Indice

1. L'idea di partenza
2. I protocolli attualmente definiti
2.1 Specifiche del protocollo AODV
2.1.1 Considerazioni
2.2 Specifiche del protocollo TORA
3. Il gossip
3.1 Gossip vs Flooding
3.2 Il Gossip in reti di varie dimensioni
3.2.1 Comportamento del Gossip in reti regolari
3.2.2 Comportamento del Gossip in reti "random"
3.3 Analisi dei risultati ottenuti
4. Applicazione del Gossip al protocollo AODV
5. Applicazione del Gossip al protocollo TORA
5.1 Casi possibili
5.2 Analisi e possibili soluzioni

1. L'idea di partenza

L'idea di utilizzare il Gossip nelle reti wireless Ad Hoc nasce da un osservazione del mondo in cui viviamo.
Nella nostra società capita spesso di conoscere qualcosa perché ci è stato riferito. Se poi noi stessi lo riferiamo a qualcun altro e via dicendo, la conoscenza di tale informazione diventa sempre più consistente.
Basti pensare che spesso si sa qualcosa pur non essendosi mai interessati direttamente di quell'arogomento. Lo si conosce perché lo si è sentito da qualche parte. E' quel che si chiama comunemente pettegolezzo.
Analizzando il concetto di pettegolezzo all'interno della società si possono trarre quindi delle considerazioni importanti:

1. La diffusione dell'informazione avviene in modo casuale
2. Non è importante che tutti sappiano tale informazione
3. Non è utile verificare che tutte le persone effettuino Gossip, ma è importante che a livello dell'intero sistema l'informazione sia in generale conosciuta
4. L'insieme delle informazioni dette dai singoli porta al conseguimento dell'obbiettivo in modo costruttivo ed efficace

Il Gossip è quindi uno strumento molto utile di diffusione di un' informazione e, da non sottovalutare, è a costo zero.
Anche nel regno animale, seppur con alcune differenze, vi è un comportamento simile; ad esempio nella ricerca di cibo da parte delle formiche.
La formica è un animale quasi cieco con la particolarità di lasciare una traccia chimica di sostanze mentre cammina, i così detti ferormoni. Nonostante la vista sia pessima, possiede dei sensori che riescono a percepire in modo efficiente questi odori.
Utilizzando questa caratteristica si possono generare comportamenti collettivi complessi. Laddove i ferormoni sono preponderanti sarà identificato il cammino minimo da seguire per arrivare al cibo.
Quindi l'intelligenza può stare non nel cervello della formica, ma nell'interazione tra tanti cervelli individualmente piccoli.

Nonostante entrambi i comportamenti siano molto semplici, il risultato finale è raggiunto con estrema efficacia.
Analizzando quando detto nel campo dell'intelligenza artificiale si nota come sia possibile risolvere in modo efficiente diversi tipi di problemi. L'intelligenza a cui si riferisce è essenzialmente di tipo distribuito. Il problema, la diffusione di un informazione o la ricerca del percorso, è suddiviso tra vari moduli distribuiti comunicanti tra loro.
Il punto di partenza è l'esistenza di entità autonome (agenti) in grado di risolvere un problema e ciò che è utile valutare è l'efficacia dei singoli a livello dell'intero mondo. I sistemi non sono rappresentabili in termini dei componenti individuali ma sono la composizione di entità in grado di interagire tra loro. Il problema complessivo viene risolto componendo le capacità risolutive dei singoli agenti coordinandone le attività in modo da raggiungere lo scopo prefissato (goal).

Il possibile utilizzo del Gossip spazia dalla robotica alle telecomunicazioni, con applicazioni che vanno dagli spray computer al routing nelle reti fisse o mobili.
Ad esempio sarebbe interessante studiare una possibile ottimizzazione, tramite l'utilizzo del Gossip, del processo di auto-organizzazione dei nodi all'interno degli spray computer (applicazioni come il mantello dell'invisibilità o il muro invisibile)
Occupandosi di telecomunicazioni ci concentreremo sull'analisi di una possibile implementazione in questo ambito, ove il trasferimento di informazioni in modo efficace è un problema molto sentito. Basti pensare a Sensor Networks, Internet e reti P2P, Telefonia mobile, Wireless Networks, ecc..
Si è pensato che applicare il Gossip a reti di tipologia wireless Ad Hoc potesse risolvere problemi legati alla sovrabbondanza dei messaggi di segnalazione spesso inutilmente generati. Il funzionamento si basa sull'idea che i nodi della rete effettuino Gossip in modo costruttivo. Così come per le persone vi è una sorta di propagazione casuale dell'informazione all'interno della rete, che permette di trovare impostando opportuni parametri, il percorso tra un nodo sorgente ed un nodo destinazione.
Nei capitoli seguenti verrà discusso in dettaglio il funzionamento di tale algoritmo in reti di piccole e grandi dimensioni.

2. I protocolli attualmente definiti

Prima di approfondire cosa è l'algoritmo di Gossip e di definirne il comportamento è utile fare una panoramica di quali sono attualmente i algoritmi di routing utilizzati nelle reti wireless Ad Hoc.
Gli algoritmi di instradamento per una rete Ad Hoc devono essere in grado di fornire il corretto percorso che un pacchetto deve seguire adattandosi, allo stesso tempo, ai frequenti e impredicibili mutamenti della topologia della rete. Questo fa sì che l' ammontare del traffico di segnalazione, necessario ad un algoritmo di instradamento distribuito, sia molto elevato, in contrasto con la necessità delle reti wireless di minimizzare l' utilizzo delle risorse di comunicazione. La maggior parte degli studi svolti sugli algoritmi di instradamento per reti ad hoc mirano quindi a trovare il modo di diminuire il traffico di segnalazione prodotto dal livello di routing.

Gli algoritmi di instradamento esistenti possono essere classificati in tre categorie: algoritmi proactive, reactive e hybrid.
Gli algoritmi di tipo ***proactive*** mantengono costantemente aggiornate le informazioni di instradamento tramite scambi di pacchetti a intervalli temporali fissi. Questo permette di avere l' instradamento immediatamente disponibile ad ogni richiesta di routing. Lo svantaggio è che gli algoritmi proactive producono traffico di segnalazione anche quando non viene trasmesso nessun pacchetto dati; ciò può causare problemi di sovraccarico nella rete, specie se i nodi si spostano velocemente.
Nei protocolli di tipo ***reactive*** viene invocata una procedura per determinare il corretto instradamento solo nel momento in cui il pacchetto deve essere trasmesso (on-demand). In questo modo si riduce il traffico di segnalazione a scapito di un aumento dei tempi di
consegna.
Il terzo tipo di protocolli, ***hybrid,*** cerca, come dice il nome, di unire i vantaggi di entrambi i protocolli precedenti, restringendo l'applicazione di algoritmi proactive ai soli vicini del nodo che vuole trasmettere il pacchetto.
In figura 1 è riassunto quanto detto.

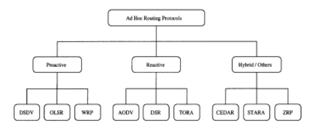


Fig.1 Classificazione dei protocolli di routing Ad Hoc

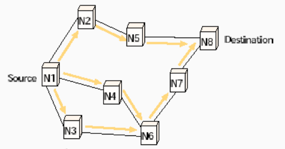
Una suddivisione alternativa alla precedente può essere fatta in base alla topologia della rete, che può essere "gerarchica" (hierarchical) o "piatta" (flat). In una rete "gerarchica" i nodi sono partizionati in gruppi detti cluster.Per ogni cluster è selezionato un "cluster head" attraverso i quali passa il traffico della rete.
In una rete ad hoc "piatta" non è prevista nessuna centralizzazione. Due nodi sono connessi se le condizioni radio sono tali da permettere al nodo destinazione di sentire la trasmissione del vicino (ovvero se due nodi sono in copertura radio).
Un vantaggio della rete "piatta" è la possibilità di stabilire più di un percorso tra nodo sorgente e destinazione; questo permette di valutare in modi diversi quale collegamento è da preferire, a seconda delle richieste e dell' utilizzo della rete.
Il vantaggio della rete "gerarchica" è invece quello di minimizzare il numero dei pacchetti di routing che vengono scambiati tra i nodi di uno stesso cluster e tra i cluster head.
Di seguito viene fornita una descrizione della struttura del protocollo di routing AODV e del protocollo TORA (di tipo reactive) in quanto è a tali protocolli che verrà applicato l'algoritmo di Gossip.

2.1 Specifiche del protocollo AODV

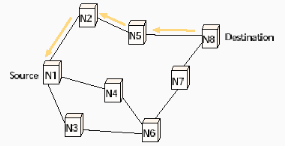
Caratteristiche principali:

- Ogni nodo cerca un path solo quando ne ha bisogno
- Uso del Sequence Number per tenere traccia dell'accuratezza delle informazioni
- Ogni nodo tiene traccia solamente del primo hop invece di tutto il cammino
- Tramissione periodica di un messaggio HELLO per riconoscere/confermare l'esistenza dei vicini

Il protocollo Ad hoc On-demand Distance Vector (AODV) [1] è un protocollo di routing di tipo reactive basato sull' algoritmo Distance Vector che permette lo scambio di messaggi tra i vari nodi anche in condizioni di non visibilita' radio diretta fra source e destination.
Il cuore del protocollo e', ovviamente, una tabella di routing dinamica generata automaticamente che permette di instradare i pacchetti con successo. Su tale tabella viene fatta un'analisi per il shortest path ed un'altra per evitare loops. In caso di disconnessione da parte di un nodo dal sistema il protocollo riaggiorna la tabella di routing evitando cosi' situazione di errore.
Una caratteristica fondamentale del protocollo è quella di utilizzare "numeri di sequenza", i quali forniscono ai nodi uno strumento per valutare quanto sia aggiornato un determinato percorso. Un terminale che si trovi a dover scegliere tra più percorsi verso una certa destinazione sceglierà quello caratterizzato dal numero di sequenza maggiore, corrispondente ad un informazione di routing più recente.
In figura 2 è mostrato un esempio di come il protocollo calcola il percorso tra i nodi N1 ed N8.



(a) Propagation of the RREQ



(b) Path of the RREP to the source

Fig. 2 Percorso tra nodo N1 ed N8

2.1.1 Considerazioni

L'algoritmo, a causa del flood ad ogni ricerca di route, funziona adeguatamente per reti di piccole dimensioni (cinquanta o cento nodi al massimo),dove offre un basso overhead dovuto al fatto che i percorsi vengono cercati solamente on-demand.
Non è garantito che i percorsi siano i più corti, e, a causa del funzionamento on demand, vari pacchetti dati vengono persi finchè l'algoritmo non trova un percorso per la destinazione.

2.2 Specifiche del protocollo TORA

Il TORA (Temporally Ordered Routing Algorithm) [1] basa il suo funzionamento su una precisa sequenza temporale di eventi per cui i nodi su cui gira questo algoritmo devono disporre di orologi sincronizzati, ad esempio con un protocollo di rete come "Network Time Protocol".
L' algoritmo cerca di costruire un grafo diretto aciclico (DAG) dalla sorgente alla destinazione che viene considerata come radice, quindi con altezza zero, mentre gli altri nodi del grafo aciclico hanno un'altezza pari al numero di nodi da attraversare per arrivare alla radice.
In questo modo l'algoritmo trova potenzialmente più di un percorso per arrivare a destinazione, quindi la manutenzione dei percorsi in caso di guasto di un nodo o del suo spostamento è semplicata, in quanto nella maggior parte dei casi non bisogna ripartire con la richiesta di un nuovo percorso, ma utilizzare uno di quelli rimasti.
Il funzionamento è mostrato in figura 3[1].

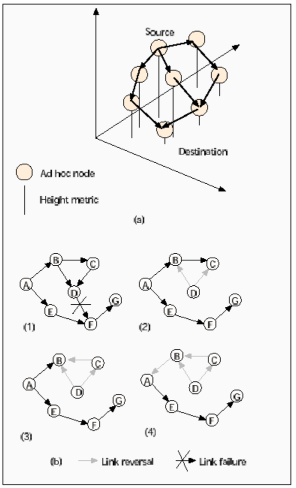


Fig.3.
(a) creazione del percorso di Routing (sono mostrati i link direttamente assegnati)
(b) mantenimento del percorso

Sebbene una volta trovato un percorso multiplo l' algoritmo necessiti di meno manutenzione, è stato osservato che il tempo necessario all' algoritmo per creare il grafo diretto aciclico è notevole e ne compromette le prestazioni.
E' anche vero che utilizzando il DAG si ha un notevole vantaggio in reti di grandi dimensioni con una grande densità di nodi.

3. Il gossip

3.1 Gossip vs Flooding

Poichè il flooding è spesso l'elemento base di alcuni protocolli di routing è molto interessante confrontare il gossiping con esso[3].
Il protocollo di gossip di base è molto semplice. Un nodo sorgente trasmette una richiesta di percorso con probabilità 1. Quando un primo nodo riceve la richiesta,la trasmette con probabilità *p* in modo broadcast ai nodi vicini e con probabilità *1-p* la scarta. Se successivamente la riceve nuovamente la scarta senza trasmettere nulla. Quindi, un nodo trasmette al più una sola volta la medesima richiesta.
Questo semplice protocollo è denominato *GOSSIP1(p)* ed ha un problema di condizioni iniziali. Se la sorgente ha relativamente pochi nodi vicini vi è la possibilità che nessuno di loro effettui gossip con il conseguente fallimento dell'algoritmo.
Per far si che questo non si verifichi, viene trasmesso con probabilità 1 per i primi *k* nodi prima di continuare con probabilità p.
Denominiamo questo protocollo modificato *GOSSIP1(p,k)*. Le prestazioni dipendono ovviamente dalla scelta di p e di k (con p,k=1 sia ha il comune flooding).
In generale ciò che accade è legato alla tipologia della rete considerata, al numero di collegamenti possibili da ogni nodo, alla probabilità p ed alle condizioni iniziali determinate da k.
Fondamentalmente le reti analizzate sono di due tipi:

- Reti a topologia regolare: reti dove è facile analizzare il comportamento del *GOSSIP1* impostandone diversi parametri, quali ad esempio il valore di *k,* la probabilità *p* e la dimensione della rete.
- Reti "random": reti che vengono costruite posizionando i nodi in modo casuale in una area di dimensioni (x,y). Sono più utili per modellare le caratteristiche di una reale Ad Hoc network.

Per entrambe valgono le considerazioni fatte nell'introduzione (potenza di trasmissione limitata, banda limitata, ecc..).

3.2 Il Gossip in reti di varie dimensioni

In questa sezione si analizza il comportamento dell'algoritmo di Gossip in reti regolari ed in reti casuali, più vicine al comportamento delle reti realmente utilizzate.

3.2.1 Comportamento del Gossip in reti regolari

Ci poniamo in un caso già trattato in letteratura [3]così da analizzare i pregi ed i difetti dell'algoritmo. Impostando i seguenti parametri:

- Gossip(p,4)
- rete regolare di 1000 nodi (una griglia di 20 x 50)
- la sorgente si trova nella colonna 10.
- propagazione ideale
- una centinaia di simulazioni così da avere una buona base da analizzare.

Con probabilità p=0.72 si ottiene il grafico di figura 4.

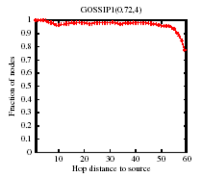


Fig.4 Gossip con p=0.72

In dettaglio, il grafico mi indica quanti nodi (in percentuale) ad una certa distanza dalla sorgente ricevono la richiesta di percorso.
Con p=0.72 la maggior parte dei nodi a distanza minore di 50 riceve correttamente la richiesta.
Con probabilità p=0.60 invece si nota subito un peggioramento delle prestazioni (fig.5):

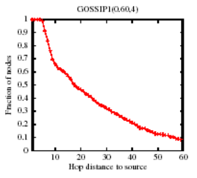


Fig.5 Gossip con p=0.60

Un ulteriore grafico interessante è quello di figura 6 dove il numero di esecuzioni viene legato al numero di nodi che riceve correttamente la richiesta di routing.

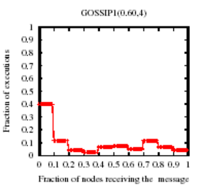


Fig.6 Gossip con p=0.60

Solo nel 4% delle esecuzioni dell'algoritmo un 90% dei nodi riceve il messaggio e in oltre il 50% la richiesta viene ricevuta da poco meno del 20% dei nodi.

3.2.2 Comportamento del Gossip in reti "random"

Le reti regolari non sono un valido modello di ciò che realmente succede in quanto i nodi possiedono una certa mobilità all'interno dell'area considerata e vi sono molti effetti negativi causati dalla vicinanza dei suddetti al bordo della rete.
Utilizzando il Gossip del precedente paragrafo in una rete di dimensioni elevate (dell'ordine del Km quadro) contenente 1000 nodi posizionati in modo casuale si ottiene il seguente grafico (figura 7):

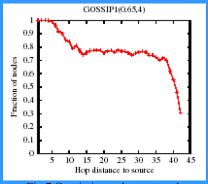


Fig.7 Gossip in random network

Come ci si aspettava si ha un peggioramento delle prestazioni man mano che ci si sposta verso il bordo della rete. Ad una distanza di 40 poco meno del 50% dei nodi riceve correttamente il messaggio.

3.3 Analisi dei risultati ottenuti

Come si deduce dal precedente paragrafo, uno dei problemi fondamentali del Gossip è il fatto che se la probabilità *p* assume un valore troppo piccolo la richiesta di routing si estingue prima di giungere a destinazione. Ancor più grave è il fatto che i nodi della rete non hanno modo di accorgersi di ciò. Tenendo conto del fatto che se pur un nodo non propaga la richiesta (ad esempio a causa di una coda di processi) i nodi vicini, supponendo che il messaggio non si sia estinto, effettuano il Gossip.
Questo suggerisce una possibile ottimizzazione. Se un nodo avente n nodi vicini riceve una richiesta di percorso ma non la propaga e dopo un certo tempo ragionevole t, non riceve la richiesta da parte di altri m nodi vicini, la propaga a tutti gli n ed m nodi.
Chiameremo questo algoritmo modificato *GOSSIP1(p,k,m)* ed il problema fondamentale al fine di evitare un eccessivo numero di messaggi sarà la scelta di *m.*
Ulteriori ottimizzazioni non vengono prese in considerazione.

4. Applicazione del Gossip al protocollo AODV

Come spiegato nell'introduzione si è scelto di valutare le prestazioni del Gossip unito all'algoritmo di routing AODV in reti di piccole dimensioni in quanto è possibile trovare in letteratura prove effettuate al simulatore [7] [10]. Tra i parametri che vengono impostati i più importanti sono:

- 150 nodi posizionati in modo random all'interno di un'area di circa 2 km quadrati.
- traffico di tipo CBR.
- modello di propagazione a due raggi [11].
- altre condizioni che rispecchiano un'ipotesi reale di funzionamento.

Di particolare interesse è l'analisi dei parametri di quella che viene chiamata *expanding-ring search*.
Tale impostazione riguarda solo l'AODV puro e, più precisamente, in primis viene propagata una richiesta di percorso in una zona avente raggio pari a 5 nodi (valore di default del simulatore) e se non si ottiene risposta viene effettuato flooding su tutta la rete.

Tali condizioni rappresentano una stima del caso peggiore delle prestazioni del Gossip.
Dai risultati ottenuti è possibile realizzare i seguenti grafici (fig.8) discussi poi in dettaglio.

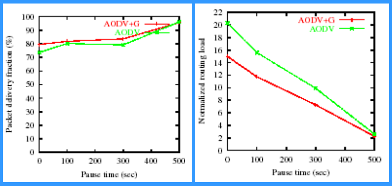


Fig.8 Risultati ottenuti

Il primo grafico indica il rapporto tra il numero di pacchetti consegnati e quelli generati dalle sorgenti CBR (throughput)
Il secondo mi indica invece il carico della rete, cioè il numero di pacchetti trasmessi per quelli arrivati a destinazione.

Senza sorprese, le prestazioni aumentano quando il carico della rete diminuisce (come si nota anche in figura.8).
Come si può notare utilizzando il AODV+G si ha un incremento (al tempo t=0) del throughput del 8% ed una riduzione del carico della rete che varia lungo tutto il periodo dal 14% al 27%.
Un ulteriore verifica sull'incremento delle prestazioni apportate dal Gossip lo si nota in termini di ritardo nella consegna dei pacchetti. Ritardo dovuto a qualsivoglia fenomeno. Dai risultati ottenuti si riscontra un guadagno al tempo t=0 del 36%.

Come detto prima un elemento importante di valutazione è l'analisi del comportamento della *expanding-ring search*.
Nel caso di reti di piccole dimensioni è facile che trovi il percorso con facilità, ma se le dimensioni della rete crescono non è più così scontato. Questo comporterebbe un nuovo flooding a livello dell'intera rete con la conseguente generazione di un numero elevatissimo di messaggi che influirebbero in modo pesante sulle prestazioni.
Utilizzando invece la tecnica del Gossip ciò non avviene. Si evince che anche in reti di grandi dimensioni l'utilizzo del Gossip comporterebbe un notevole aumento delle prestazioni. Nel Capitolo seguente verrà analizzato il Gossip applicato all'algoritmo TORA, più adatto a reti di grandi dimensioni.

5. Applicazione del Gossip al protocollo TORA

Come citato nei paragrafi precedenti la particolarità del protocollo TORA è quella di mantenere memorizzati oltre al percorso sorgente destinazione scelto, anche gli ulteriori percorsi così da essere subito pronto ad esempio nel caso di perdita di un link. Nonostante questa caratteristica essendo anch'esso un protocollo che utilizza la tecnica del flooding si ha un sovraccarico della rete dovuto alla propagazione di messaggi non necessari.

Nel paragrafo successivo si discuterà di come il Gossip porti ad un sostanziale miglioramento anche in questo tipo di protocollo.

5.1 Casi possibili

Utilizzando reti di grandi dimensioni la sovrabbondanza di messaggi propagati influenza in modo drastico le prestazioni. Come per il protocollo AODV è necessario che la probabilità con cui si effettua gossip sia sufficiente a portare la richiesta di percorso dal nodo sorgente al nodo destinazione.

Un analisi di quella che potrebbe essere la probabilità *p* con cui si effettua Gossip all'interno di una rete che utilizza il protocollo TORA è la seguente:

- *CASO 1:* *p* con valore eccessivamente piccolo: il protocollo TORA + Gossip non garantisce che la richiesta di percorso arrivi a destinazione. Il messaggio si estingue prima ed il funzionamento del protocollo ne è compromesso.
- *CASO 2: p* con un valore sufficientemente adeguato: Il messaggio arriva a destinazione senza problemi e vengono memorizzati solo alcuni cammini alternativi tra sorgente e destinazione.
- *CASO 3: p* con un valore adeguato: Il messaggio arriva a destinazione e viene memorizzata la maggior parte dei percorsi alternativi tra sorgente e destinazione.
- *CASO 4: p* con valore pari ad 1. Si effettua praticamente il comune flooding.

5.2 Analisi e possibili soluzioni

Osservando i casi sopra citati si vede come la scelta della probabilità *p* sia vincolata alla gestione della rete. In particolare, scartando il caso 1 che non da nessun tipo di vantaggio (anzi attua un malfunzionamento) si possono analizzare gli altri tre casi.

Se la rete in cui ci troviamo è una rete diciamo stabile, in cui la mobilità degli utenti è relativamente bassa e le condizioni del canale sono buone è possibile effettuare Gossip con una probabilità *p* non troppo elevata. Ovviamente tenendo conto di pochi percorsi alternativi il punto di forza del protocollo originale TORA è limitato.

Se invece siamo in una rete di dimensioni più grandi ed il numero di nodi è elevato ci si propone di avere a disposizione un numero di percorsi alternativi più elevato così da gestire gran parte della mobilità dei vari nodi. Anche in questo caso non si ha la certezza che a causa della rottura di un link o dello spostamento di un utente si sia memorizzato il percorso alternativo sorgente-destinazione ma diciamo che la probabilità gioca a nostro favore. Nel caso sia un link non recuperabile sarà necessario rieffettuare il calcolo del percorso di routing perdendo quindi il vantaggio del protocollo TORA.

Riassumendo la scelta della probabilità *p* va effettuata tenendo conto principalmente di:

1. caratteristiche della rete in questione (numero di nodi, mobilità degli utenti,ecc..).
2. specifiche sull'efficienza che si vuole ottenere (fisso una specifica sull'efficienza che desidero avere e calcolo la probabilità con cui è necessario effettuare Gossip).
3. specifiche di manutenzione della rete e quindi del numero di percorsi alternativi memorizzabili.