_init.radius 4
#default 16
init.Omanet_init.fieldsize 16
#default 1
init.Omanet_init.maxdegree 6
#default "RANDOM"
init.Omanet_init.controlstrategy MIN_DISTANCE
# ::::: CONTROLS :::::
control.Omanet_movement cs.project.controls.ManetMovement
#int default 2
control.Omanet_movement.max_speed 3
#ind default 1
control.Omanet_movement.min_speed 0
# ::::: OBSERVER :::::
control.lmanet_observer cs.project.observers.ManetObserver
```

Fig. 2. Parte del file .cfg contenente le configurazioni per la MANET

4 Simulazione

4.1 Il Modello

Il modello che è stato simulato per questo lavoro è, appunto, una MANET. In specifico si è trattato una serie di nodi che si muovono liberamente, in maniera random, in uno spazio aperto di forma quadrata, con le caratteristiche di una torus.

Ogni nodo è caratterizzato da diversi aspetti:

- **Velocità**, Ogni nodo si muove per ogni ciclo ad una velocità random compresa tra due valori definiti nella configurazione.
- **Raggio**, Il raggio di copertura dei nodi è definito di misura standard per tutti i dispositivi della rete.
- **Grado massimo**, Tutti i nodi della MANET hanno lo stesso numero massimo di connessioni attivabili.
- **Strategia di rimpiazzamento**, La strategia di rimpiazzamento è comune a tutti i nodi e sarà approfondita nella prossima sezione.

4.2 Strategia di rimpiazzamento

Altro aspetto che caratterizza i nodi della rete è la strategia di rimpiazzamento. La strategia definisce il comportamento dei nodi quando, raggiunto il grado massimo, devono scegliere se creare, o meno, una connessione verso un nuovo nodo all'interno dell'area di copertura e, nel caso si creasse, scegliere il nodo con cui terminare la connessione.

Per fare test su una rete con queste caratteristiche, sono state definite quattro tipi di strategia, di seguito analizzate.

RANDOM La strategia RANDOM al presentarsi di una nuova possibile connessione, con il nodo che ha già raggiunto il grado massimo, elimina una delle connessioni attive e aggiunge la nuova.

Si è scelto di implementarla principalmente come misura di confronto con le altre strategie.

MAX_DEGREE L'idea è di dare la precedenza alla connessione con nodi con grado minore di altri. Quindi nel momento in cui c'è la possibilità di connettersi con un nuovo nodo, questa viene presa in considerazione solo se il dispositivo ha il grado minore di uno dei nodi già connessi, sempre se si è già raggiunto il numero massimo di connessioni.

Si è scelto di implementare la MAX_DEGREE per cercare di aumentare la connettività all'interno della MANET.

MAX_DISTANCE La strategia è la stessa vista sopra se non ché, invece di controllare il grado massimo, si rimuove la connessione con il nodo fisicamente più distante. L'idea è di mantenere la connessione con nodi che hanno meno probabilità di uscire dal raggio di copertura del nodo.

MIN_DISTANCE La strategia inversa alla MAX_DISTANCE, rimuove la connessione con il nodo fisicamente più vicino. In questo modo si vuole cercare di coprire il maggior spazio possibile con l'area di copertura del nodo e dei suoi vicini.

4.3 Scenari

Gli scenari su cui sono stati effettuati le simulazioni sono stati presi tra gli scenari più utilizzati raccolti in [2] e sono visualizzabili in tabella 1.

Agli scenari in [2] sono state aggiunte due caratteristiche:

- **Massimo grado**, che varia, nei vari esperimenti, nel range visualizzato in tabella 1
- **Velocità**, che assume un valore random, per ogni nodo, compreso tra gli estremi visualizzati in tabella 1

Si è cercato di scegliere scenari che potessero rispecchiare il più possibile situazioni reali.

Table 1. Scenari utilizzati per la simulazione

	Nodi	Grandezza dello spazio	Raggio di copertura	Massimo grado	Velocità
Scenario A	10	1km X 1km	250m	[1,...,9]	rand(0,2) km/s
Scenario B	50	1km X 1km	250m	[2,4,8,16,32]	rand(0,2) m/s
Scenario C	100	1km X 1km	250m	[2,4,8,16,32]	rand(0,2) m/s

4.4 Metriche di analisi

Per l'analisi dei dati raccolti sono stati tenuti in considerazione concetti noti della teoria dei grafi.

Grandezza del giant component Uno degli aspetti fondamentali che una MANET deve cercare di garantire è quello della connettività. A causa della forte caratteristica random che hanno queste reti (nodi in continuo movimento), non è sempre possibile avere sempre una rete completamente connessa. Per questo lavoro il concetto di 'connettività' è stato trattato come la grandezza del giant component, ovvero il numero di nodi nella componente connessa più grande della rete.

Coefficiente di clustering della giant component Numerosi protocolli di routing definiti per le MANET si poggiano sul concetto di cluster [3]. Si andrà a considerare il coefficiente di clustering solo della giant component, per il confronto tra le diverse strategie.

Diametro della giant component Sempre inerente ai protocolli di routing, si analizzerà il diametro del giant component. Utile per l'analisi del numero di hop che un ipotetico messaggio dovrà compiere per arrivare a destinazione.

Distribuzione dei gradi Sarà inoltre utilizzata la distribuzione media dei gradi all'interno della MANET al fine di identificare le condizioni che portano alla connettività (grandezza del giant component) completa.

4.5 Risultati

Di seguito verranno mostrati i risultati ottenuti dalla simulazione del modello sugli scenari prima discussi.

Su ogni scenario sono stati effettuati 10 esperimenti da 200 cicli l'uno e i dati mostrati nei grafici che seguiranno sono la media di tutti i valori raccolti (2000). Si è scelto di non effettuare la media, sui 10 esperimenti, ciclo per ciclo in quanto ritenuti indipendenti tra loro. I dati e i grafici sono stati elaborati con Anaconda [4].

Nelle figure 3, 4, 5 si può osservare come varia la grandezza del giant component al variare del grado massimo per le quattro strategie nei diversi scenari.

La figura 3 ci mostra che le diverse strategie si comportano in modo praticamente identico questo a causa dello scarso numero di nodi in rapporto alla grandezza dello spazio e, di conseguenza, non viene mai raggiunta la connettività massima. Per questi motivi, si è ritenuto non necessario approfondire ancora lo scenario A.

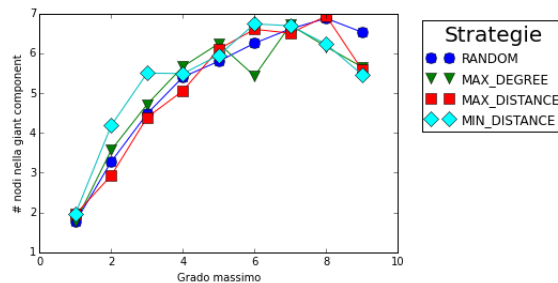


Fig. 3. SCENARIO A: Grandezza della giant component in rapporto al grado massimo

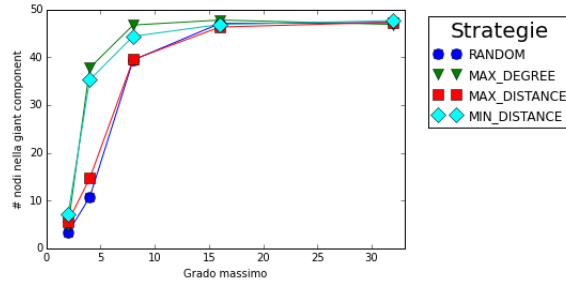


Fig. 4. SCENARIO B: Grandezza della giant component in rapporto al grado massimo

Osservando invece le figure 4, 5 si può osservare il diverso comportamento delle strategie. I due grafici sono molto simili tra loro e per questo analizzeremo solo il 5 in quanto è l'unico che raggiunge a pieno la connettività massima e mostra maggiori differenze tra le strategie.

Le principali caratteristiche emergenti dal grafico in figura 5 sono due:

Convergenza, tutte le strategie convergono ad una connettività piena con un grado massimo di 32.

MAX_DEGREE, com'era logico aspettarsi, la MAX_DEGREE è la strategia che prima crea una connettività massima. Nella scelta della strategia infatti era stato detto espressamente che lo scopo era quello di aumentare la connettività all'interno della rete. Come anche per la MIN_DISTANCE, che segue da vicino la precedente, il cui scopo era quello di coprire la maggior parte dell'area al fine di connettere il maggior numero di nodi.

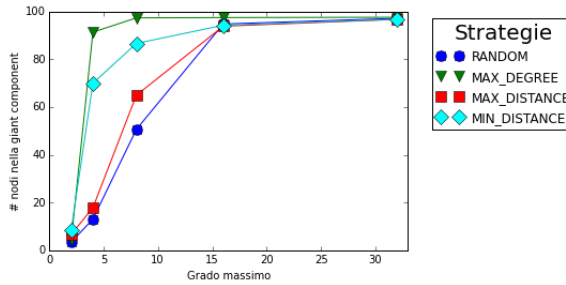


Fig. 5. SCENARIO C: Grandezza della giant component in rapporto al grado massimo

Osservando la figura 6, si nota come la grandezza del giant component nelle quattro strategie è proporzionato alla percentuale di distribuzione dei gradi con valori $\Theta(\log n)$ (in questo caso consideriamo in gradi ≥ 3). Infatti, come di-

mostrato in [5], per garantire una connettività completa in una rete wireless ogni nodo dovrebbe avere un numero di connessioni $\Theta(\log n)$.

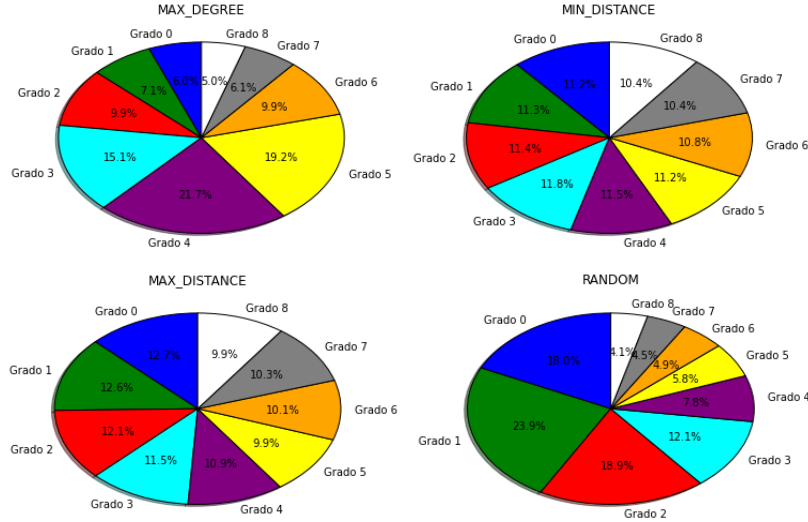


Fig. 6. SCENARIO C: Media della distribuzione dei gradi, con massimo grado pari a 8

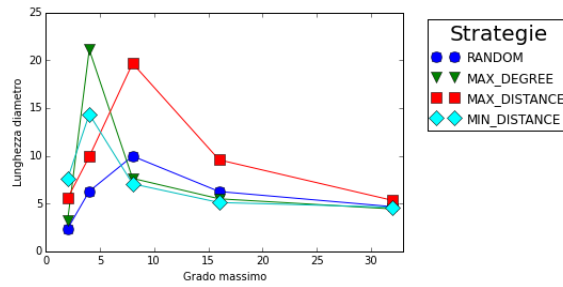


Fig. 7. SCENARIO C: Lunghezza del diametro della giant component in rapporto al grado massimo

Spostiamo ora l'analisi su un aspetto delle MANET di cui il modello, qui implementato, non tiene conto, ovvero lo scambio di messaggi. Nell'ambito di un calcolo del possibile numero di salti che un messaggio, scambiato tra due nodi, effettuerà è di sicura importanza il diametro del grafo.

In figura 7 possiamo osservare l'analisi della lunghezza del diametro nella giant component. Anche qui, come in figura 5 sono due gli aspetti emergenti:

Convergenza, tutte le strategie convergono ad un diametro di lunghezza $O(n)$ con un grado massimo di 32.

MAX_DEGREE, anche se in questo caso non è la strategia con il diametro più basso, è l'unica però che con massimo grado 8 raggiunge connettività piena e mantiene uno dei diametri più bassi. Anche in questo caso la MIN_DISTANCE segue da vicino.

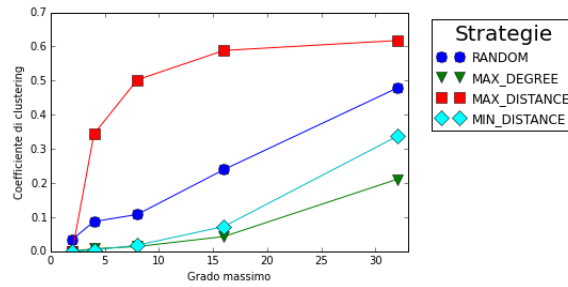


Fig. 8. SCENARIO C: Coefficiente di clustering della giant component in rapporto al grado massimo

Infine in figura 8 possiamo osservare l'andamento del coefficiente di clustering del giant component al crescere del grado massimo.

Anche qui i risultati rispecchiano quelli che erano gli obiettivi delle strategie. Infatti la strategia con il più alto coefficiente è la MAX_DISTANCE che con l'eliminazione delle connessioni con dispositivi più distanti ha più probabilità di creare gruppi di nodi connessi tra loro (cluster).

5 Conclusioni e lavori futuri

Abbiamo quindi implementato e analizzato una MANET con il simulatore Peer-Sim. In particolare si è cercato di trovare una strategia di rimozione ottimale rispetto alle quattro definite.

Possiamo concludere quindi sostenendo che la strategia 'migliore' dipende soprattutto dal protocollo di routing che verrà implementato sulla MANET. Dai risultati comunque si può affermare che se le connessioni massime di ogni nodo fossero basse, nello scenario $C \leq 8$, la strategia emergente sarebbe la MAX_DEGREE mentre se avessimo un numero più alto di connessioni possibili, nello scenario $C \geq 16$, allora sarebbe da valutare la MAX_DISTANCE grazie all'alto coefficiente di clustering.

Il lavoro, qui discusso, ha analizzato sommariamente anche l'aspetto legato ai protocolli di routing per le MANET. Sarebbe sicuramente interessante espandere

il modello implementato permettendo lo scambio di messaggi tra nodi al fine di testare diversi protocolli e comparare, magari confermando, i risultati con quelli di questo lavoro.

Bibliografia

1. PeerSim HOWTO: Build a new protocol for the PeerSim 1.0 simulator. <http://peersim.sourceforge.net/tutorial1/tutorial1.html>. Last visited 24 June 2015.
2. S. Kurkowski, T. Camp, M. Colagrosso. MANET Simulation Studies: The Incredibles. Mobile Computing and Communications Review, Volume 9, Number 4, 2005
3. R. Agarwal, Dr M. Motwani. Survey of clustering algorithms for MANET. International Journal on Computer Science and Engineering Vol.1(2), 98 - 104, 2009
4. Anaconda Scientific Python Distribution. <https://store.continuum.io/cshop/anaconda/>. Last visited 24 June 2015.
5. F. Xue, P.R. Kumar. The Number of Neighbors Needed for Connectivity of Wireless Networks. Wireless Networks 10, 169–181, Kluwer Academic Publishers, 2004.