

## MSA - 13.04.2021

Riprendiamo da dove ci siamo fermati. Stavamo parlando di approcci distribuiti alla risoluzione del problema della mobilità. La volta scorsa abbiamo dato le motivazioni generali, per cercare soluzioni alternative a quelle più gerarchiche e centralizzate che avevamo discusso nelle lezioni precedenti. Le argomentazioni generali le abbiamo già discusse, vediamo soluzioni che sono in corso di esame ed in particolare in ambito IETF. Ci sono due famiglie di soluzioni che discuteremo: quella basata sulla costruzione di tunnel che servono ad instradare i messaggi che si scambiano un nodo mobile con un suo partner, vedremo due variazioni sul tema, e dopo l'altra possibilità basata sul principio del routing personalizzato per ogni singolo nodo mobile. Non ricordo se lo avevo già rimarcato, ma nel dubbio puntualizzo adesso che queste due soluzioni non hanno la stessa valenza. La prima del tunneling risolvono il problema della raggiungibilità di un nodo mobile ma per quanto riguarda sessioni che sono già attive nel momento in cui il nodo mobile cambia la sua posizione, sono soluzioni parziali. Non risolve il problema nel caso di nodo inattivo. La seconda soluzione è non nuova, ne abbiamo già parlato, è una soluzione globale.

Vediamo la prima. L'idea della soluzione è che il punto di ancoraggio del nodo mobile, il punto verso cui chi interagisce con il nodo indirizza i suoi messaggi, il punto di ancoraggio coincide con l'access router della rete in cui si trova il nodo mobile nel momento in cui inizia una sessione verso il suo partner. Sono soluzioni limitate alla continuità della sessione, l'ulteriore specificazione è che si parla di sessioni iniziate dal dispositivo mobile. Tutto questo vale fintanto che il nodo mobile attiva una sessione e per la durata della sessione rimane nella stessa rete. Cosa succede, essendo il nodo mobile, se si sposta in una sotto rete diversa in cui l'access router è diverso? L'idea in questo caso è che il partner del nodo mobile continua ad utilizzare l'indirizzo IP che il nodo mobile aveva al momento dell'avvio della sessione, continua ad inviare i pacchetti verso la precedente rete del nodo mobile. L'access router di quella rete deve fare il forward dei pacchetti verso la nuova rete in cui si trova la rete mobile. Il nodo mobile deve informare l'AR della nuova rete in cui si trova il nodo ed informare il vecchio della nuova posizione. Il forward avviene tramite il classico meccanismo del tunnel. Per quanto riguarda sessioni attivate nella sessione attuale e si concludono prima di un cambio di posizione, le sessioni non hanno bisogno di essere incapsulate in un tunnel. Sessioni che invece perdurano al di là di un cambio di posizione, saranno parzialmente incapsulate, un pezzo avviene in chiaro ed un pezzo più o meno lungo a seconda di quanto tempo questa comunicazione persiste, sarà incapsulato e di conseguenza anche il routing non sarà ottimale. Quanto non ottimale dipende da quanto velocemente il nodo si sposta e quanto dura la sessione di lavoro.

CN anchoring. Questa soluzione che abbiamo già discusso quando abbiamo parlato del protocollo Mobile IP, uno dei punti di debolezza del protocollo e del Proxy Mobile IP che si ispira al protocollo base. Era il problema dell'instradamento non ottimale causato dalla triangolazione, a causa della posizione del nodo mobile rispetto al suo HA. L'idea per ridurre l'impatto negativo della triangolazione è molto elementare: se voglio rendere efficiente questo scambio un modo per farlo è di ridurre al minimo possibile i lati del triangolo. Triangolazione c'è sì, ma uno dei lati è sostanzialmente inesistente. L'idea è che

lo HA non è un'entità fissa e statica ma è un'entità creata dinamicamente a seconda delle necessità. Viene creato nella stessa rete in cui si trova il nodo corrispondente del nodo mobile. Questo vuol dire che se un nodo mobile ha in contemporanea più sessioni di lavoro attive, avrà tanti HA di appoggio quanti sono i suoi corrispondenti. Dal punto di vista di CN, MN è un nodo che sta all'interno della sua stessa rete. Quello che nella realtà succede è che i pacchetti che CN indirizza verranno catturati dallo HA che conosce la posizione effettiva del nodo mobile, incapsula i pacchetti, si crea un tunnel e li fa arrivare fino a destinazione. Se il nodo mobile cambia posizione, dal punto di vista End-to-end non cambia nulla, continua ad avere l'indirizzo CoA assegnatogli in precedenza, ma cambia la situazione a livello di tunnel. XS2 è l'indirizzo assegnato al nodo mobile nella nuova rete in cui si è spostato, con un meccanismo analogo a quello visto in Mobile IP. Da ognuna delle reti in cui risiede un partner attivo parte un tunnel che si indirizza verso la posizione attualmente occupata dal nodo mobile. In questo caso il routing è sempre ottimizzato, ma sono tunnel totali in quanto per ogni comunicazione abbiamo la presenza di un tunnel, a prescindere dal movimento o no del nodo mobile. E' evidente la puntualizzazione fatta all'inizio. Riguardano solamente il mantenimento della continuità della sessione. Non si pongono il problema se è il nodo mobile a parlare con qualcuno, ma non si fanno carico del sistema inverso, cioè qualcuno che vuole parlare con il nodo mobile.

Host routing, l'idea è di avere un routing personalizzato. Ogni router possiede un entry che dice, nel momento in cui arriva un pacchetto indirizzato ad uno specifico nodo mobile, il prossimo hop, dove deve essere instradato. In questo caso non c'è incapsulamento, l'instradamento è sempre ottimale. C'è l'overhead dal punto di vista dei messaggi di controllo, in quanto l'informazione sulla posizione del nodo mobile e sui suoi cambi di posizione vanno propagate sulla totalità della rete, su tutta la porzione di rete che si intende mantenere aggiornata sulla posizione di nodo mobile. In questo caso questa soluzione risolve anche il problema della raggiungibilità del nodo mobile.

Giusto se vogliamo riassumere, fare un confronto, dal punto di vista dei cammini dei pacchetti del nodo mobile, nel caso di ancoraggio AR, prima che avvenga un cambio di posizione è ottimale, dopo sarà più o meno sub-ottimale a seconda di quanto ci si allontana dalla posizione originaria. Negli altri due casi è sempre ottimale. Quanti pacchetti vanno nel tunnel: il percorso è parzialmente dentro un tunnel nel caso di AR, tutti i pacchetti nel caso di CN e nessuno nel caso dell'host routing. La lunghezza del tunnel, nel primo caso è relativamente breve, dipende dal rapporto tra durata di una sessione e velocità di spostamento. Se questo rapporto è alto, quindi se la sessione dura a lungo ed il nodo si sposta velocemente, la lunghezza del tunnel cresce. Nel secondo caso invece dipende dalla distanza tra MN e CN. Nell'ultimo caso non abbiamo tunnel. Quanti accoppiamenti tra posizione attuale e posizione virtuale al nodo mobile vanno mantenute in giro per la rete: nel primo caso dipende dal rapporto della durata della sessione e durata del tempo di permanenza in una rete, nel secondo caso dipende dal rapporto tra durata di sessione e tasso di arrivo di nuove richieste di sessione, nell'ultimo caso invece è a macchia d'olio, tutta la rete è coinvolta. Dal punto di vista del traffico di segnalazione: nel primo caso avviene tra AR verso cui passano sessioni attive, nel secondo caso dipende da quanti sono il numero di reti diverse con cui il nodo mobile stabilisce contemporaneamente sessioni, nell'ultimo caso c'è il coinvolgimento totale della rete. Il numero di indirizzi IP che

identificano un nodo mobile: nel primo caso dipende da quanti sono gli AR attraverso cui passa una sessione attiva, nel secondo caso dal numero di corrispondenti e nell'ultimo caso ne abbiamo 1 fisso.

Queste tre soluzioni sono state oggetto di valutazioni in vari modi. Quello che brevemente presento è una valutazione che è stata fatta, simulazione con i parametri presenti sulle slide. Da vari esperimenti vengono valutati i costi di queste tre soluzioni in funzione del numero di nodi mobili che sono attivi nel sistema. In questa prima figura viene valutata quella che è il data cost, il traffico dati a livello applicativo, dove questo traffico dati viene valutato in termini sia di dimensione dei pacchetti scambiati, ma anche della lunghezza del percorso che viene seguita. La quantità di dati che viene scambiata viene moltiplicata per la lunghezza del percorso. Da questo punto di vista, rispetto a questa metrica delle tre tecniche considerate, quella che dà i risultati migliori è quella dell'host routing, proprio perché trae beneficio dal fatto che i pacchetti non sono incapsulati e seguono sempre un routing ottimale. Se si guarda però al traffico di controllo la situazione si inverte: host routing dà il costo maggiore, tra l'altro esplode al crescere del numero dei nodi. Gli altri si tengono più bassi, in particolare l'AR è quello che dà il costo minore. Se si guarda invece a qual'è l'impatto dal punto di vista della dimensione del sistema, quanto è ampia la rete e quanto grande è il numero degli AR coinvolti, di nuovo le soluzioni CN ed AR hanno costi bassi, mentre come è facilmente intuibile host routing di nuovo quanti più sono i routing coinvolti, quanto più in modo esponenziale cresce il costo.

Alla fine facendo variare il tempo tra due cambi di posizione, quello che vediamo è che per nodi che si muovono lentamente, quanto più statico è un nodo, il costo per il CN decresce, mentre l'inverso succede nel caso contrario. Le altre tecniche sono sostanzialmente indipendenti. Con questo chiudiamo questo capitolo ed apriamo quello che sarà l'ultimo prima della prova intermedia.

---

## Mobility management in Internet: no infrastructure networks

Ultimo capitolo, come dicevo, prima della prova intermedia. Rimaniamo nell'ambito di metodologie per risolvere il problema della mobilità e della raggiungibilità di un nodo mobile e/o del mantenimento di una sessione in corso. Parleremo ancora di soluzioni standardizzate dall'organismo IETF. In questo caso muoviamo in un contesto diverso da quello preso in considerazione fin'ora. Non assumiamo più che esista un'infrastruttura di appoggio, al di là dei nodi mobili interessati ad interagire, che possa offrire supporto per risolvere questo problema.

Dell'esistenza di questa infrastruttura ci abbiamo fatto affidamento fino ad ora. Cosa succede se ci mettiamo nell'ipotesi che questa infrastruttura non esista più. Non esiste più per molteplici ragioni, più o meno negative: un'infrastruttura può non esistere nativamente se mi trovo in un contesto nuovo, o poteva essere pre-esistente ma distrutta, oppure esiste ma ci possono essere motivazioni economiche o di privacy, diritti di accesso di varia natura, che suggeriscono di non usare un'infrastruttura esistente. Se non posso usare un'infrastruttura per realizzare l'instradamento, i nodi che vogliono interagire si devono

rimboccare le maniche e cercare di risolvere per loro conto il problema dello scambio di messaggi. Ovviamente questo può tornare utile in vari contesti, i complementari di quanto descritto un attimo fa. E' una soluzione che viene esplorata anche in ambito di reti cellulari per cercare di risolvere il problema causato dai picchi improvvisi di traffico che si possono generare all'interno di una cella. Se le comunicazioni riguardano nodi che si trovano nella stessa cella o celle vicini, i nodi potrebbero provare a parlarsi direttamente tra di loro bypassando la necessità di passare attraverso l'infrastruttura.

Piccola nota di inquadramento. L'argomento di cui parleremo è un caso particolare di una problematica più generale che potrebbe essere affrontata, ma che per questioni di spazio ed interesse non trattiamo. La situazione più generale è quella delle reti opportunistiche: è una rete in cui non esiste infrastruttura di appoggio per garantire comunicazione di un certo numero di nodi, ed inoltre se io fotografo il sistema in un certo istante, tipicamente non esistono cammini end-to-end, se prendo a caso due coppie qualunque di nodo, è altamente probabile che non esista un cammino completo ent-to-end tra questi due nodi. Quello che mi posso aspettare è che esistano frammenti di cammini, il grafo delle connessioni diventa un grafo incompleto. In queste situazioni se sono interessato a fare parlar de nodi qualunque di questo sistema, devo presupporre che i nodi siano mobili, almeno qualcuno, allora esiste una probabilità non nulla che due nodi si avvicinino uno con l'altro. Se sono interessati direttamente possono scambiarsi le informazioni, o se non sono interessati direttamente possono fare da punti di appoggio per comunicazioni partite da altri nodi. Ogni nodo quando passa vicino a qualcuno si carica di una parte dei dati in attesa di essere inoltrati e se li porta in giro fin quando non trova qualcuno interessato a quei dati.

Questo è lo scenario più generale possibile. La sigla MANET, mobile ad hoc network, si assume che sia altamente probabile che una connessione end-to-end esista. La configurazione della connessione può cambiare nel tempo, ma assumiamo che mediamente sia altamente probabile che ci sia. E' uno scenario in cui ci sono vincoli di banda e di energia, visto che parliamo tipicamente di nodi mobili alimentati da batterie, cercare di risparmiare energia il più possibile. Sono soluzioni intrinsecamente distribuite chiaramente, è difficile immaginare una gestione centralizzata. Non esistono punti di fallimento singoli.

Giusto per collocarla rispetto alle soluzioni visto fino ad ora possono essere riassunte in questo modo. Immaginiamo un'infrastruttura fissa formata da router, i quadratini arancioni che fanno da infrastruttura di supporto per le comunicazioni a cui possono essere collegati nodi terminali fissi o mobili. Nel nostro caso invece assumiamo che sia tutto quanto mobile. Tutto mobile a questo punto chi si fa carico dell'instradamento? In generale tutti i nodi oppure un sottoinsieme dei nodi interessati può farsi carico del compito di fare da punto di passaggio intermedio per gli scambi tra nodi che fanno parte del sistema. Ci muoviamo nel nuovo scenario.

Reti di questo tipo potrebbero essere, rispetto alla totalità di un'infrastruttura di internet, le potremmo vedere o come reti di transito o come stub network: reti di transito vuol dire che i punti finali di una comunicazione stanno fuori dalla rete MANET, questa fa solo da punto di passaggio delle informazioni. Oppure la rete MANET è il punto di partenza o di

terminazione della comunicazione. La prima è più complicata delle due, noi assumeremo di essere nella seconda situazione.

Caratteristica fondamentale è il suo dinamismo, la mancanza di una topologia fissa. Vuol dire che se io prendo una qualsiasi coppia di nodi ed osservo lo stato della connessione nel corso del tempo e la qualità della connessione, questa varia in maniera più o meno ampia. Varia sia la qualità nei due sensi, ma la variazione può essere anche drastica. La qualità può non essere la stessa nelle due direzioni ma potrebbe anche mancare del tutto. Il fatto che un nodo riesca a parlare con un altro non significa necessariamente l'inverso. Un meccanismo di instradamento per queste reti dovrebbe essere in grado di gestire situazioni di asimmetria nei link tra due nodi qualunque.

Il problema di come realizzare l'instradamento in una rete di nodi mobili è molto affrontato. Gli algoritmi di routing tradizionali hanno o tempi di convergenza lunghi, non sono adatti ad uno scenario molto dinamico come quello delle reti MANET, o anche sono inefficienti in quanto la quantità di messaggi scambiati è molto grande, o anche perché non sono in grado di gestire la presenza di link asimmetrici, il fatto che se A parla con B, non è detto che B riesca a parlare con A. Altra cosa, tipicamente questi algoritmi usando come metrica il numero hop scelti tra vari percorsi, che non è detto che sia la metrica più efficace per valutare la qualità di un algoritmo di instradamento.

Approcci generali, non gerarchici, in cui tutti i nodi del sistema sono alla pari, si fanno carico di dare il loro contributo all'inoltro dei messaggi. Approcci gerarchici dove invece si definisce una gerarchia tra i nodi del sistema, quindi alcuni alla base della gerarchia sono liberati dal carico, dall'incombenza di risolvere l'instradamento, e solo un sottoinsieme dei nodi ha questo compito. In questo caso c'è il problema aggiuntivo di come selezionare chi saranno i nodi che dovranno contribuire a risolvere il problema. E poi altra classificazione, una cosa ben nota, è una delle direzioni che abbiamo già esplorato quando parlavamo di tecniche di soluzioni al problema della mobilità, quindi possiamo distinguere approcci di tipo reattivo che entrano in gioco solo se c'è un'esigenza reale di comunicazione, da approcci di tipo pro attivo che invece cercano a priori di tenere aggiornate le informazioni su dove indirizzare i messaggi verso una certa destinazione. Qualunque soluzioni venga proposta va valutata la sua efficacia, che può essere valutata usando misure esterne, di interesse per un utilizzatore finale di questi meccanismi, quindi il ritardo che subiscono i messaggi, il throughput, quanti messaggi riesco ad inviare per unità di tempo. Altre misure sono quelle interne, chi è responsabile del sistema che prezzo deve pagare per garantire il tutto, dove il prezzo può essere valutato in termini di overhead, di sforzo da fare per garantire queste misure esterne. Ovviamente gli indici di prestazione sono dipendenti da una serie di parametri che caratterizzano il sistema: dimensione del sistema, qual'è il grado di connettività, ogni nodo con quanti riesce a parlare direttamente, il dinamismo della rete, quanto velocemente cambia la topologia, qual'è la banda disponibile, quanti link possono essere assunti come bidirezionali o meno, come sono i pattern di traffico, quindi se il traffico si disperde in maniera uniforme verso tutte le destinazioni possibili o invece si indirizza verso delle posizioni che è una frazione del totale dei nodi, dipende anche da quanti nodi mettono in atto politiche di risparmio energetico. Sono vari fattori.

Questo tipo di reti ha avuto un picco di interesse un po' di anni fa, per cui esiste una grande varietà di proposte. Ci concentreremo solamente su due protocolli. L'IETF si assume la responsabilità di un protocollo di tipo reattivo ed un protocollo di tipo pro attivo. Questi due protocolli sono il primo di tipo reattivo, **AODVv2**, che in realtà è una sintesi di due protocolli proposti in precedenza. Quello di cui discuteremo è in realtà *DSR* in particolare, che è quello che presenta caratteristiche più innovative su cui vale la pena soffermarsi. **OLSRv2** protocollo di tipo pro attivo, parleremo della versione 1.

Partiamo con **OLSR**. E' stato concepito diciotto anni fa circa. Appartiene alla famiglia degli approcci pro attivi. L'idea è quella di tenere, a prescindere dall'esigenza o meno di comunicare, informazione sulla posizione del nodo mobile, rischiando di tenere aggiornate informazioni che magari non verranno utilizzate. Questo problema dello sforzo inutile è particolarmente grave e sentito, in quanto il risparmio di energia è vitale nelle reti MANET. Come si cerca di ovviare a questo problema in questo algoritmo? Lo si fa agendo in due direzioni: cercando di ridurre la quantità di messaggi di controllo, di aggiornamento della posizione dei nodi, che vengono fatti circolare nella rete. Dall'altro cercando anche di ridurre la dimensione che hanno questi messaggi, in modo da minimizzare l'impatto del traffico di controllo sui consumi energetici del sistema. L'idea che sta dietro questi due obiettivi si basa sul concetto che adesso spieghiamo di *multipoint relay set*.

E' un concetto che si applica ad ogni nodo del sistema. Dato un nodo A, questo set è l'insieme minimo di vicini che stanno a distanza 1 da A, ed attraverso i cui A è in grado di inviare un messaggio a tutti i suoi vicini di distanza 2. Il minimo insieme di nodi che possono fare da intermediari per parlare con un nodo che sta a distanza 2 da A. Solo i nodi che fanno parte del multipoint relay set di qualche nodo, saranno quelli coinvolti nello scambio di messaggi. Quali messaggi? I messaggi di controllo sono di tipo: hello, mandati da tutti i nodi, ed ogni nodo in questo pacchetto mette la lista dei nodi che stanno a distanza 1 da lui. Di questi nodi specifica anche qual'è il sottoinsieme che costituisce il suo multipoint relay set. Se ogni pacchetto manda un pacchetto di hello concepito in questo modo, ogni nodo è in grado di costruirsi la mappa di quali sono i nodi a distanza 2 da lui. Se ascolto i messaggi di hello di tutti i miei vicini capisco quali sono a distanza due da me. A questo punto sono in grado di calcolare il mio multipoint relay set. Poi a questo punto altri pacchetti di topology control TC vengono inviati solo dai nodi che fanno parte del set *multipoint relay set* e contengono la lista di quali sono i nodi per cui il nodo mittente di questo pacchetto è membro del loro multipoint relay set. I nodi di questo tipo sono sostanzialmente i router del sistema.

Ogni nodo determina quali sono i nodi che stanno a distanza 1. Si calcola l'insieme dei nodi che si trovano a distanza 2. Si calcola l'MPR ed a questo punto costruisce il suo messaggio di hello, che viene periodicamente inviato. Se un nodo riceve un pacchetto di tipo hello, si chiede se fa parte del set MPR del nodo che mi ha inviato questo pacchetto. Se la risposta è no, non devo fare nulla. Se scopro di fare parte del set MPR di qualche nodo, in questo caso mi devo fare carico di inviare un pacchetto TC, perché scopro di essere stato eletto come un router per il nodo che mi ha inviato questo pacchetto di hello. Se un nodo qualunque riceve un pacchetto di tipo TC, si chiede se fa parte del set MPR di qualcuno, se la risposta è no non faccio nulla. Se la risposta è sì lo devo inoltrare a qualcun altro. In questo modo solo i nodi che fanno parte dell'insieme MPR ricevono questi aggiornamenti sulla topologia della

rete. Tutti i nodi MPR acquisiscono l'informazione generale sulla topologia della rete. Alla fine i messaggi di controllo viaggiano solamente tra i nodi MPR e riguardano l'informazione su solo come raggiungere uno di questi nodi. Viene ridotta la quantità di messaggi che circolano ed anche la loro dimensione.

Continuiamo con **DSR**. E' un protocollo di tipo reattivo, quindi viene attivato soltanto se qualcuno vuole parlare con qualcun altro, ha un messaggio da inviare verso la destinazione. Si basa su due algoritmi: uno di scoperta del cammino, nel momento in cui dal momento applicativo arriva un messaggio destinato verso una certa destinazione. A questa funzione di scoperta si affianca una funzione di manutenzione, il cui senso è quello di non disperdere lo sforzo fatto per la scoperta dei cammini, volta per volta. L'informazione raccolta sulla raggiungibilità dei nodi viene raccolta dal sistema. Essendo un sistema dinamico quest'informazione è soggetta ad obsolescenza, quindi è necessario fare un lavoro di manutenzione e verificare fino a che punto l'informazione raccolta nel passato più o meno recente sia ancora effettiva o va in qualche modo aggiornata.

Cosa ha di particolare questo algoritmo di instradamento? L'aspetto particolare è che non c'è in nessun punto della rete una tabella di instradamento. Ogni pacchetto che viaggia all'interno della rete ha scritto dentro di sé qual'è il percorso che il pacchetto deve seguire. Questo è il concetto di source routing. Chi invia il pacchetto oltre a costruire il carico utile, nello header del pacchetto metterà oltre al destinatario finale anche tutti i passaggi intermedi attraverso cui quel pacchetto deve passare per arrivare alla destinazione finale. Il concetto è un po' diverso.

Per mettere in atto questo meccanismo questo algoritmo fa affidamento su alcune strutture dati. Mantiene una cache di percorsi già scoperti in precedenza verso un certo numero di destinazioni, con le tipiche operazioni di una cache. Avrò una tabella di richieste che sono in corso di risoluzione, quindi quali sono i messaggi di route request che sono stati generati dal nodo perché qualcuno a livello applicativo ha prodotto un messaggio che deve essere inviato verso una certa destinazione, o messaggi ricevuti da altri ed inoltrati dal nodo in questione. Poi un buffer di messaggi che sono in attesa di essere trasmessi, perché è in corso di risoluzione il processo di scoperta del cammino finale. Ed il buffer di ritrasmissione, pacchetti in attesa di ricezione dell'ACK di ritorno.

Date queste strutture dati, come avviene la funzione di scoperta di un cammino? In prima battuta, tempo 0, se un nodo deve inviare un pacchetto verso qualcuno che non sta a distanza 1, invia un messaggio di broadcast di tipo *ROUTE REQUEST*, dove è specificato chi è il destinatario finale del messaggio. Questo pacchetto ha un identificativo che lo rende univocamente identificabile all'interno di tutto il sistema. Questi messaggi inviati in broadcast iniziano a propagarsi nel sistema ed arrivano ad un nodo. Ci sono tre possibilità. La prima, il nodo che riceve il pacchetto si riconosce come destinatario finale, ma ne parliamo per ultimo. Seconda possibilità, il pacchetto che io ricevo ha un ID che mi consente di rivelare consultando la tabella dei messaggi di route request che ho già gestito, consente di capire se ho già ricevuto in precedenza lo stesso messaggio di route request. In questo caso viene cestinato per evitare la creazione di cicli. Altrimenti prendo il pacchetto, appendo nell'header del pacchetto il mio identificativo, e mando il broadcast il messaggio. A questo punto, questo messaggio inizia a diffondersi a macchia d'olio nella rete evitando

cicli, e prima o poi questo messaggio arriverà ad un nodo che si riconosce come destinazione finale. Il messaggio conterrà, grazie al meccanismo precedente, l'elenco di tutti i nodi che sono stati attraversati prima di arrivare a destinazione finale. Porta scritto in se qual'è la strada, se A è il nodo che ha mandato il messaggio originariamente e B quello finale, qual'è il percorso che connette A e B. Il nodo B invia un messaggio di *ROUTE REPLY*. Se facciamo l'ipotesi che i link siano bidirezionali, la strada che il messaggio deve seguire è l'inverso della strada che sta scritta nel pacchetto di *ROUTE REQUEST* arrivato a B. Questo è vero solo nell'ipotesi che tutti i link siano bidirezionali. Se non faccio questa ipotesi, adesso questo diventa carico utile e devo riavviare in senso inverso un meccanismo di *ROUTE REQUEST* per fare arrivare il messaggio a destinazione. Il destinatario finale può ricevere più messaggi di *ROUTE REQUEST* che hanno seguito strade diverse per arrivare a lui. La situazione più è che il mittente può decidere cosa fare, se accontentarsi del primo messaggio che gli arriva e lo usa per inoltrare il messaggio in attesa nel buffer dei messaggi di send, oppure aspetta un po per ricevere più messaggi di reply e magari scegliere il migliore tra i vari percorsi che gli sono stati indicati.

L'idea di base è semplice, ma perché sia anche efficacemente utilizzabile va adottata un qualunque accortezza. L'idea è quella di, a parte limitare il numero massimo di passi, conoscendo il diametro della rete, è fare tesoro delle esperienze precedenti. Al tempo 0 nessuno sa nulla di niente quindi non c'è alternativa ad una diffusione a macchia d'olio. Man mano che il sistema va avanti l'informazione comincia a circolare e ad essere acquisita, che oltre ai punti estremi viene acquisita anche come informazione parziale da tutti i nodi coinvolti in questo scambio. Nel momento in cui mi arriva nel futuro più o meno prossimo una richiesta per andare da un punto ad un altro della rete, può essere che io abbia già quell'informazione, senza ricostruirla da 0. Il funzionamento a regime diventa sempre più efficiente, non c'è più necessità di diffondere messaggi a macchia d'olio ma l'informazione diventa sempre più probabile che sia scritta da qualche parte. Se per esempio A vuole parlare con B ed un nodo intermedio riceve il messaggio di route request che da se stesso porta a B, piuttosto che seguire il meccanismo standard di inoltre, può agire generosamente e fa lui *ROUTE REPLY*, velocizzando la latenza della scoperta.

**Manutenzione della route.** Questo meccanismo serve per definire il percorso da seguire. Il mittente originario che ha ricevuto informazione sul percorso che deve essere seguito per arrivare a destinazione prende il pacchetto che stava in attesa, aggancia un header che contiene tutta la strada da seguire ed inoltra il pacchetto che deve arrivare a destinazione. Però può essere che quella configurazione sia diventata non più valida. Ogni nodo che fa l'inoltro di un pacchetto con le informazioni scritte nel pacchetto stesso deve verificare se quelle informazioni sono valide o se vanno aggiornate. La validità delle informazioni può essere valutata in vari modi: per esempio, guardando se dal livello sottostante arriva un ACK se lo prevede il protocollo, oppure se stiamo usando un protocollo link layer che non prevede ACK posso ascoltare il messaggio che ho inviato e se vedo se viene rimbalzato dal mio vicino a distanza 1, oppure se i protocolli non sono affidabili e non mi fido dell'ascoltare o meno, posso prevedere la richiesta di un ACK dal nodo a cui ho indirizzato. A questo punto a causa di un qualunque di questi motivi, chi ha inoltrato il pacchetto capisce che c'è un problema, posso o informare il mittente originario per non indicare la validità del percorso indicato nel pacchetto, oppure mi faccio carico in prima persona di



attivare localmente la procedura di riparazione, individuazione del nuovo percorso da seguire.

Si tratta di un protocollo semplice ma efficace, la cui efficacia dipende da vari fattori tra i quali gioca un ruolo preponderante il livello di dinamismo della rete. Se la topologia cambia lentamente io riesco a capitalizzare molto, in maniera notevole, lo sforzo fatto in precedenza e quindi l'informazione che tengo nella mia cache saranno validi per intervalli di tempo lunghi e l'efficienza del protocollo diventa notevole. Se la rete è molto dinamica il tipo di protocollo cambia, rischia di diventare un po' troppo oneroso.

Piccolo schema riassuntivo preso da questo libro di qualche anno fa. I due algoritmi sono uno di tipo reattivo e l'altro di tipo pro attivo. In quali scenari lavorano meglio? Questa è una caratterizzazione che possiamo provare a fare, Da un lato mettiamo in ascissa il livello di dinamismo del traffico, dove per dinamismo intendo la porzione di nodi che sono coinvolti in una comunicazione. Il dinamismo è elevato se ogni nodo ha molti partner. L'altro è la sua imprevedibilità, la probabilità di parlare con un nodo piuttosto che con un altro è uniforme, non c'è una probabilità concentrata su pochi nodi. E' difficilmente predicibile con cui un nodo parlerò. Una rete con un alto dinamismo ha poca predicibilità, ampio set di partner. Dal punto di vista del dinamismo della rete intendo invece quanto rapidamente cambia la topologia del sistema, quindi la mobilità dei nodi e la non validità di link del corso del tempo. Questo schema dice che reti che hanno un dinamismo elevato dal punto di vista della rete, ma poco dinamismo dal punto di vista del traffico di controllo, algoritmi reattivi funzionano mediamente meglio rispetto a quelli pro attivi. Il viceversa funzionano meglio gli algoritmi pro attivi. Nelle situazioni intermedie si può immaginare che soluzioni ibride possano risultare più vantaggiose.

Gli algoritmi di cui abbiamo parlato sono sostanzialmente piatti. L'idea dei metodi gerarchici è quello appunto di aumentare l'efficacia di questi protocolli in situazioni e scenari in cui il numero dei nodi comincia a diventare importante. Se i nodi sono tanti, algoritmi piatti rischiano di generare traffico di controllo eccessivo. L'idea è quella di avere un'insieme di nodi di questo tipo, che posso immaginare di ripartire in sottoinsiemi tra cui eleggo un nodo particolare come rappresentante, punto di accesso ad ogni sottoinsieme, per cui l'instradamento viaggia in prima battuta soltanto tra questi nodi e successivamente si propaga all'interno della rete. Questa rete può essere fatta a più livelli, potendo creare dei super cluster. Il problema di fondo è in base a quale criterio fare la ripartizione. E si possono adottare criteri differenti: uno può essere quello di massimizzare il livello di connettività fisica all'interno di ogni sottoinsieme, caso limite sotto insieme con tutti i nodi a distanza uno tra di loro. Oppure potrebbe essere quello di connettività logica, mettendo assieme nello stesso cluster i nodi che interagiscono con maggiore probabilità tra di loro. Un altro criterio che magari si accoppia ad uno dei due precedenti è quello di fare in modo che ogni cluster ci sia almeno un nodo che possa svolgere il ruolo di punto di accesso, sperabilmente un nodo con disponibilità di energia illimitata o capacità computazionali superiori rispetto agli altri. Suddivisi i nodi, individuali i cluster, all'interno va eletto il rappresentante. Questo può essere il criterio di nodo più performante, il criterio fondamentale.

Ultimo argomento e poi chiudiamo, metriche. La metrica ovvia, per valutare la bontà di un cammino, è contare il numero di hop ma non è detto che sia la metrica più vantaggiosa. Immaginate questo scenario: il nodo S1 voglia parlare con R1, il nodo S2 voglia parlare con R2. Se guardiamo i percorsi possibili da S1 ed S2 verso le rispettive destinazioni, sono le linee continue o tratteggiate. Se usiamo come metrica il numero di hop, da questo punto di vista i due percorsi sono equivalenti, stessa distanza, per quanto riguarda la linea continua, mentre differenti per la linea tratteggiata. Se guardiamo invece alle cose con un'ottica diversa vediamo che questi percorsi non sono equivalenti e non è detto che siano la scelta migliore da fare, soprattutto se fatta in contemporanea. Visto che alcuni nodi, in giallo, indicano il fatto che dal punto di vista di comunicazione wireless sono nodi che sono in grado di disturbarsi a vicenda, dunque le loro comunicazioni wireless si possono sovrapporre. Ognuno ascolta, è coperto dal segnale radio inviato dall'altro. Può succedere che condividendo lo stesso mezzo wireless, se questi nodi inoltrano messaggi in parallelo si disturbano uno con l'altro. Il throughput di questi percorsi, se seguiti in parallelo, può essere notevolmente più basso di quello che nominalmente ci potremmo aspettare. Da questo punto di vista i percorsi tratteggiati sarebbero quelli da preferire in quanto ci danno la minore interferenza possibile. Una metrica alternativa molto semplice che cerca di privilegiare questo tipo di argomento rispetto alla lunghezza di per sé del percorso è la metrica di percorso con la minima interferenza, dove si valuta l'interferenza potenziale. Conto per ogni tappa da quanti nodi può essere disturbato potenzialmente, ed a questo punto scelgo il cammino che ha potenzialmente il minimo numero di disturbatori possibili. Sono disturbatori potenziali in quanto non è detto che il disturbatore metta in atto il suo disturbo, ma mi metto nell'ipotesi di caso peggiore e scelgo il percorso che mi garantisce il minimo caso peggiore. E' una tecnica molto facile da valutare in quanto si basa su informazioni locali calcolabili da ogni elemento del percorso. Ci sono metriche più sofisticate e precise, ma più difficili da valutare.