Università degli Studi di Pavia
Facoltà di ingegneria

Corso di Laurea in Ingegneria Informatica
Sede di Mantova

Analisi delle problematiche nelle reti multicast ipv4

Relatore:
prof. Giuseppe f. Rossi

Tesi di laurea di:
Andrea Bianchi
Matricola: 302039/46

Anno Accademico 2005/2006

Indice

Capitolo 1

Introduzione

"Non dar retta ai tuoi occhi, e non
credere a quello che vedi.
Gli occhi vedono solo ciò' che è
limitato. Guarda col tuo intelletto, e
scopri quello che conoci già',
allora imparerai come si vola.".

Il gabbiano Jonathan Livingston
Richard Bach

Il grande sviluppo tecnologico avvenuto negli ultimi anni nella produzione di calcolatori e nello sviluppo di infrastrutture di rete informatiche ha permesso la nascita di nuove soluzioni software in grado di venire sempre più in contro alle esigenze degli utenti. Il campo multimediale è stato uno di quelli che ha ricevuto una spinta maggiore da questo sviluppo, infatti è sempre più alla portata di tutti avendo la possibilità di collegarsi ad Internet, poter ascoltare trasmissioni radiofoniche, vedere file video, partecipare a videoconferenze, tutto questo attraverso la rete. Il motivo per cui queste soluzioni non sono state adottate in passato è legato al fatto che esse richiedono un ingente consumo di risorse computazionali, di memoria e di banda, tutti requisiti che qualche anno fa non potevano essere alla portata di tutti. Ora che invece si dispone di tali potenzialità, bisogna trovare il modo per coordinare le risorse ed ottimizzare il lavoro al fine di ottenere servizi sicuri e veloci, attraverso la comunicazione multicast. I servizi IP multicast sono stati introdotti su Internet a partire dal 1991. Risale infatti a quella data l'attivazione di un insieme di tunnel IP fra macchine (dette mrouter) abilitate a svolgere le funzioni di multicast router. Questa struttura, nota come Mbone, ha permesso l'uso di applicazioni multicast da parte degli elaboratori situati sulle sottoreti adiacenti ad un mrouter. Nel 1993 è stato attivato il primo mrouter collegato a Mbone sulla rete GARR. Da allora la presenza di Mbone in Italia ha avuto una lenta ma costante crescita. Attualmente l'infrastruttura Mbone italiana comprende circa 40 mrouter. Nell'ultimo paio di anni Mbone è stata utilizzata per le sperimentazioni di IP multicast ed ha permesso lo sviluppo e il testing di protocolli di routing, protocolli di trasporto e programmi applicativi. Molte reti della ricerca e alcuni provider commerciali hanno già iniziato ad abilitare al multicast le loro reti, utilizzando soluzioni tecniche assai diverse da quelle inizialmente adottate in Mbone.
In questo lavoro ci proponiamo di esaminare il problema a livello introduttivo: Vogliamo fare il punto sulla comunicazione multicast, mettendo in evidenza i concetti fondamentali che la differenziano da quella unicast e successivamente addentrarsi per

studiare gli algoritmi ed i protocolli impiegati per ottenere una comunicazione di questo tipo. Vogliamo in pratica osservare come tali tecnologie lavorino effettivamente tramite l'analisi del traffico su una rete realizzata nel laboratorio di reti dell'Università di Pavia, presso la sede di Mantova concentrandoci sul protocollo di routing DVMRP.
A questo proposito, il lavoro è stato suddiviso nelle seguenti parti:

- il **Capitolo 2** si occupa di esporre i concetti base della comunicazione multicast confrontandoli con quelli della comunicazione unicast; verranno poi presi in esame gli indirizzi di tipo multicast per poi finire con una panoramica sulle applicazioni che si basano su tale servizio.

- il **Capitolo 3** si concentra sull' analisi del protocollo IGMP (***I**nternet **G**roup **M**anagement **P**rotocol*), ovvero a cosa serve e come si integra con la comunicazione di tipo multicast, mostrando esempi pratici di applicazioni di tale protocollo.

- il **Capitolo 4** illustra i principali protocolli di routing multicast soffermandosi principalmente sul DVMRP.

- il **Capitolo 5** si occupa di mostrare i risultati delle prove tecniche svolte in laboratorio e chiarire tutti i concetti visti nei capitoli precedenti. Si vuole testare l'effettiva efficacia della comunicazione multicast basata sul protocollo DVMRP.

- il **Capitolo 6** infine illustra le conclusioni del lavoro svolto.

La speranza è che questo elaborato risulti di utile ispirazione a chiunque volesse cercare chiarezza riguardo ai concetti inerenti alla comunicazione multicast ed a chi volesse intraprendere studi paralleli a quello trattato, usando questo come struttura portante.

Capitolo 2

Multicast

> “La disumanità del computer sta nel fatto che, una volta programmato e messo in funzione, si comporta in maniera perfettamente onesta”
>
> *Isaac Asimov*

In questo capitolo verranno esposti i concetti fondamentali della trasmissione di tipo Multicast.

In particolare verranno sviluppati i seguenti punti:

- Cosa si intende per trasmissione multicast
- Differenze fondamentali tra trasmissione unicast e multicast
- Indirizzi di classe multicast
- Impiego di tale tecnologia nelle applicazioni

2.1 Il concetto di trasmissione Multicast

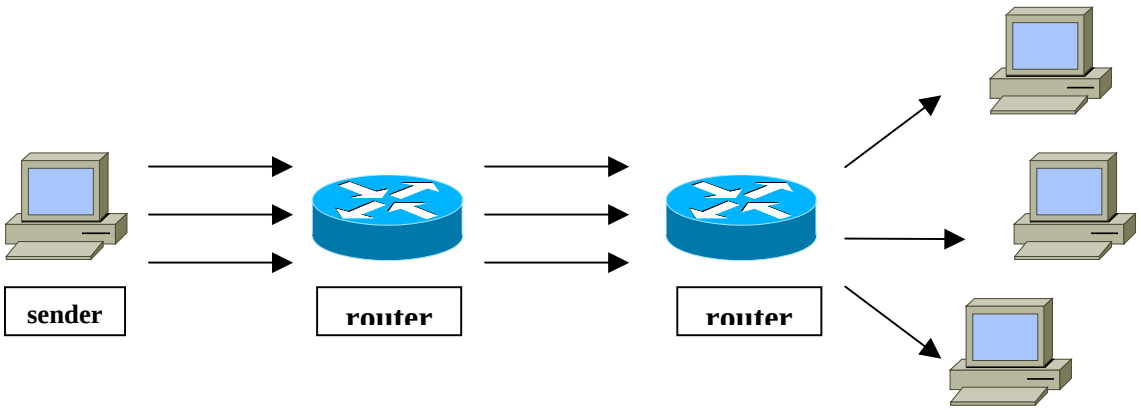
Con il termine *Multicast*, nelle reti di calcolatori, si indica la distribuzione simultanea di informazione verso un gruppo di destinazione.
Il termine viene utilizzato anche per indicare un pacchetto inviato con tale modalità. Un indirizzo che si riferisce a un gruppo di destinazione è detto a sua volta indirizzo multicast.
In alternativa, un pacchetto destinato a tutti i calcolatori di una rete è detto Broadcast, uno destinato ad uno qualunque di un gruppo anycast, uno destinato ad un solo calcolatore è unicast.
Il modello di servizio multicast prevede che un calcolatore invii i pacchetti ad un indirizzo associato al gruppo multicast; il calcolatore sorgente invia una sola copia dell'informazione (indipendentemente dal numero di destinatari), saranno poi gli Mrouter (Multicast Router) che moltiplicheranno l'informazione quando necessario. In questo modo se 50 computer (Gruppo) devono ricevere lo stesso file dalla stessa sorgente, quest'ultima invierà una sola copia del file, man mano che si naviga nella rete saranno gli Mrouter che moltiplicheranno le informazioni fino al raggiungimento dei 50 computer. I computer che vogliono ricevere le "trasmissioni" del gruppo multicast si devono registrare per quel gruppo con qualche meccanismo, e la rete si occuperà di consegnare i pacchetti multicast a tutti quelli che si sono registrati.

Il servizio di multicast è stato pensato per permettere la diffusione efficiente di programmi multimediali su una rete di calcolatori, in analogia con la radio e la televisione trasmesse nell'etere, e viene anche utilizzato per funzioni di gestione della rete (per risolvere problemi come "trova tutti i computer su una sottorete che implementano la funzione X o che hanno bisogno della funzione Y", basta che questi si iscrivano tutti ad uno stesso gruppo multicast).
Per la natura del servizio di rete multicast, risulta molto difficile usare protocolli di trasporto orientati alla connessione come TCP, per cui si usano protocolli senza connessione come UDP.
Il Multicast è implementato in ethernet in modo abbastanza semplice: una classe di indirizzi ethernet è riservata all'uso come indirizzi multicast. Questi pacchetti sono trattati dalla rete come se fossero broadcast, ovvero sono ritrasmessi a tutti i computer collegati. Se un processo è interessato a ricevere la trasmissione su un gruppo multicast, il sistema operativo lo comunica alla scheda di rete, che riceve il pacchetto e lo passa al sistema operativo, il quale a sua volta lo passa al processo interessato. Naturalmente questo sistema non è scalabile, in quanto tutto il traffico multicast viene fisicamente inviato a tutti i computer collegati alla rete, rischiando di saturare tutta la banda disponibile.
In Internet, il servizio multicast è implementato solo parzialmente, e in modo molto più complesso, perché la funzionalità di routing multicast deve essere aggiunta a tutti i router. Il protocollo IGMP viene usato dai computer per richiedere di essere iscritti ad un gruppo multicast,; esistono appositi algoritmi di routing per il traffico multicast, come DVMRP e PIM. Come per il traffico unicast, ci si appoggia ad eventuali meccanismi di multicast forniti dalle reti utilizzate.

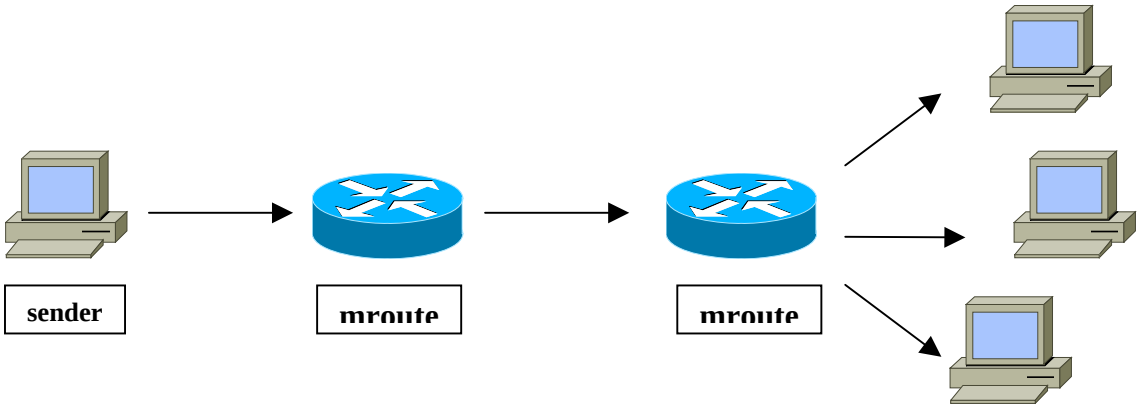
2.2 Differenze tra trasmissione unicast e multicast

Con unicast e multicast si indicano nel mondo Internet due distinte modalità di trasmissione dati su rete IP. Tutti i sistemi di invio delle informazioni secondo protocollo TCP/IP si basano in linea di massima sulla trasmissione di dati dal server ad un indirizzo composto dalla coppia IP/porta. Le differenze fra trasmissione unicast e multicast consistono fondamentalmente nel modo di operare dei server. Infatti, mentre nel sistema unicast il server distribuisce in rete pacchetti di informazioni distinti (ad es. flussi streaming) al singolo client, nel multicast il flusso dati viene inviato in rete una volta sola, ed è a disposizione di qualsiasi utenza sia in grado di riceverlo e ritrasmetterlo.

TRASMISSIONE UNICAST



TRASMISSIONE MULTICAST



La tecnologia Multicast, è un processo che trasmette le informazioni da una sorgente a più destinazioni con un unico flusso di dati, invece di usarne molteplici

In un certo senso l'**Unicast** può essere definito come un protocollo "on demand", perché l'attivazione di ogni connessione client-server avviene sempre in base ad una richiesta esplicita da parte del client, mentre il **Multicast** non è vincolato alle richieste del client.

Con la crescita di Internet in termini di utenti collegati e di servizi offerti, capita sempre più spesso che un certo numero di utenti voglia accedere contemporaneamente alle stesse informazioni. L'utilizzo del multicast in questi casi ha alcuni significativi vantaggi:

- **Banda**

Un qualsiasi flusso di dati fruito contemporaneamente da più utenti provoca un'occupazione di banda sui link di collegamento fra gli host, che cresce linearmente col numero di utenti nel caso unicast, mentre rimane costante nel caso multicast. E' evidente il vantaggio che si ottiene, dal fatto che a parità di banda disponibile, non si trasmettono più flussi identici ma uno solo con contenuto informativo maggiore.

- **Carico dei server**

Al crescere del numero di utenti collegati contemporaneamente, ogni server dovrà gestire un solo flusso in uscita, lasciando così libere risorse computazionali per altri processi. Nel caso unicast, al crescere degli utenti collegati, cresce la richiesta di CPU e di banda sull'interfaccia di rete, portando potenzialmente al collasso il sistema, con conseguente deterioramento delle prestazioni del servizio fornito.

- **Carico della rete**

Gli apparati di rete risentono chiaramente positivamente della riduzione del consumo di banda e del numero di flussi da gestire ottenuta tramite il multicast.
Tuttavia il carico di ridistribuire i contenuti forniti ricade generalmente proprio sui router, che nelle implementazioni più recenti del software permettono anziché di duplicare i pacchetti sulle interfacce di uscita, di scambiare solo puntatori ad aree di memoria condivisa. In tal modo ogni interfaccia accede allo stesso buffer di dati, riducendo così il consumo di CPU necessario ad inoltrare il pacchetto multicast.

Esistono tuttavia alcune limitazioni che bisogna tener presenti nel rilascio del servizio:

- **Inoltro non garantito dei pacchetti**

Allo stesso modo dell'IP unicast, il multicast è intrinsecamente inaffidabile. I meccanismi di garanzia dell'inoltro che nell'unicast vengono ottenuti a livello 4 o superiore con protocolli affidabili (TCP), non sono presenti nel multicast, che tipicamente utilizza UDP per il trasporto. Il multicast risente perciò del problema della possibile perdita di pacchetti, soprattutto in concomitanza con variazioni topologiche della rete che causino rerouting. Infatti il routing multicast e' dotato di meccanismi intrinseci per evitare i routing loop, ed in tali situazioni i pacchetti vengono scartati.

- **Duplicazione dei pacchetti**

Il trasporto UDP dei pacchetti porta inevitabilmente anche la possibilità che ci sia consegna di pacchetti duplicati alla destinazione. Il meccanismo di duplicazione intrinseco al funzionamento del multicasting può causare che (soprattutto in caso di ridondanza dei path fra sorgente e destinazione) finchè i protocolli di routing multicast non convergono, ci sia duplicazione alla destinazione.

- **Congestione della rete**

UDP non ha meccanismi di controllo della congestione. Il motivo della scelta di UDP è che per fornire contenuti real-time, i meccanismi di ritrasmissione di TCP sono di scarsa utilità, in quanto al momento in cui un pacchetto ritrasmesso arriva, il suo contenuto informativo può essere già vecchio. Perciò si preferisce inondare la rete di pacchetti via UDP nel modo più veloce possibile, senza aspettare che la finestra di trasmissione TCP

cresca o si adatti alla banda disponibile. Ciò significa anche che link sottodimensionati rispetto al flusso di informazioni vengano saturati senza scampo, a meno di prendere misure preventive di multicast rate-limiting.

PRO	**CONTRO**
Larghezza di banda	Inaffidabilità della consegna
Carico del server	Duplicazione dei pacchetti
Carico della rete	Congestioni di rete

Aspetti positivi e negativi della trasmissione multicast

2.3 Gli indirizzi Multicast

Come noto la gamma degli indirizzi ip si divide in classi. L'appartenenza a tali classi si può riconoscere guardando la parte più significativa dei 32 bit che compongono l'indirizzo:

gamma indirizzi	Parte alta indirizzo					bit 32
0.0.0.0 – 127.255.255.255	1					**INDIRIZZI DI CLASSE A**
128.0.0.0 – 191.255.255.255	1	0				**INDIRIZZI DI CLASSE B**
192.0.0.0 – 223.255.255.255	1	1	0			**INDIRIZZI DI CLASSE C**
224.0.0.0 – 239.255.255.255	1	1	1	0		**INDIRIZZI MULTICAST CLASSE D**
240.0.0.0 – 247.255.255.255	1	1	1	1	0	**RISERVATI**

Gli indirizzi che interessano a noi sono quelli di classe D, ovvero quelli che iniziano con la serie di bit corrispondente a "11110". I rimanenti 28 bit identificano il gruppo verso il cui il datagramma multicast deve essere spedito. Per fare un' analogia, quando si vuole ascoltare una determinata stazione radio, ci si deve sintonizzare su una specifica frequenza; allo stesso modo, se si vogliono ricevere pacchetti dati da un determinato gruppo multicast devo mettermi in "ascolto" sull' indirizzo che lo identifica.

Esistono particolari gruppi di multicast riservati, che non possono essere utilizzati nelle normali applicazioni in quanto il loro scopo differisce dai normali intenti di una trasmissione multicast:

224.0.0.1 – corrisponde a “tutti gli host del gruppo”: eseguendo un ping a questo indirizzo, tutti gli host facenti parte del gruppo di multicast rispondono alla richiesta.

224.0.0.2 – corrispondo a “tutti i router multicast del gruppo”: tutti gli mrouter devono eseguire il join su questo gruppo per ogni interfaccia attiva che possiedono.

224.0.0.4 – corrisponde a tutti i DVMRP router, 224.0.0.5 – tutti gli OSPF router, 224.0.0.13 – tutti i PIM router, ecc..

In ogni caso il range di indirizzi che va da 224.0.0.0 a 224.0.0.255 è riservato per scopi locali, amministrativi e di manutenzione. I pacchetti destinati a tali indirizzi non vengono inoltrati dagli mrouter.
Qui di seguito sono riportati in tabella i principali indirizzi multicast riservati:

224.0.0.0	Base address
224.0.0.1	All System on thi subnet
224.0.0.2	All routers on this subnet
224.0.0.3	Unassigned
224.0.0.4	All DVMRP routers on this subnet
224.0.0.5	OSPFIGP all routers
224.0.0.6	OSPFIGP designated routers
224.0.0.7	ST routers
224.0.0.8	ST hosts
224.0.0.9	RIP2 routers
224.0.0.10	Mobile – agents
224.0.0.11	DHCP server / relay agent
224.0.0.12	All PIM routers
224.0.0.13	RSVP Encapsulation
224.0.0.14	All CBT routers
224.0.0.15	Designated Sbm
224.0.0.16	All Sbms
224.0.0.17	VRRP

2.4 Applicazioni basate sul servizio Multicast

In questo paragrafo si vuole dare una panoramica sulle potenzialità che il servizio multicast può offrire, elencando alcune delle principali applicazioni che implementano tale servizio.

• **Tecnologie Streaming**

Alla base delle tecnologie di streaming c'è un concetto: comprimere e dividere i file multimediali in tanti piccoli pacchetti, così da permettere la visualizzazione e l'ascolto continuo del contenuto mentre viene ricevuto, senza doverne aspettare il download completo e senza la necessità di un elevato spazio sull'hard-disk del PC. Basta collegarsi al sito Web per assistere ad una trasmissione in diretta oppure on-demand fin quando si resta collegati ad Internet. Le nuove applicazioni soprattutto multimediali, il diffondersi di applicazioni di distribuzione audio e video, di audio-video conferenza, di formazione a distanza, di diffusione di dati a gruppi di utenti, su vasta scala, spinge ad utilizzare la trasmissione dati multicast in cui la sorgente di un flusso informativo con molti destinatari, emette pacchetti IP con indirizzo di gruppo (*multicast*). Tali pacchetti vengono recapitati a tutti i destinatari attraverso i meccanismi di instradamento forniti dai router multicast che fanno sì che solo una copia di ogni pacchetto passi su ogni linea, duplicando i pacchetti solo sui router in cui i percorsi divergono.
La trasmissione in Multicast è nata dall'applicare alla rete quello che è il metodo utilizzato dalla televisione tradizionale. Il media server trasmette un segnale che può essere ricevuto da n utenti. Anche in questo caso gli utenti potranno essere infiniti, ma dovranno far parte di una certa "isola multicast", ossia di una certa porzione di rete appositamente configurata per lasciar passare i pacchetti contenenti la trasmissione. La banda passante occupata dalla trasmissione è, al contrario di quanto succede per l' unicast, fissa. Attraverso la rete configurata per il multicast passa un flusso che è costante a prescindere dal numero di utenti collegati. La quantità di banda passante è pari al bitrate del contenuto audio/video oggetto dello streaming.

• **Teleconferenze**

Il flusso informativo emesso da un terminale è ricevuto da tutti gli altri membri di un gruppo. La teleconferenza consente di trasmettere e ricevere in tempo reale immagini e suoni anche tra luoghi fisicamente distanti per organizzare convegni in modo da poter coinvolgere il maggior numero di interessati .
La teleconferenza è inoltre utilizzata come risorsa per chi vuole fare istruzione; all'interno dell'ambiente didattico facilita le attività di cooperazione tra gli studenti e i docenti. Una caratteristica di gestione delle conferenze e' che il gruppo di partecipanti e' in numero variabile, e questa proprieta' e' molto piu' agevole da gestire in modalita' multicast.

• **Database**

Le copie di uno stesso file contenute in uno stesso database sono aggiornate contemporaneamente.

- **Distributed computing**

I risultati di un elaborazione sono trasferiti verso altri computer facenti parte dello stesso gruppo di multicast.

- **Resource discovery**

Individuazione della topologia di una rete da parte di un router. Questo processo serve al router per individuare i propri vicini e in seguito utilizzare le informazioni raccolte per l'inoltro dei pacchetti.

Capitolo 3

Internet Group Management Protocol (IGMP)

> “Tutti sanno che una cosa è impossibile da realizzare, finché arriva uno sprovveduto che non lo sa e la inventa”.
>
> *Albert Einstein*

In questo capitolo verrà descritto il funzionamento del protocollo IGMP, ovvero lo “strumento” che un router multicast (*mrouter*) utilizza per essere a conoscenza della presenza di host che entrano a far parte di un gruppo (*join*), e viceversa per sapere quando un host appartenente ad un gruppo lascia questo (*leave*). Successivamente queste informazioni saranno utilizzate per massimizzare l’efficienza dell’inoltro dei pacchetti sulla rete.
Verranno sviluppati i seguenti punti:

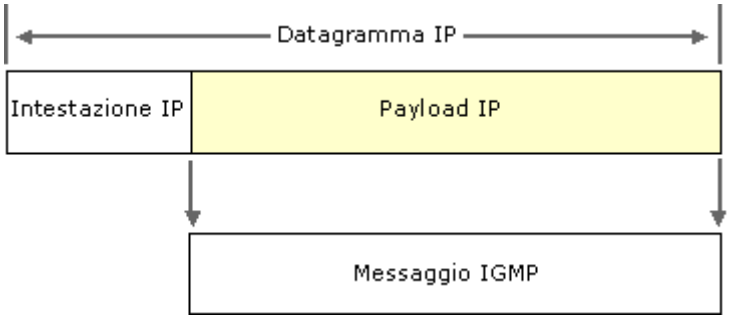
- Introduzione al protocollo IGMP
- Formato dei messaggi IGMP
- Funzionamento
- Esempio procedura di JOIN
- Esempio procedura di LEAVE
- Diagrammi di stato

3.1 Introduzione al protocollo IGMP

Per poter operare correttamente, una rete multicast deve "conoscere" quali siano gli host (o gruppi di host) interessati a ricevere un determinato flusso. Queste informazioni sono raccolte dai router multicast che colloquiano con gli host utilizzando un protocollo chiamato IGMP (Internet Group Management Protocol).
Questo protocollo prevede delle interrogazioni periodiche degli host da parte dei router multicast per avere uno stato coerente ed aggiornato su quali host sono interessati alla ricezione di flussi multicast. Fondamentalmente, a ciascun host viene trasmesso con cadenza costante un pacchetto (Query) per tutta la durata della connessione con il fine di reperire l´informazione dell´eventuale Host Group di appartenenza di quello specifico host. Un host group non è altro che un insieme di host interessati a ricevere un determinato flusso e, quindi, identificati da un unico indirizzo multicast.
L´host risponde alle query con un secondo pacchetto (Report) che contiene l´ informazione degli host group a cui esso appartiene.
Il protocollo IGMP è parte integrante del protocollo IP e quindi i messaggi IGMP sono "incapsulati" in datagram IP.
I messaggi IGMP sono i seguenti:

- *Membership query*: inviato da router e può essere generale o specifico.
- *Membership report*: inviato da host
- *Leave group*: inviato da host

Il primo messaggio viene utilizzata da un router multicast per eseguire periodicamente il polling di una rete per i membri del gruppo. Con il protocollo IGMP versione 3, in una query di appartenenza il router può chiedere all'host di specificare le preferenze di ricezione del traffico multicast proveniente da un determinato elenco di origini.
Il secondo messaggio viene inviato quando un host entra a far parte di un gruppo multicast per dichiarare l'appartenenza a uno specifico gruppo di host. I messaggi IGMP di dichiarazione di appartenenza al gruppo di host vengono inviati anche in risposta a una query di appartenenza inviata da un router. Con il protocollo IGMP versione 3, nel messaggio di dichiarazione di appartenenza l'host può richiedere di ricevere il traffico multicast proveniente da origini specifiche oppure da qualsiasi origine escludendo un determinato insieme di origini. I rapporti specifici per origine impediscono ai router abilitati per multicast di inviare il traffico multicast a una subnet in cui non è presente alcun host in ascolto.
Il terzo messaggio viene inviato da un host quando questo abbandona un gruppo ed è l'ultimo membro di tale gruppo sul segmento di rete. Esistono due versioni di questo protocollo: IGMP v1 ed IGMP v2. Esse sono descritte rispettivamente nei documenti RFC 1112 e RFC 2236 (RFC: Request For Comment) a cui si rimanda per una descrizione dettagliata.
I messaggi IGMP vengono incapsulati e inviati all'interno dei datagrammi IP come mostrato nell'illustrazione che segue.



3.2 Formato dei messaggi IGMP

Il formato dei messaggi IGMP è il seguente:

TYPE	MRT	CHECKSUM
GROUP ADDRESS		

Formato di un datagramma IGMP

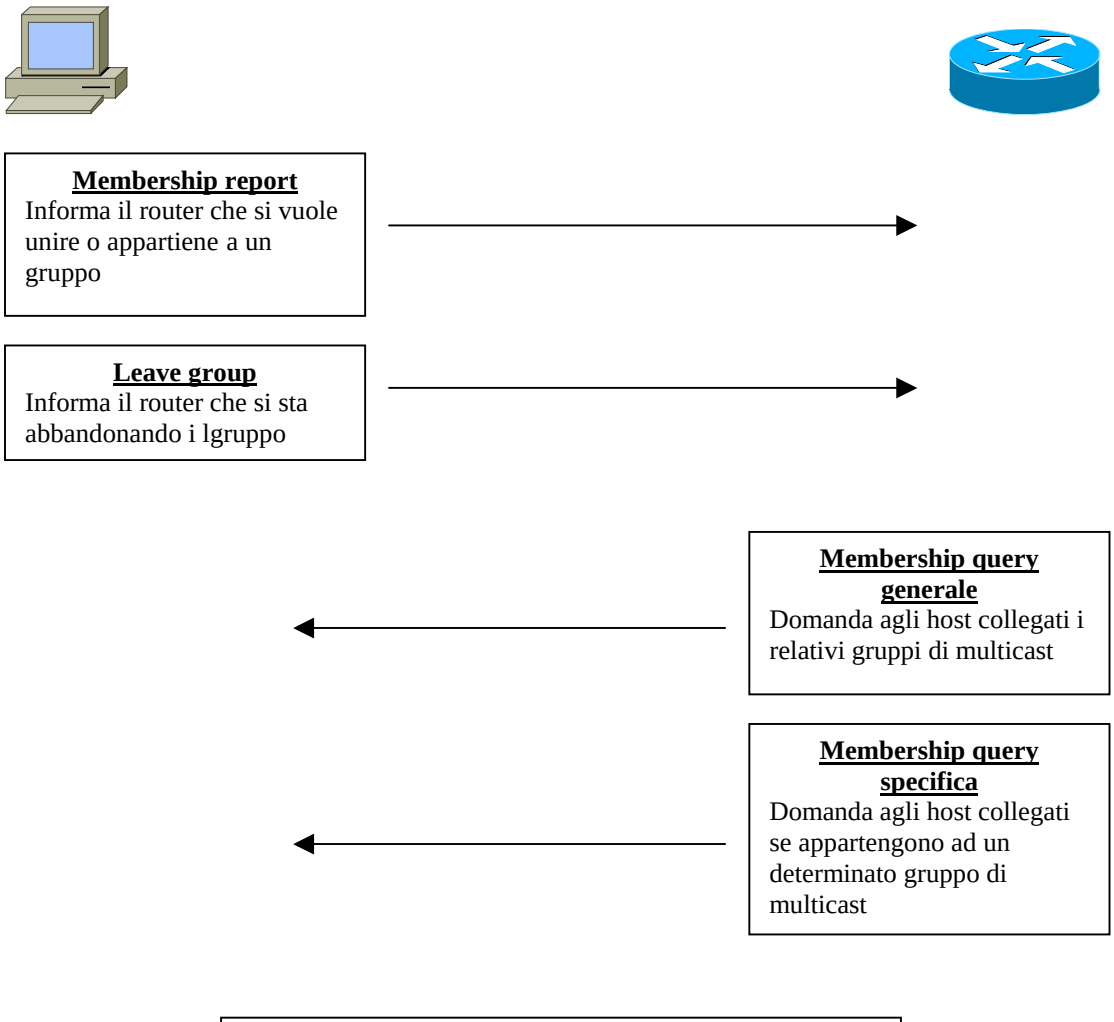
• TYPE (8 bit): specifica di quale messaggio IGMP si tratta ovvero *Membership query*, *Membership report* oppure *Leave group*. Nel caso in cui il Type corrispondesse ad un Membership query questo a sua volta potrebbe essere *General query* (consente di

conoscere quali gruppi sono attivi in una sotto rete) oppure *Group specific query* (consente di conoscere se ci sono membri attivi di un certo gruppo in una sottorete).

• MAX RESPONSE Time (MRT – 8 bit): nei messaggi di query specifica il timeout per l'invio di un *Membership Report* da parte degli host .

• CHECKSUM: questo campo è una somma di controllo per l'intero messaggio IGMP. Serve ad assicurare che le informazioni non si siano danneggiate durante il transito. Per calcolare le somme di controllo IGMP, si usa lo stesso algoritmo che serve per il calcolo delle somme di controllo dell'intestazione IP.

• GROUP ADDRESS: questo campo contiene il multicast IP del gruppo al quale un host dichiara di appartenere. Nel caso di un interrogazione di gruppo multicast, questo campo è impostato su valori tutti uguali a zero.



Schema dei quattro messaggi IGMP

3.3 Funzionamento

Ogni multicast router presente in una sottorete mantiene una lista dei gruppi di multicast attivi. Questo può essere di due tipi:

- *Querier*: Unico in una sottorete, ha il compito di gestire i colloqui con gli host, scambiando messaggi del protocollo IGMP, visti nel paragrafo precedente. Un router di questo tipo è eletto da una procedura automatica all'atto dell'inizializzazione della rete
- *Non Querier*: Si limita a mantenere aggiornata la lista dei gruppi multicast, informazione che consentirà di ottimizzare la procedura di inoltro dei pacchetti.

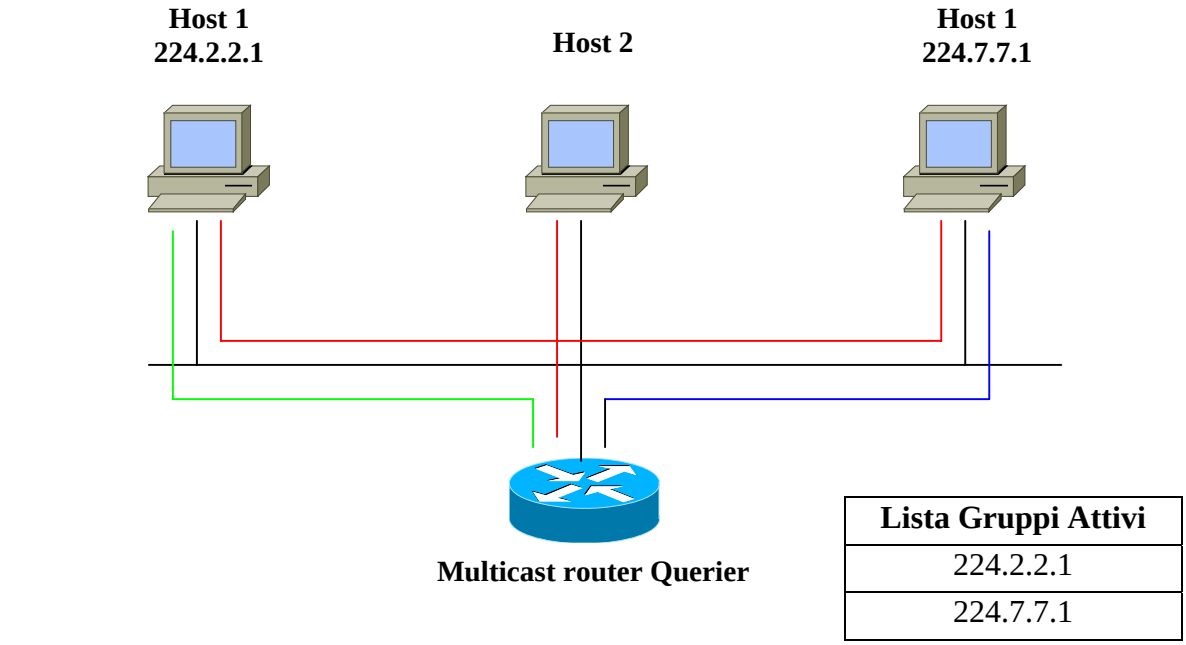
Il router Querier emette periodicamente messaggi di tipo general Query per raccogliere informazioni sullo stato dei gruppi multicast nella sottorete. Tale messaggio viene inviato a tutti gli host della sottorete all'indirizzo 224.0.0.1 e contiene l'informazione riguardante il Max Response Time. Quando un host riceve questo messaggio, fissa i valori del timeout associati ai gruppi multicast a cui appartiene a un valore random compreso tra 0 e Max Response Time.
Quando il timeout associato ad un gruppo multicast si esaurisce, l'host emette verso i multicast router un messaggio di Membership Report, tale messaggio contiene l'indirizzo del gruppo multicast associato.
A questo punto quando un router riceve un messaggio di report, aggiorna se necessario la lista dei gruppi multicast (evidentemente solo nel caso in cui qualcosa fosse cambiato dal precedente controllo) e resetta il timer associato a quel gruppo di multicast.
Nel caso in cui nessun messaggio di report è ricevuto prima che il timer di un gruppo multicast si esaurisca, il router assume che non vi siano più membri di tale gruppo nella sottorete, e successivamente cancellerà l'indirizzo corrispondente dalla propria lista. Il router non rilancerà più pacchetti in quella sottorete per quel gruppo di multicast.

I meccanismi definiti per l'uscita da un gruppo (*leave*) sono due, a seconda della situazione in cui ci si trova:

- Se l'host che esegue il *leave* è l'ultimo appartenente a quel gruppo nella sottorete, deve emettere un messaggio di *Leave Group* per segnalare ai router multicast l'evento.
- Se non è l'ultimo membro del gruppo, non è necessario effettuare alcuna procedura. Questo ottimizza il traffico di rete.

Quando un router querier riceve un messaggio di leave group, emette un messaggio di *Group Specific Querier* per quel gruppo e se untro un determinato timeout non riceve nessun messaggio di report, cancella il gruppo multicast dalla lista.

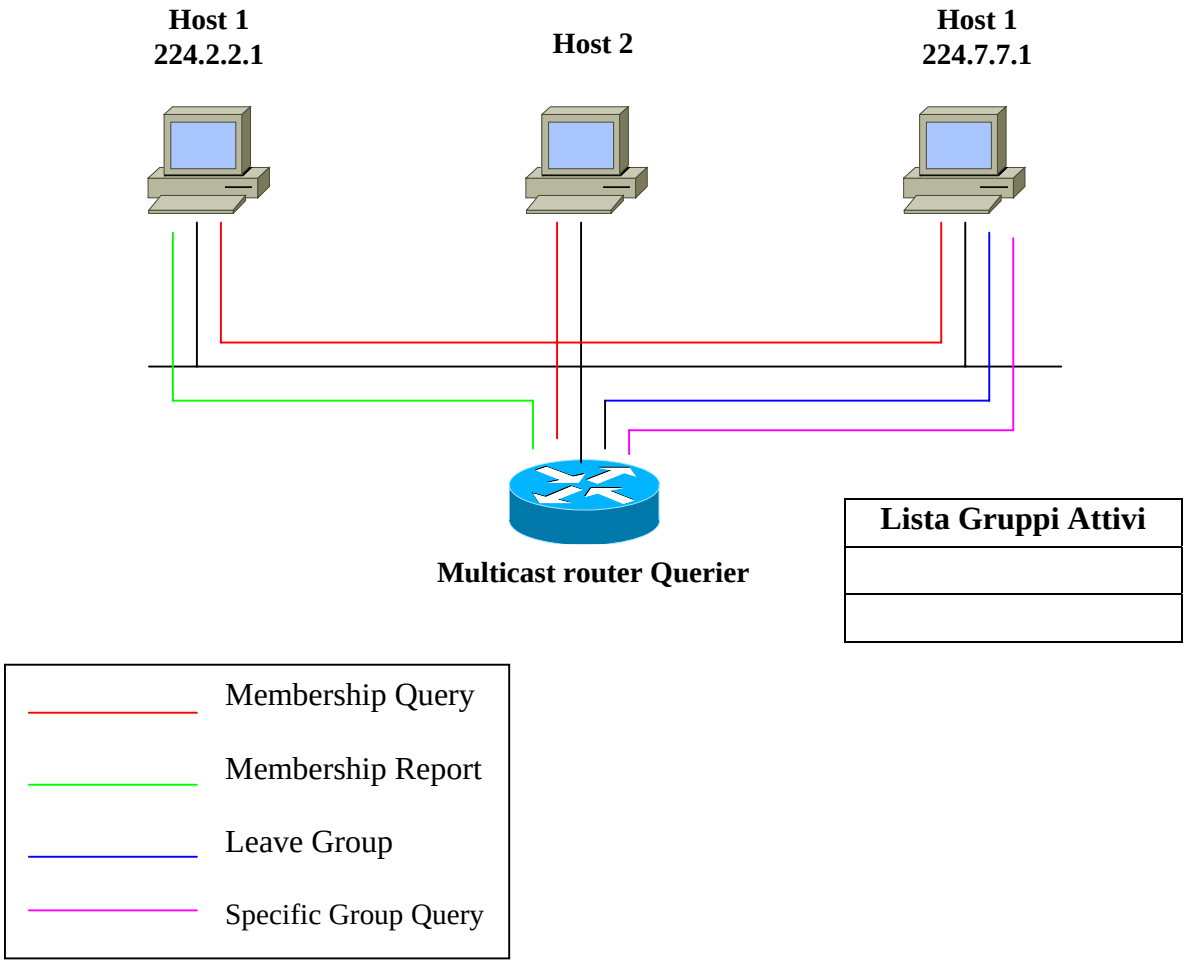
3.4 Esempio procedura di JOIN



Scambio di messaggi durante una procedura di JOIN

Il router Querier emette periodicamente un messaggio di *General Query*, chiedendo agli host adiacenti l'appartenenza ai rispettivi gruppi di multicast (evidenziato in figura in rosso). Gli host emettono i messaggi di membership report relativi ai gruppi a cui appartengono (evidenziati in figura in verde e blu). Da notare che l'host 2 non facendo parte di nessun gruppo, non invia al router alcun messaggio di Report. Con questa procedura il router aggiorna la lista dei gruppi multicast attivi.

3.5 Esempio procedura di LEAVE



Scambio di messaggi durante una procedura di LEAVE

Come prima il router Querier emette messaggi di tipo *General Query* verso tutti gli host della sottorete. Gli host emettono regolarmente i propri *Membership Report* fintanto che l'host 1 cessa di emetterli (l'host 1 non essendo l'ultimo membro del gruppo non deve eseguire la procedura di leave). L'host 3 a questo punto emette il messaggio di Leave Group (ultimo partecipante) ed il router manda un Group Specific Query, per verificare se fa ancora parte del gruppo di multicast a cui il router fa riferimento in tabella. Una volta verificato il *Leave* il router cancella il gruppo dalla tabella.

3.6 Diagrammi di stato

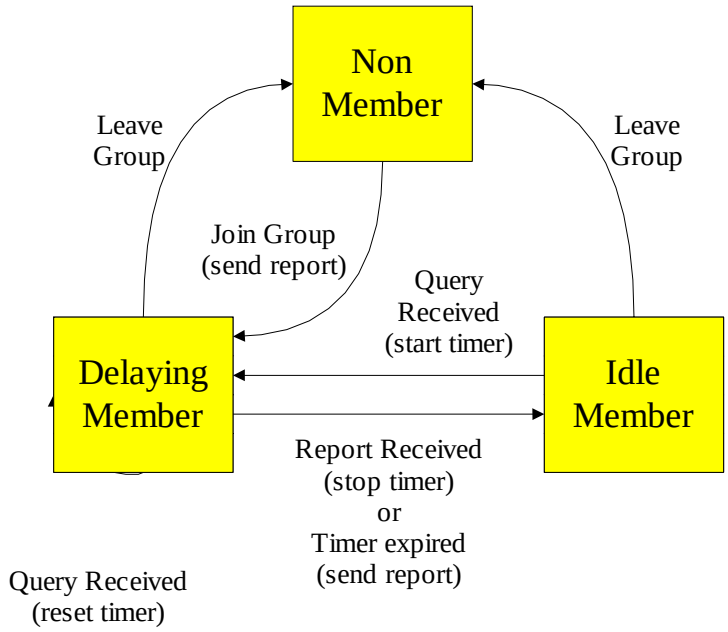


Diagramma di stato di un Host

- *Non Member*: l'host non appartiene al gruppo di multicast.
- *Dealing Member*: l'host appartiene al gruppo e ha il timer attivo per quel gruppo
- *Idle Member*: l'host appartiene al gruppo, ma non ha attivo il timer per quel gruppo

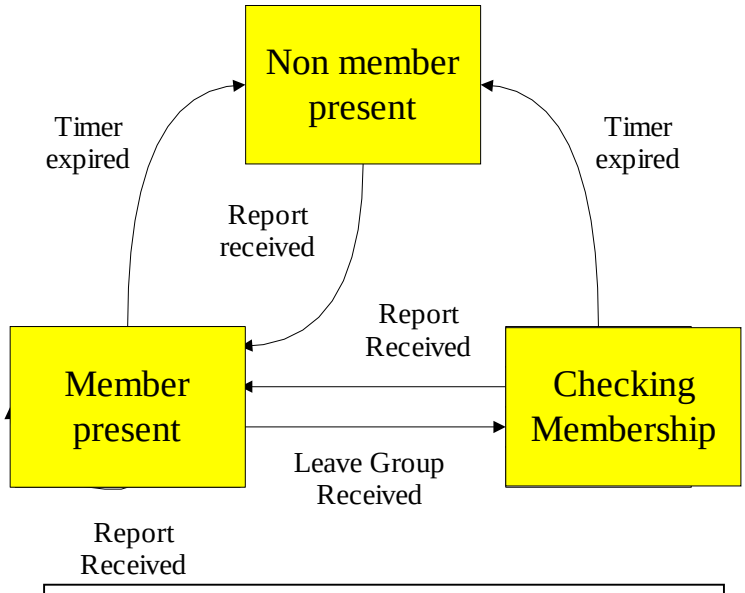


Diagramma di stato di un Router

- *Non Member Present*: non ci sono host appartenenti al gruppo multicast nella sottorete.
- *Member Present*: c'è almeno un host appartenente al gruppo multicast nella sottorete.
- *Checking Membership*: il router ha ricevuto il messaggio di *Leav Group* ma non è ancora scaduto il timer del messaggio di report.

Capitolo 4

Routing multicast

> “Le opinioni nuove sono sempre sospette e in genere contrastate per nessun'altra ragione all'infuori del fatto che non sono già comuni”.
>
> *John Locke*

In questo capitolo verranno descritti i protocolli di routing utilizzati dalla trasmissione Multicast. Capire il funzionamento di ciascuno di essi è importante per metterne in evidenza le reciproche differenze e poter esaminare in modo critico il protocollo DVMRP, impiegato, sui router, nelle prove di laboratorio esposte in seguito.
Verranno sviluppati i seguenti punti:

- Routing Multicast – il problema
- Algoritmi di routing Multicast
- Multicast BackBone

4.1 Cenni preliminari sul routing multicast

Un nodo con capacità multicast deve essere in grado di:

- Inviare e ricevere pacchetti multicast.
- Registrare gli indirizzi multicast ascoltati dal nodo con router locali, in modo che sia possibile inoltrare i pacchetti multicast alla rete del nodo.Le applicazioni di multicast IP che inviano traffico multicast devono creare pacchetti IP con l'indirizzo multicast IP appropriato definito come indirizzo IP di destinazione. Le applicazioni di multicast IP che ricevono traffico multicast devono informare il protocollo TCP/IP del fatto che sono in ascolto di tutto il traffico diretto a uno specifico indirizzo multicast IP. I nodi IP utilizzano il protocollo IGMP (Internet Group Management Protocol) per registrare il proprio interesse alla ricezione del traffico multicast IP proveniente dai router IP. I nodi IP che utilizzano il protocollo IGMP inviano un messaggio di rapporto di appartenenza IGMP per informare i router locali del fatto che sono in ascolto su uno specifico indirizzo multicast IP.

Un router con capacità multicast deve essere in grado di:

- Ascoltare tutto il traffico multicast su tutte le reti connesse. Alla ricezione di traffico multicast, inoltrare il pacchetto multicast alle reti connesse in cui sono presenti nodi in ascolto o in cui i router hanno nodi in ascolto.
- Ascoltare i messaggi di rapporto di appartenenza IGMP e aggiornare la tabella di inoltro multicast TCP/IP.
- Utilizzare un protocollo di routing multicast per propagare le informazioni di ascolto dei gruppi multicast agli altri router dotati di capacità multicast.
-

4.2 Routing Multicast – il problema

Dato il modello di servizio multicast fornito precedentemente, Il problema che viene da porsi è come facciano i router a smistare il traffico dai mittenti ai destinatari, nonostante questi non siano a conoscenza della topologia della rete. Potremmo assumere che se il mittente ed il destinatario avessero informazioni l'uno dell'altro, potrebbero trasmettere e ricevere tramite una trasmissione di tipo unicast. In altre parole, esiste una rete con percorsi bidirezionali e un meccanismo di fondo di tipo unicast già funzionante.
Una volta che i *router* sono conoscono la presenza di host ad essi connessi appartenenti a gruppi e sono in grado di inviargli e/o ricevere l'informazione *multicast*, il problema si sposta nel gestire il *routing muticast* fra i *router.*

Dato questo scenario, per risolvere il problema che ci siamo posti esiste una gamma di soluzioni: da un lato si può pensare di inviare il traffico verso a tutti i possibili destinatari, avendo dei router anche in quei segmenti di rete dove non sono presenti effettivamente dei ricaventi. Dall'altro lato si può pensare di scambiare informazioni attraverso un protocollo di routing Multicast, che informa i router della posizione di tutti i riceventi; questa procedura viene eseguita per tutti i possibili mittenti.
Su scala globale, nessuno dei due metodi esposti è conveniente; la soluzione migliore sta nella nel mezzo dei due estremi.

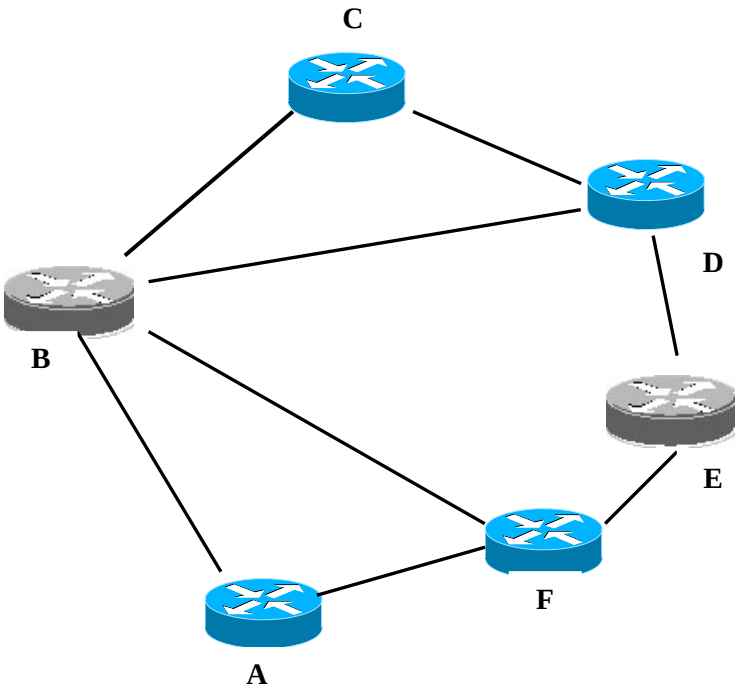
Le difficoltà a cui deve far fronte un algoritmo di routing multicast sono i seguenti:
- Modifiche nella struttura dei gruppi impongono modifiche nell'instradamento
- L'instradamento dipende dalla posizione del mittente e non solo dall'indirizzo multicast del destinatario.
- L'invio è anche permesso ad host che non appartengono al gruppo
- L'inoltro può attraversare reti che non hanno alcun host appartenente al gruppo
- Sono necessarie informazioni aggiuntive rispetto all'indirizzo di destinazione

Nel mondo reale, ci sono molti protocolli di routing ognuno con i propri vantaggi e svantaggi. Nei prossimi paragrafi verranno esposti i principali algoritmi di routine impiegati su scala globale.

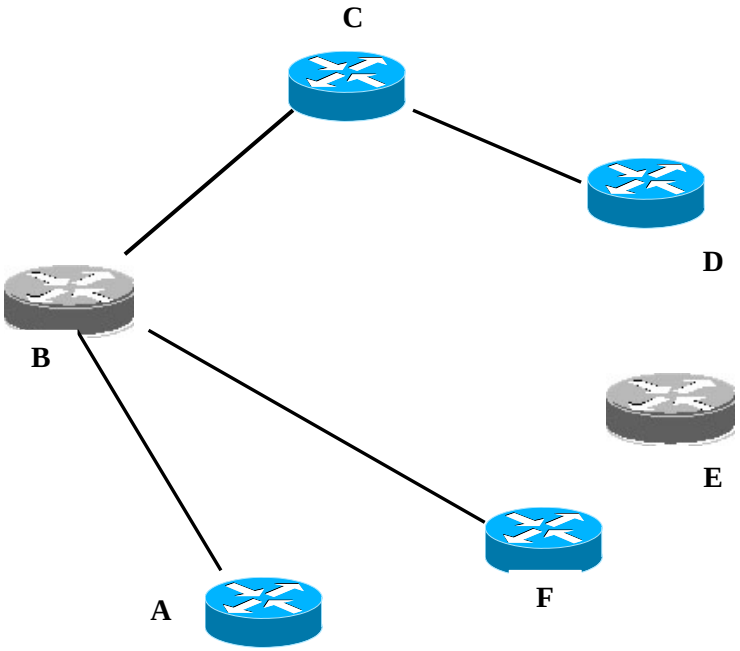
4.3 Algoritmi di routing Multicast

Si suppone, per semplicità, che ogni membro del gruppo multicast, oltre al ruolo di receiver, assuma anche il ruolo di sender. Per individuare l'albero di instradamento dei pacchetti dal sender al receiver, possono essere utilizzati due approcci:

- ***Shared Distribution Tree*:** si utilizza un unico albero di distribuzione condiviso tra tutti i sender
- ***Source Distribution Tree***: per ciascun sender si utilizza un albero di distribuzione specifico, diverso dagli altri

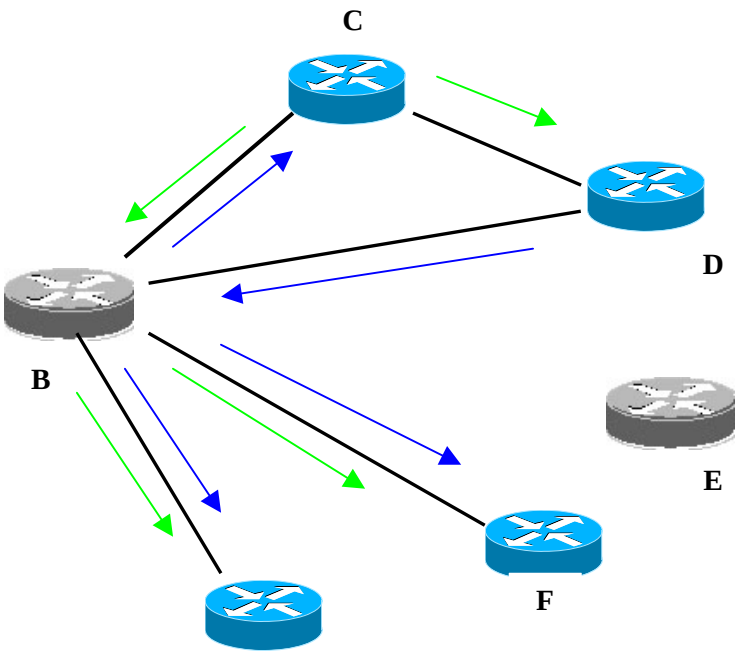


Rete in cui sono presenti sender e receiver



Albero calcolato usando l'algoritmo Shred Distribution Tree

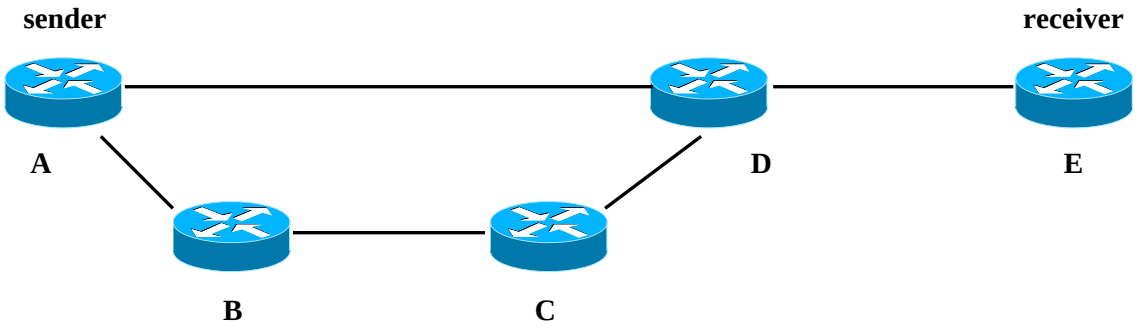
Si tratta di trovare lo *spannino tree* la cui somma dei costi sui link che lo compongono sia minima. Avendo un unico albero e potenzialmente più sorgenti, non si può ottimizzare il percorso rispetto ad una specifica sorgente, di conseguenza l'unica ottimizzazione fattibile è quella che cerca il *Minimum weigth spannino tree* (MST). Nonostante l'esistenza di metodi per il calcolo del MST, su Internet non viene utilizzato questo approccio per il calcolo ottimale del percorso per varie ragioni: Bisogna conoscere il costo di ogni link sulla rete e conseguentemente ripetere i calcoli per ogni cambio di costo; Non si riesce a sfruttare correttamente le tabelle di routine già calcolate per il routing unicast; le prestazioni hanno comunque dei limiti, perché il costo medio e quello massimo per coppia sorgente – destinazione del gruppo sono elevati.
Un approccio alternativo in questa categoria è il *Center Based Tree* che fa uso di un nodo di riferimento. Tutti i router con un host che aderisce ad un gruppo, inviano un messaggio di *join* lungo il percorso unicast verso il nodo del gruppo; fino a che il messaggio o raggiunge il centro o incontra un router già parte del gruppo, crea un percorso dell'albero.



Albero calcolato usando l'algoritmo Source Distribution Tree

Procedendo con quest'altro approccio ora il problema è di calcolare un percorso minimo riferito a ciascun possibile sender della rete. Gli algoritmi come Quello di Dijkstra, Steiner, Kou, Ottimo ricavano uno *Sort-Path Spannino Tree* per ogni nodo. Depurando l'informazione della rete dei *router* noninteressati al gruppo, ogni *router* può calcolare lo *shor tpath spanning tree* del gruppo per qualunque sorgente.
Un modo elegante e più semplice per realizzare l'instradamento *multicast* è utilizzare il *Reverse Path Forwarding* (RPF). Nel RPF una sorgente invia i pacchetti di dati sulla

sua rete locale; non appena un router riceve questi pacchetti multicast, effettua un controllo per accertare che l´interfaccia di provenienza utilizzata sia effettivamente la migliore e, quindi, più breve. In questo caso i pacchetti vengono rispediti su tutte le interfacce eccetto quella di provenienza, altrimenti verrebbero scartati perché rischierebbero di occupare due volte la banda a disposizione.
In sostanza l´RPF genera alberi di scansione diversi a seconda della sorgente garantendo un utilizzo più distribuito dei nodi della Rete e, quindi, una maggior velocità di recapito dei pacchetti multicast. E´ necessario, però, che i router, che appartengono alla lista di recapito del router a monte, abbiano almeno un membro collegato e interessato alla comunicazione multicast altrimenti questi vengono letteralmente potati dall´albero in modo da non occupare inutilmente le infrastrutture; i messaggi di potatura generati da questa operazione vengono, a questo punto, rispediti all´indietro nell´albero per creare una sorta di database dinamico che tenga conto della distribuzione dei router nella Rete interconnessa.
Un modo per migliorare l' RPF è dato dall'utilizzo della tecnica di *pruning* (potatura), ossia l'esclusione dalla disseminazione dell'informazione dei nodi "foglia" (*leaf*) o terminali non interessati ad un gruppo. Per fare ciò bisogna identificare le foglie e comunicare l'eventuale assenza di partecipanti ad un gruppo inviando un messaggio di *Pruning* verso l'indirizzo sorgente. L'eventuale ri-inserimento può avvenire tramite richiesta esplicita (*graft*) o automaticamente legando il *pruning* ad un *timeout*.
Si osservi, però, che l'RPF non evita la presenza di copie multiple dello stesso pacchetto e quindi non corrisponde a utilizzare uno *Shortest tree*. Un miglioramento (***Extended* RPF**) lo si ottiene imponendo che un nodo C invii il pacchetto verso D solo se il percorso più corto da A (sorgente) a D include C stesso.



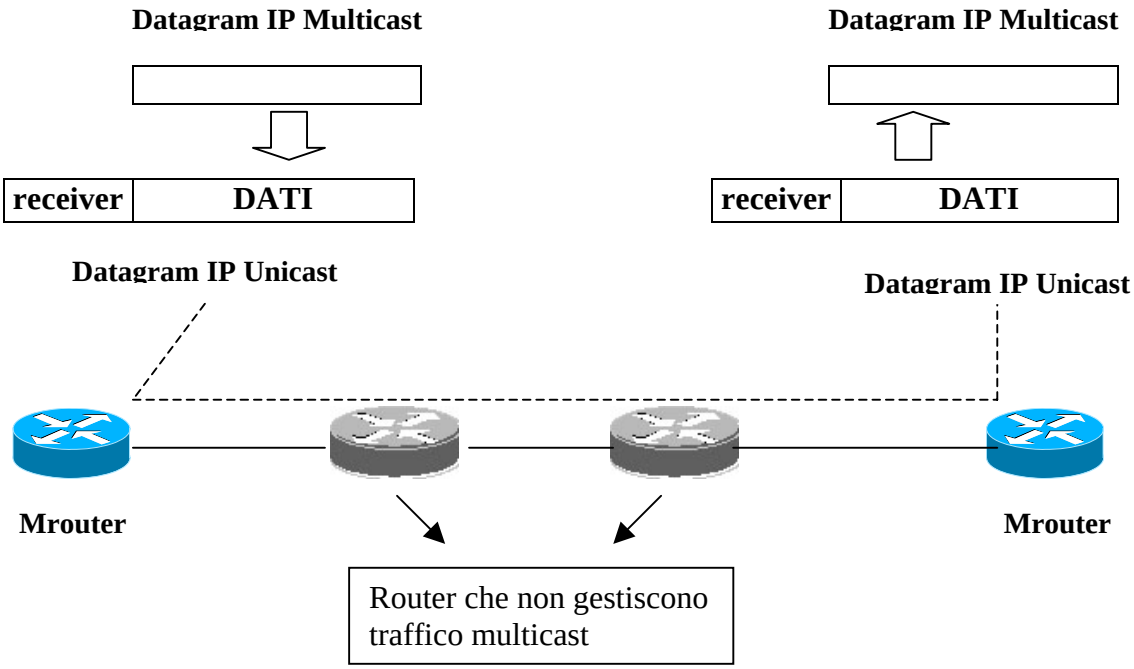
Esempio di applicazione dell' Extended Riverse Path Forwarding

Questo implica però che nodo sappia di essere transito per quella sorgente.

4.4 Multicast BackBone (MBone)

Le funzionalità di routing Multicast non sono presenti in tutti i router, infatti per fornire servizi multicast in Internet è stata realizzata una rete virtuale che intercorre tutti i router multicast. Tale rete è stata chiamata *Multicast BackBone* (MBONE). Nata nel 1992,

essa e' basata su una estensione del protocollo IP, l'IP Multicast, che sfrutta la classe D di indirizzi IP (quelli compresi nel range che va da 224.0.0.0 a 239.255.255.255); essa costituisce una sorta di "rete multicast virtuale" creata sopra le dorsali fisiche (backbone) e le reti regionali. La sua ideazione e' nata dalla necessita' di poter inviare dati audio e video su una rete geografica ad un certo numero di destinatari senza congestionare il traffico. La topologia di MBone e' stata progettata in maniera tale da facilitare la distribuzione efficente dei pacchetti, riducendo al minimo il carico delle linee che collegano i nodi intermedi. Questi nodi vengono chiamati IP Multicast Router o anche mrouter, ed hanno il compito di distribuire e replicare i dati ai loro destinatari.
In pratica diverse sottoreti multicast sono state collegate tramite dei tunnel, cioè dei semplici collegamenti unicast sui quali i pacchetti multicast viaggiano incapsulati in normali pacchetti IP, l'instradamento è gestito dai protocolli di routing multicast all'interno delle sottoreti mentre sui tunnel vengono usati i normali protocolli di instradamento unicast



Esempio di impiego di un tunnel

Ogni tunnel è definito da un local end point (chi trasmette) e da un remote end point (chi riceve). E' specificato un *Metric*, ovvero un valore dinamico che rappresenta il costo (riferito ai link su cui avviene la trasmissione) ed un valore di *Threshold* anch'esso dinamico, ed è il valore minimo del Time To Leave (TTL) perché il pacchetto possa essere instradato nel tunnel. Ogni Mrouter decrementa il TTL di 1 nel pacchetto Multicast.

4.5 protocolli di routing Multicast

All'interno della rete virtuale Mbone vengono impiegati i seguenti protocolli di routine multicast:

- Protocolli Flood and Prune
  - Distance Vector Multicast Routing Protocol (DVMRP)
  - Protocol Indipendent Multicast Sparse Mode (PIM DM)
- Multicast OSPF (MOSPF)
- Protocolli Center Based Tree (CBT)
  - Core Based Tree (CBT)
  - PIM Sparse Mode (PIM SM)

MOSPF

Ogni router calcola la via da se' stesso verso tutte le possibili destinazioni costruendo un albero per ogni sorgente/gruppo. Un pacchetto IP multicast è indirizzato in base alla sorgente e alla destinazione secondo una tecnica nota come source/destination routing; dopo essere stato instradato il pacchetto segue la via più breve e viene replicato solo quando il percorso per i diversi host diverge. MOSPF funziona solo nelle reti che utilizzano OSPF e lavora bene quando ci sono pochi gruppi attivi.
Nel MOSPF l'indirizzamento del datagramma multicast ha le seguente proprietà:

- Il path risultante dall'analisi del datagramma dipende sia dal mittente che dal destinatario a differenza di molti algoritmi per il routing unicast che considerano solo l'indirizzo del destinatario.
- La strada scelta per mandare il datagramma è quella con il minimo costo possibile (dove il costo si esprime funzione della metrica OSPF) - la metrica del OSPF è configurabile, viene assegnata ad ogni router di interfaccia una metrica che rappresenta il costo per mandare un pacchetto, il costo dell'invio di un datagramma verso un router dato dalla somma di tutte le metriche dei router da dove passa il pacchetto.
- Il routing MOSPF cerca di approfittare al massimo dei pezzi di strada comune a tutti i pacchetti (quando il multicast viene indirizzato a più reti diverse è inevitabile la duplicazione dei pacchetti).
- Per un datagramma dato i router calcolano un shortest path identico.
- Ad ogni passo il MOSPF manda i pacchetti multicast come data-link multicast con due eccezioni: sulle reti non broadcast, e poi in situazioni anomale quando il router MOSPF viene configurato in modo da instradare pacchetti multicast come data-link unicast.

PIM (Sparse Mode – Dense Mode)

Protocol-Independent Multicast supporta due tipi di traffico multipoint: denso e sparso.
Il modo denso è più utile quando:

- I partecipanti sono vicini
- Ci sono pochi sender e tanti receiver

- Il traffico Multicast è alto e costante

Dense-mode PIM utilizza la tecnica Reverse Path Forwarding e assomiglia tanto al DVMRP. La differenza sta nel fatto che PIM non necessita di particolari protocolli unicast per il funzionamento.
Il modo sparso è più utile nel caso in cui:

- Ci sono pochi host in ricezione
- Tra i mandanti e i ricevitori ci sono i link WAN
- Il traffico è intermittente

Sparse-mode PIM lavora definendo un Rendezvous Point. Quando un host chiama, prima manda i dati al Rendezvous Point e la stessa cosa devono fare anche gli host in ricezione. Dopo che i dati cominciano a fluire dal chiamante al destinatario i routers sulla strada ottimizzano il collegamento. Sparse-mode PIM assume che nessun altro computer necessita di ricevere traffico multicast a parte quelli che lo hanno sollecitato esplicitamente.
PIM è in grado di supportare simultaneamente il modo denso per certi gruppi e il modo sparso per altri.

CBT

Il *Core Base Tree* è cronologicamente il primo dei protocolli *Center Based Tree*. Il CBT costruisce un albero bidirezionale, di modo che i pacchetti possano viaggiare sia in direzione core (nucleo dell'albero), che nella direzione opposta a seconda della direzione della sorgente. La sorgente non deve necessariamente appartenere al tree, in ogni caso il pacchetto viene inviato verso il *Core*, il primo nododell'albero che raggiunge, viene propagato sull'albero stesso. Nella rete può essere presente più di un core. I limiti intrinseci di questo protocollo Il fatto che non è sempre facile posizionare il core, e se non viene fatto correttamente l'albero risulta inefficiente. Non esiste poi un metodo consolidato per legare l'indirizzo del core a quello del gruppo.

DVMRP

DVMRP (Distance Vector Multicast Routing Protocol) combina il RIP (utilizzato all'inizio per la sua semplicita') con l'algoritmo Truncated Reverse Path Broadcasting (TRPB).
DVMRP e' un "interior gateway protocol" fatto per l'uso in un autonomous system, ma non tra diversi autonomous systems. DVMRP non e' stato sviluppato per l'uso con datagrammi di routing non-multicast, sicché un router che fa sia il routing unicast che il routing multicast deve eseguire due processi separati. Con tutto ciò il DVMRP è stato progettato per essere facilmente esteso al routing unicast. In piu' negli esperimenti sulle reti che non supportano il multicast e' stato utilizzato il meccanismo di tunneling. Per costruire l'albero di instradamento sono necessarie più informazioni sullo stato dei link di quanto il RIP è in grado di offrire, e come risultato il DVMRP è molto piu complesso di quanto non lo sia il RIP. DVMRP utilizza Internet Group Management Protocol (IGMP) per il routing. I datagrammi DVMRP hanno due porzioni: un header IGMP piccolo e di lunghezza fissa, e una parte per il flusso di dati. Il protocollo DVMRP è responsabile del calcolo del routing e scambio di informazioni sull'appartenenza ai gruppi e sul costo dell'instradamento. Per ogni gruppo i gateway costruiscono un albero

di instradamento; quando un gateway riceve un datagramma destinato ad un indirizzo IP multicast lo invia lungo i collegamenti che corrispondono ai vari rami nell'albero. Il protocollo prevede che ogni mrouter individui nell'albero la sua posizione relativa ad una data sorgente e determini quale delle sue interfacce virtuali sono ottimali per l'invio del datagramma. Un problema non facile è la determinazione delle foglie (sottoreti senza tunnel uscenti) poiché il numero di router in una rete virtuale è dinamico. Questo protocollo scandisce periodicamente ogni interfaccia virtuale per aggiornare l'albero secondo una tecnica chiamata Reverse Path Forwarding. DVMRP implementa il proprio protocollo unicast molto simile al RIP. Come risultato si ha che non è obbligatorio che tra due computer pacchetti unicast e multicast seguano lo stesso percorso. I protocolli di instradamento sono ancora in fase di sperimentazione ; DVMRP è il più uilizzato però è un po' lento e non in grado di gestire rapide variazioni della topologia di rete.

Capitolo 5

Prove pratiche di routing multicast

> "Se si escludono istanti prodigiosi e singoli che il destino ci può donare, l'amare il proprio lavoro (che purtroppo è privilegio di pochi) costituisce la migliore approssimazione alla felicità sulla terra. Ma questa è una verità che non molti conoscono".
>
> P*rimo* L*evi*

In questo capitolo verrà descritto in dettaglio come sono state realizzate le prove in laboratorio riguardo il routing multicast. Lo scopo di tali prove è verificare come i router multicast che implementano il protocollo DVMRP smistano il traffico all'interno di un autonomous system, quali messaggi vengono scambiati tra mrouter e host e come il Protocollo di routing DVMRP si comporti in presenza di situazioni di transitorio. Le prove riportate di seguito nel presente capitolo daranno modo di constatare come effettivamente i protocolli citati nelle pagine precedenti lavorino effettivamente ed avere modo di chiarire certi aspetti del routing multicast. Tali considerazioni serviranno per trarre conclusioni sull'effettiva efficacia del protocollo DVMRP.
Verranno sviluppati i seguenti punti:

- Implementazione della rete
- Software utilizzato
- Configurazione del demone MROUTED
- Cattura e analisi del traffico multicast

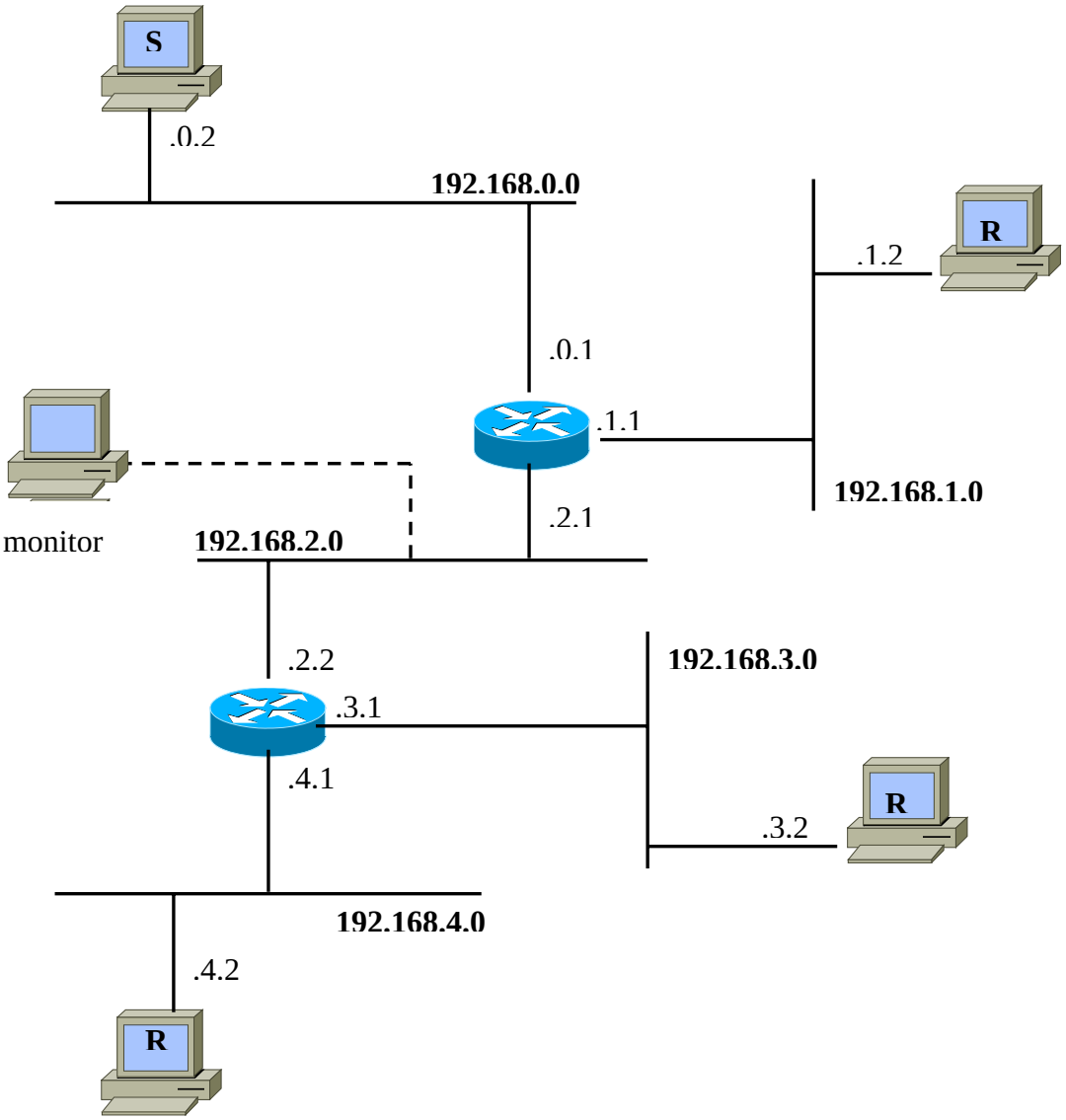
5.1 Implementazione della rete

Presso il laboratorio di reti è stata implementata una rete di PC, la cui topologia è stata studiata in modo tale da poter mettere in evidenza alcuni aspetti del routing multicast. Una volta implementata la rete, con software opportuni, verrà avviata la trasmissione multicast da un sender, verso più receiver passando per due mrouter. Oltre che alla topologia della rete verrà mostrata la configurazione dei PC per rendere possibile il traffico Unicast, base per il traffico Multicast.
La rete è composta dai seguenti elementi:

- Quattro PC (sia sender che receiver).

- Due PC impiegati come Mrouter, sui quali gira un software apposito (discusso in seguito).
- Un PC impiegato da monitor; questo PC non prende parte attivamente alla trasmissione Multicast, ma viene utilizzato come monitor per catturare il traffico in determinati segmenti di rete.
- Switch di rete.

Questo che segue è lo schema della rete utilizzata:



Schema della rete realizzata il laboratorio

La rete sopra riportata, è composta da cinque sottoreti, unite dai due router che si vedono nella parte centrale.

Per quanto riguarda la configurazione del routing Unicast, sono state configurate delle route statiche, per due motivi:

- Semplicità nell'implementazione
- Usando route statiche il traffico multicast che in seguito si vuole catturare, non verrà "sporcato" da altri protocolli.

Sulle macchine sender e receiver è stato sufficiente impostare l'indirizzo IP corretto, in moda da non generare conflitti di indirizzi ed il Gateway, ovvero l'interfaccia del router che che si trova fisicamente sul segmento di rete in questione.
Per quanto riguarda i router, il sistema operativo utilizzato è FreeBsd con versione del Kernel 6.0. E' stato scelto questo sistema operativo per la sua stabilità e praticità nell'ambito della configurazione inerente al networking.
I comandi utilizzati durante la configurazione del traffico IP unicast sono i seguenti:

- *Ifconfig*: permette di settare la varie caratteristiche delle interfacce di rete. Se usato senza opzioni il comando elenca tutte le interfacce attive; questo è il primo passo da fare sempre prima di qualunque configurazione (ed anche dopo per controllare che sia tutto a posto) per vedere lo stato del sistema. Un risultato possibile è il seguente:

```
eth0      Link encap:Ethernet  HWaddr 00:01:02:2F:BC:40
          inet addr:194.177.127.234  Bcast:194.177.127.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:82075264 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51585638 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:2858378779 (2.6 GiB)  TX bytes:2524425895 (2.3 GiB)
          Interrupt:10 Base address:0x8800

eth0:0    Link encap:Ethernet  HWaddr 00:01:02:2F:BC:40
          inet addr:192.168.1.1  Bcast:192.168.1.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:10 Base address:0x8800

eth1      Link encap:Ethernet  HWaddr 00:E0:7D:81:9C:08
          inet addr:192.168.168.1  Bcast:192.168.168.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:9 Base address:0x6000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10226970 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10226970 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1385547296 (1.2 GiB)  TX bytes:1385547296 (1.2 GiB)
```

come si vede sulla macchina sono presenti 3 interfacce di rete; due di esse (*eth0* e *eth1*) corrispondono a due schede di rete, la terza (*lo*) è una interfaccia logica, la cosiddetta

interfaccia di *loopback* che viene usata per le comunicazioni locali, e che deve essere sempre attivata anche per i computer non connessi in rete. Prima di poter configurare una interfaccia occorre verificare che essa non sia già attiva. Supponiamo che si tratti di eth0. Il comando ifconfig eth0 ci mostrerà lo stato dell'interfaccia (dando un errore nel caso il supporto nel kernel non sia attivato).

- *Sysctl*: è un'interfaccia che permette di effettuare cambiamenti ad un sistema FreeBSD già attivo. Questo include molte opzioni avanzate dello stack TCP/IP e del sistema di memoria virtuale che possono permettere di migliorare drammaticamente le prestazioni. Più di cinquecento variabili di sistema possono essere lette e modificate usando sysctl. In sostanza, sysctl serve a due cose: a leggere e a modificare le impostazioni di sistema. Per visualizzare tutte le variabili modificabili si usa l'opzione

  ```
  Sysctl –a
  ```

  Per impostare una particolare variabile, usa l'intuitiva sintassi *variabile=valore*; I valori validi per le variabili di sysctl sono generalmente o stringhe, o numeri, o valori booleani (un valore booleano può valere 1 per sì o 0 per no). In questo caso la variabile da settare è quella che permette il forwarding:

  ```
  sysctl  inet.net.ipv4.forwarding=1
  ```

- *Route*: Route è il comando Linux che viene utilizzato per manipolare le tabelle di routing. Permette di aggiungere ed eliminare route statiche e default gateway, oltre che semplicemente visualizzare la tabella di routing di un sistema.
  Per aggiungere una routing statica il comando è il seguente

  ```
  Route add –net 192.168.0.0 192.168.0.1
  ```

- *Netstat*: per vedere le tabelle di routing attive:

  ```
  netstat -nr
  ```

5.2 Software utilizzato

In questo paragrafo verrà elencato ed esposto il software che è stato utilizzato per rendere possibili le prove fatte in laboratorio.

- I sistema operativi installato sulle macchine sender e receiver è Windows XP Professional, mentre quello installato sui PC adibiti a router è FreeBSD 6.0
- Il programma usato per trasmettere e ricevere a/da un gruppo di multicast è VLC media player. VLC media player (originariamente chiamato Video lan Client) è un media player del progetto VideoLAN. VLC supporta molti codec audio e video, formati file come DVD, VCD e vari protocolli per lostreaming. Può essere utilizzato anche come server per trasmettere in stream, in unicast o in multicast su IPV4 o su IPV6 su un network a larga banda. Il media player impiega la libreria codec libavcodec del progetto FFmpeg per maneggiare molti dei formati supportati, ed utilizza la libreria di decriptazione DVD libdvdcss per gestire i playback dei DVD cifrati.VLC è uno dei media player più disponibile su varie piattaforme, infatti è

disponibile per Linux, Microsoft Windows, Mac OS X, BeOS, BSD, Pocket PC, Solaris.

- Per la cattura di pacchetti è stato utilizzato lo sniffer di rete Ethereal. è un software di analisi di protocollo, o packet sniffer (annusa-pacchetti) utilizzato per la risoluzione di problemi di rete, per l'analisi e per lo sviluppo di software e di protocolli di comunicazione e per la didattica. Ethereal possiede tutte le caratteristiche standard di un analizzatore di protocollo. Le funzionalità che Ethereal offre sono molto simili a quelle di tcpdump, ma con un'interfaccia grafica, e con molte più funzionalità di ordinamento e filtraggio. Permette all'utente di vedere tutto il traffico presente sulla rete (tipicamente una Ethernet, ma sono stati aggiunti altri tipi di rete) mettendo la scheda di rete in modalità *promiscous*. Wireshark è un software che riesce a "comprendere" la struttura di diversi protocolli di rete, quindi è in grado di visualizzare incapsulamenti e campi singoli, ed di interpretare il loro significato. Wireshark non dispone di una propria base di codice per catturare i pacchetti, ma utilizza libpcap/WinPcap per svolgere questo compito. Di conseguenza, Wireshark può funzionare solo su reti supportate da libpcap o WinPcap.
- Per eseguire sui router il demone Mrouted (programma che implementa il protocollo di routing DVMRP) bisogna prima ricompilare il Kernel di FreeBSD per abilitare il traffico Multicast. Tradizionalmente, FreeBSD ha sempre avuto quello che si chiama un kernel “monolitico”. Questo significa che il kernel era un programma di grandi dimensioni, supportava una lista fissa di device, e se tu avessi voluto cambiare il comportamento del kernel avresti dovuto compilarne uno nuovo, quindi fare il reboot del tuo computer per caricare il nuovo kernel. Oggi come oggi, FreeBSD si sta muovendo rapidamente verso un modello dove gran parte delle funzionalità del kernel sono contenute in moduli che possono essere caricati e scaricati dal kernel a seconda delle necessità. Questo permette al kernel di adattarsi a nuovo hardware appena questo diventa disponibile (come ad esempio le carte PCMCIA in un laptop), oppure fa sì che nuove funzionalità siano portate nel kernel, funzionalità che non erano necessarie quando il kernel fu compilato inizialmente. Questo è noto come kernel modulare. Nonostante questo, è ancora necessario portare avanti delle compilazioni statiche del kernel. In alcuni casi questo è necessario perchè la funzionalità è così legata al kernel che non può essere resa caricabile dinamicamente. In altri casi può essere necessario semplicemente perchè nessuno si è ancora preso il tempo di scrivere un modulo caricabile dinamicamente per quella funzionalità. L’esempio riportato di compilazione del Kernel è stato fatto sotto architettura i386.

  Se *non* c'è una directory /usr/src/sys sul sistema, significa che i sorgenti del kernel non sono stati installati. Il modo più semplice per farlo è eseguire sysinstall (/stand/sysinstall su FreeBSD di versione precedente alla 5.2) come root, scegliendo Configure, poi Distributions, poi src, poi sys. Un’alternativa a **sysinstall** è quella di installare i sorgenti dalla linea di comando da un CDROM “ufficiale” FreeBSD; i comandi sono i seguenti:

```
# mount /cdrom
# mkdir -p /usr/src/sys
# ln -s /usr/src/sys /sys
# cat /cdrom/src/ssys.[a-d]* | tar -xzvf –
```

Quindi, si entra nella directory *arch*/conf e si copia il file di configurazione del Kernel con il nome che vuoi dare al Kernel. Ad esempio:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYKERNEL
```

Ora è il momento di editare MYKERNEL con un qualsiasi editor di testi. Se si partendo da zero, il solo editor disponibile, probabilmente è **vi**, che è troppo complesso per essere spiegato in questa sede; comunque, FreeBSD offre un semplice editor chiamato **ee**.
Il formato generale di un file di configurazione è abbastanza semplice. Ogni linea contiene una parola chiave ed uno o più argomenti. Per semplicità, la maggior parte delle linee contiene solo un argomento. Tutto quello che segue un **#** è considerato un commento ed ignorato. Per una lista esaustiva delle opzioni dipendenti dall'architettura e dei devices, si legga il file NOTES nella stessa directory del file GENERIC. Per opzioni indipendenti dall'architettura, leggi /usr/src/sys/conf/NOTES.
A questo punto dobbiamo attivare l'opzione del routine Multicast aggiungendo alle optino già presenti la seguente:

```
option MCASTING
```

A questo punto devi compilare i sorgenti del kernel. Ci sono due procedure per farlo:

Procedura 1. Compilare il Kernel nel Modo "Tradizionale"

Eseguire config per generare il nuovo sorgente del kernel.

```
# /usr/sbin/config MYKERNEL
```

Andrea nella directory di compilazione. Config scriverà il nome di questa directory dopo essere stato eseguito come sopra.

```
# cd ../compile/MYKERNEL
```

Per versioni di FreeBSD precedenti all 5.X, seguire queste istruzioni:

```
# cd ../../compile/MYKERNEL
```

Compilare il kernel.

```
# make depend
# make
```

Installare il nuovo kernel.

```
# make install
```

Procedura 2. Compilatre il Kernel nel "Nuovo" Modo

Entrare nella directory /usr/src.

```
# cd /usr/src
```

Compilare il kernel.

```
# make buildkernel KERNCONF=MYKERNEL
```

Installare il nuovo kernel.

```
# make installkernel KERNCONF=MYKERNEL
```

Il nuovo kernel sarà copiato nella directory /boot/kernel come /boot/kernel/kernel e il kernel precedente sarà copiato in /boot/kernel.old/kernel. Ora, riavviare il sistema e ripartire per usare il nuovo kernel.

5.3 Configurazione del demone Mrouted

Sulle macchine con sistema operativo BSD, ovvero quelle che sono state adibite a router, è stato necessario installare il programma demone *Mrouted* Questo programma ha il compito di gestire il traffico tra i vari gruppi di multicast implementando il protocollo di routing DVMRP, visto in precedenza. Il programma sopraccitato può essere installato su un sistema operativo basato su kernel Linux compilando e installando i sorgenti manualmente, oppure, come nel nostro caso è stato possibile installarlo grazie all'utiliti presente sui sistemi BSD che gestisce i *port*.
La struttura delle tabelle di routine del demone *Mrouted* sono nella forma di pruned brodcast delivery tree, ovvero, contiene solo l'informazione riguardanti quali subnet sono presenti degli host che fanno parte di un determinato gruppo di multicast. In altre parole, ogni router determina quali delle sue interfacce sono incluse nello *Shortest Path Tree*, in questo modo il programma può decidere se un determinato pacchetto multicast debba essere inoltrato su una sua sottorete oppure no. Senza questa caratteristica i pacchetti multicast saturerebbero in breve tempo la banda della rete.
Per supportare il traffico multicast attraverso sottoreti che non implementano il multicast, il protocollo DVMRP fornisce il meccanismo di tunneling, visto in precedenza. Tale opzione può essere configurata tra macchine su cui è installato il programma *Mrouted* attraverso il seguente comando:

```
tunnel local-addr remote-addr [metric m] [threshold t] [rate_limit b]
       [boundary (boundary-name|scoped-addr/mask-len)]
```

Questo comando può anche essere usato per associare un valore di *metric* o di threshold al tunnel, per dare un peso al percorso che si crea. Il *local-addr* può essere espresso come il nome dell'interfaccia, come ad esempio rl0; il *remote-addr* può essere espresso come un host name, ma solo se l'host name in questione ha associato un solo indirizzo IP. Prima che un tunnel possa essere usato, questo deve essere configurato nei file di configurazione del programma Mrouted, per ogni Mrouter partecipanti al tunnel.
Quando il demone Mrouted viene avviato, automaticamente legge il file ASCII di configurazione che si trova nella cartella `/etc/mrouted.conf`. Si possono cambiare le configurazioni di default specificando un file alternativo, durante la fase di avvio. Se vengono apportate delle modifiche al file mrouted.conf mentre il programma è già in esecuzione è possibile usare il comando `kill`, per arrestarlo e riavviarlo con le modifiche apportate. Di default mrouted si autoconfigura per l'inoltro del traffico multicast su tutte le interfacce attive ad eccezione di quelle di loopback; quindi a meno di particolari configurazioni come ad esempio un tunnel, non si dovrebbe avere bisogno di andare a modificare il file `/etc/mrouted.conf`.

Alcuni comandi di configurazione:

```
phyint local-addr [disable] [metric m] [threshold t] [rate_limit b]
       [boundary (boundary-name|scoped-addr/mask-len)]
       [altnet network/mask-len]
```

Il commando *phyint* può essere usato per disabilitare il routing multicast su un' interfaccia fisica identificata da un indirizzo IP locale, o per associare un valore non-default metric o threshold ad una specifica interfaccia.

```
cache_lifetime ct
```

Il valore di cache_lifetime determina l'ammontare di tempo una routing multicast rimane nella cache prima che scada il valore di time-out. Questo valore è specificato in secondi e può essere compreso tra 300 (5 minuti) ed i 86400 (24 ore) secondi. Il valore di default è 300.

```
pruning off/on
```

Il comando per disattivare il pruning serve per configurare il programma Mrouted come un router "non-pruning". Quando il pruning è off i pacchetti multicast sono inoltrati su tutte le subnet appartenenti all'albero di routine, anche se queste non hanno partecipanti ad alcun gruppo di multicast. Questa modalità è utilizzata solo in fase di test. Di default l'opzione pruning è settata su on.

Mrouted viene avviato con il seguente comando:

```
/etc/mrouted [-p] [-c config_file] [-d debug_level]
```

L'opzione –p disabilita il pruning sovrascrivendo la configurazione settata nel file di configurazione.
L'opzone –c sostituisce il file di configurazione di default con uno diverso, specificandolo di seguito.
L'opzione –d *debug_level* specifica il livello di debug. Questo può essere tra 0 e 3. In conseguenza al livello di debug scelto i messaggi di log verranno riportati in un file che generalmente si trova nella directory `/var/adm/syslog`.
Per verificare che il demone Mrouted sta effettivamente lavorando, si può controllare lo stato dei processi con il comando

```
ps -ef | grep mrouted
```

Esistono tre tipi di routing table associate ad mrouted: la *Virtual Interface Table*, la *Multicast Routing Table* e la *Multicast Routing Cache Table*. La *Virtual Interface Table* contiene informazioni sia per interfacce virtuali che per interfacce impiegate per i tunnel, il numero di pacchetti in entrata ed in uscita sulle rispettive interfacce ed il valore delle configurazioni del *metric* e del threshold. La *Multicast Routing Table* contiene informazioni riguardo alla connettivita per ogni subnet da cui i pacchetti multicast vengono originati. La *Multicast Routing Cache Table* è una copia della forwarding cache table presente nel kernel.
Anche se non sono stati usati nelle prove in laboratorio, esistono tool per il supporto di routine multicast:
mrinfo: Il comando mrinfo si interroga un determinato router multicast con un messaggio IGMP (Internet Group Management Protocol). La risposta alla query

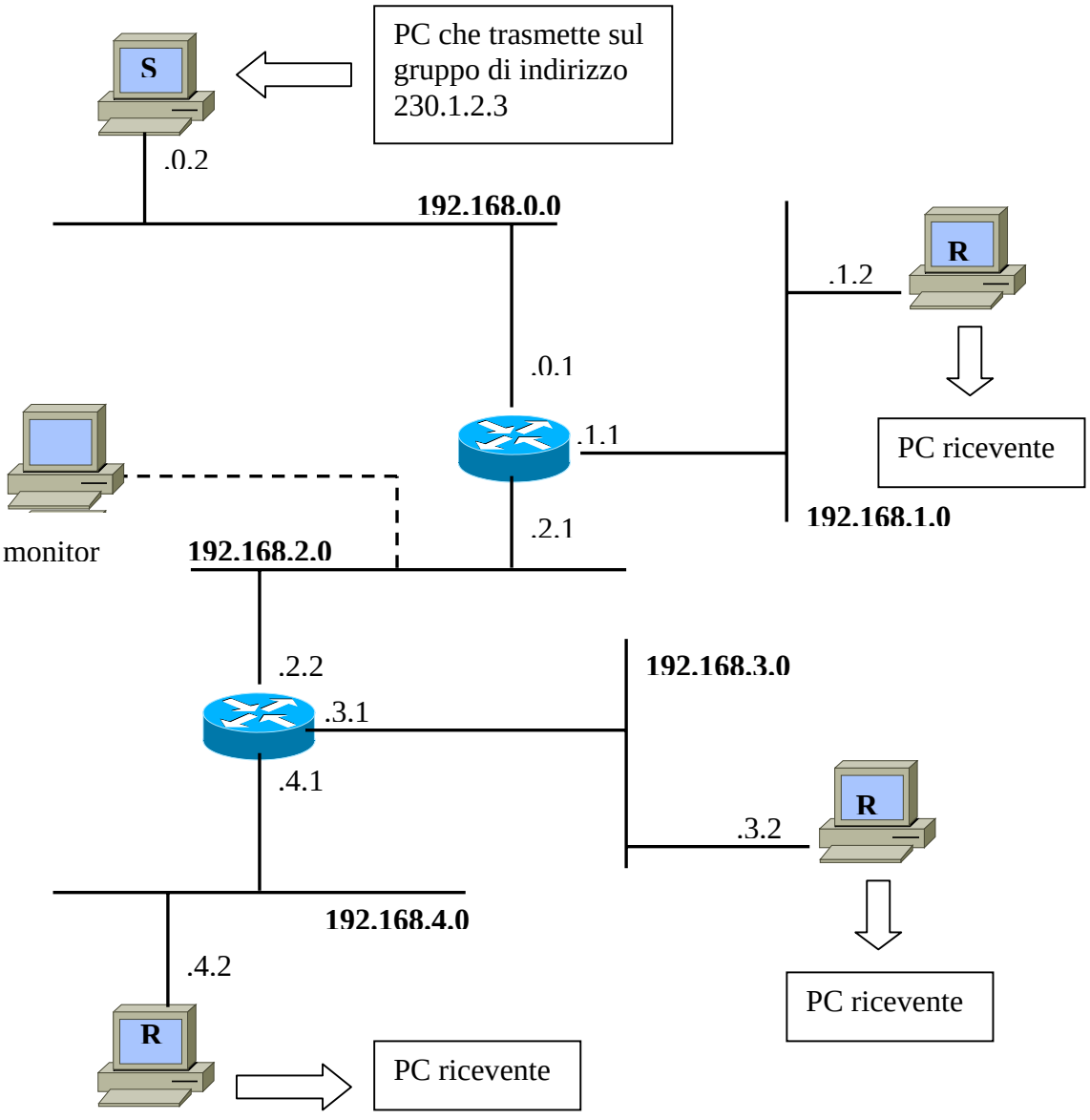
contiene un numero di versione, l'elenco delle interfacce e dei router adiacenti di ciascuna interfaccia, la metrica, le soglie TTL (Time to Live) e i flag. La sintassi del comando mrinfo è:

```
mrinfo [-n] [ -i indirizzo ] [ -rconteggio_tentativi ] [ -t
conteggio_timeout ] router_multicast
```

map-mbone: recupera informazioni di connessione sul routing multicast e visualizza la "*connection map*" su standard output.

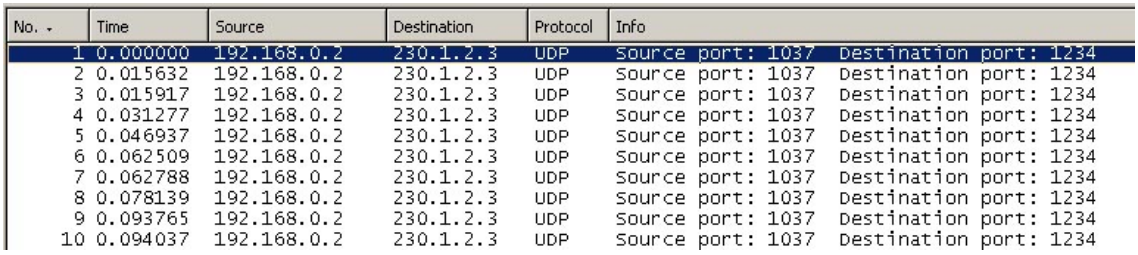
netstat: è usato per recuperare statistiche sulla rete locale e per visualizzare le multicast routine tables.

5.4 Cattura e analisi del traffico multicast



Schema della rete realizzata il laboratorio

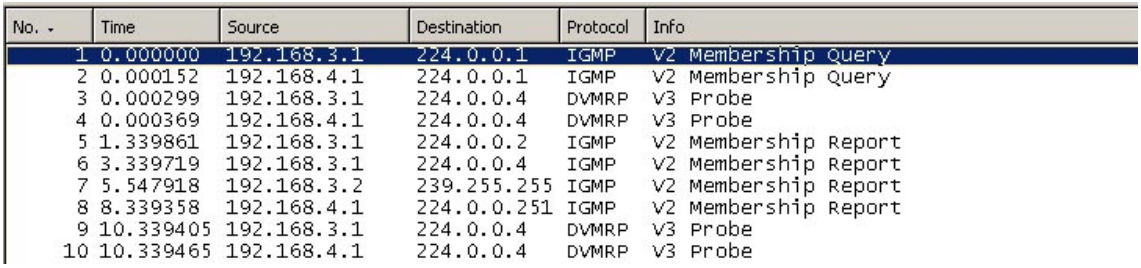
La trasmissione avviene (come mostrato in figura) dal PC con indirizzo 192.168.0.2. I riceventi sono i 3 PC rispettivamente con indirizzo 192.168.1.2, 192.168.3.2, 192.168.4.2. Tramite il programma VLCmedia, il PC sender trasmette sul gruppo di multicast 230.1.2.3 (indirizzo scelto casualmente tra quelli disponibili di classe D). Questo che segue è il traffico catturato sul PC sender quando questo inizia a trasmette sul gruppo:



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
2	0.015632	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
3	0.015917	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
4	0.031277	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
5	0.046937	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
6	0.062509	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
7	0.062788	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
8	0.078139	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
9	0.093765	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234
10	0.094037	192.168.0.2	230.1.2.3	UDP	Source port: 1037 Destination port: 1234

Traffico sul PC sender

Dal traffico catturato si può vedere che il PC con indirizzo IP 192.168.0.2 trasmette sul gruppo di multicast di indirizzo 230.1.2.3, in particolare i pacchetti sono nel formato del protocollo UDP. La ragione per cui i datagrammi trasmessi sono in questo formato, è perché il protocollo UDP non richiede procedure di connessione al contrario del TCP, la trasmissione risulta quindi più snella e veloce.
Prima di analizzare il traffico sui PC destinatari durante la ricezione è opportuno vedere quali pacchetti i router inviano a queste macchine quando non fanno ancora parte del gruppo di multicast, in modo da vedere come il demone Mrouted implementa il DVMRP. Qui di seguito viene riportata la cattura del traffico sul PC destinatario di indirizzo 192.168.3.2.

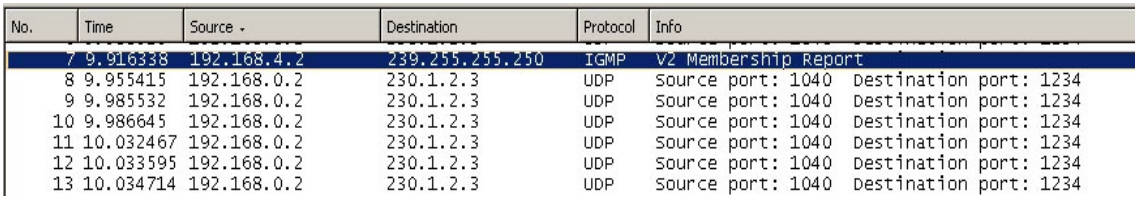


No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.1	224.0.0.1	IGMP	V2 Membership Query
2	0.000152	192.168.4.1	224.0.0.1	IGMP	V2 Membership Query
3	0.000299	192.168.3.1	224.0.0.4	DVMRP	V3 Probe
4	0.000369	192.168.4.1	224.0.0.4	DVMRP	V3 Probe
5	1.339861	192.168.3.1	224.0.0.2	IGMP	V2 Membership Report
6	3.339719	192.168.3.1	224.0.0.4	IGMP	V2 Membership Report
7	5.547918	192.168.3.2	239.255.255	IGMP	V2 Membership Report
8	8.339358	192.168.4.1	224.0.0.251	IGMP	V2 Membership Report
9	10.339405	192.168.3.1	224.0.0.4	DVMRP	V3 Probe
10	10.339465	192.168.4.1	224.0.0.4	DVMRP	V3 Probe

Traffico su un PC quando non fa ancora parte del gruppo di multicast

Verso il destinatario arriva un messaggio di Membership query tramite il protocollo IGMP dai router ad esso adiacenti. Tale messaggio serve al router per chiedere

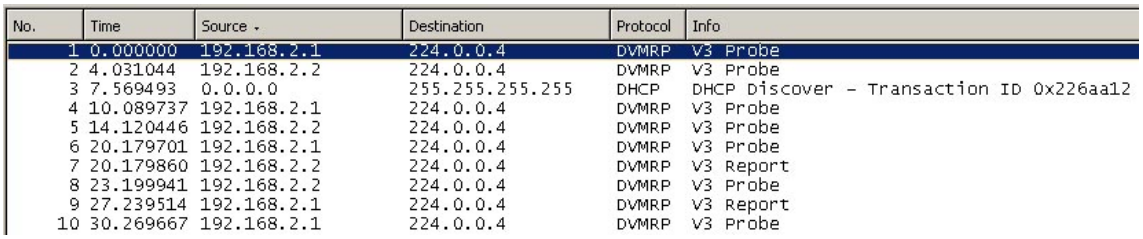
periodicamente il gruppo di multicast al quale fanno parte gli host ad esso collegati. Questo messaggio viene inviato all'indirizzo 224.0.0.1, un indirizzo riservato che rappresenta tutti gli host di una determinata sottorete. Il destinatario per il momento non richiede la partecipazione al gruppo, quindi non risponde al messaggio di *Membership query*. Il resto del traffico riguarda i router, ma lo esamineremo successivamente.
Ora che abbiamo visto come è il traffico multicast senza nessun partecipante al gruppo, andremo a vedere invece cosa avviene quando un PC entra a far parte del gruppo di multicast su cui si sta trasmettendo; ci si aspetta che il PC interessato ad entrare nel gruppo risponda al messaggio di Membership query inviatogli da un router. La cattura del traffico mostrata è quella riguardane sempre il PC con indirizzo IP 192.168.3.2, dal momento che sugli altri due riceventi il risultato è il medesimo.



No.	Time	Source	Destination	Protocol	Info
7	9.916338	192.168.4.2	239.255.255.250	IGMP	V2 Membership Report
8	9.955415	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
9	9.985532	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
10	9.986645	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
11	10.032467	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
12	10.033595	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
13	10.034714	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234

Traffico su un PC destinatario quando entra a far parte di un gruppo di Multicast

Ora i PC destinatari si mettono in ricezione sul gruppo con indirizzo 230.1.2.3. Dopo avere ricevuto il messaggio di *Membership query* dal router, ecco che questa volta risponde a tale messaggio con un *Membership report*, il router, grazie al protocollo IGMP ha così modo di sapere che su una delle sue sottoreti c'è almeno un partecipante al gruppo. L' informazione raccolta serve per aggiornare l'albero di instradamento.
Il traffico scambiato dai router, catturato dal PC che giace fisicamente sulla rete 192.168.2.0, ma che non prende parte alla trasmissione multicast, mostra come i 2 router si scambino messaggi attraverso il protocollo DVMRP, per ottenere reciprocamente informazioni riguardanti lo stato del proprio vicino, in modo da rendere efficiente l'eventuale cambio della topologia dell'albero che si è costruito.
Prima viene mostrata la cattura fatta quando non vi sono ancora partecipanti al gruppo:



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.1	224.0.0.4	DVMRP	V3 Probe
2	4.031044	192.168.2.2	224.0.0.4	DVMRP	V3 Probe
3	7.569493	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x226aa12
4	10.089737	192.168.2.1	224.0.0.4	DVMRP	V3 Probe
5	14.120446	192.168.2.2	224.0.0.4	DVMRP	V3 Probe
6	20.179701	192.168.2.1	224.0.0.4	DVMRP	V3 Probe
7	20.179860	192.168.2.2	224.0.0.4	DVMRP	V3 Report
8	23.199941	192.168.2.2	224.0.0.4	DVMRP	V3 Probe
9	27.239514	192.168.2.1	224.0.0.4	DVMRP	V3 Report
10	30.269667	192.168.2.1	224.0.0.4	DVMRP	V3 Probe

Traffico tra i due router quando non ci sono partecipanti al gruppo di Multicast

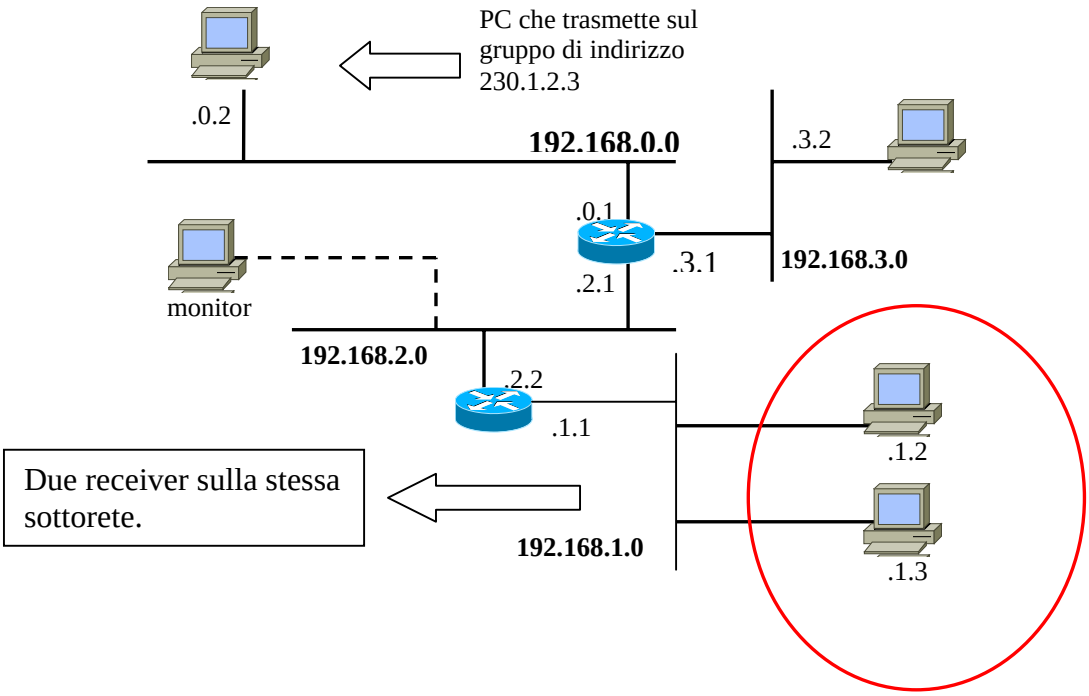
I neighbor router possono essere scoperti in modo dinamico inviando periodicamente messaggi di *probe* su tutte le interfacce attive abilitate per il traffico multicast. Tali messaggi vengono scambiati ad intervalli regolari tra i router che implementano il protocollo DVMRP. Ogni messaggio di *probe* contiene la lista di tutti i neighbor router

da cui è stata ricevuta una risposta, in questo modo ogni router è sicuro di essere in contatto con qualunque altro.
Ecco invece il traffico tra i router quando i PC receiver prendono parte al gruppo di Multicast:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.2	224.0.0.4	DVMRP	V3 Probe
2	3.022744	192.168.2.1	224.0.0.4	DVMRP	V3 Report
3	3.122852	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xa8570473
4	6.053646	192.168.2.1	224.0.0.4	DVMRP	V3 Probe
5	9.859606	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
6	9.906447	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
7	9.953323	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
8	9.984564	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
9	10.031461	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
10	10.031669	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
11	10.078321	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
12	10.089153	192.168.2.2	224.0.0.4	DVMRP	V3 Probe
13	10.156452	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
14	10.218971	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234
15	10.281474	192.168.0.2	230.1.2.3	UDP	Source port: 1040 Destination port: 1234

Traffico tra i due router quando ci sono partecipanti al gruppo di Multicast

Ora il traffico è composto sia dai messaggi di probe, come nella precedente cattura, am anche dai pacchetti UDP inviati dal PC sender. Da notare che nel caso in cui non vi fossero partecipanti al gruppo il traffico multicast si riduce ai messaggi di servizio del protocollo DVMRP; in altre parole la trasmissione avviene solo nel caso in cui ci siano effettivamente dei destinatari del gruppo di multicast, in questo modo la rete non rischia di congestionarsi nel caso in cui molte macchine trasmettessero datagrammi multicast, infatti questi pacchetti vengono inoltrati soltato se c'è effettiva necessità.
Vediamo ora cosa succede quando un PC receiver lascia per ultimo nella sua sottorete il gruppo di Multicast. Per questa cattura, la rete è stata modificata in questo modo:



Schema di rete modificato

Entrambi i PC stanno ricevendo dal medesimo indirizzo di multicast. Ad un certo punto il PC con indirizzo IP 192.168.1.3 lascia il gruppo effettuando una procedura di *leave*. Questo non comporta alcuna modifica nel traffico che viene scambiato tra host e router, infatti il PC con IP 192.168.1.2 sta ancora ricevendo. Quando invece anche il PC con indirizzo 192.168.1.2 effettua il *leave*, ecco che invia un messaggio di *prune* al suo router adiacente, come mostrato in figura.

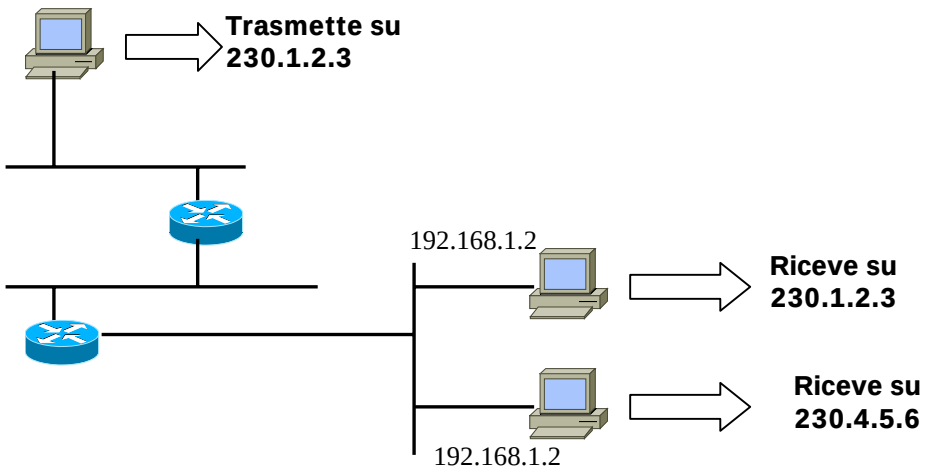
2363	41.310929	192.168.0.2	230.1.2.3	UDP	Source port: 1220 Destination port: 1234
2364	41.326541	192.168.0.2	230.1.2.3	UDP	Source port: 1220 Destination port: 1234
2365	41.342183	192.168.0.2	230.1.2.3	UDP	Source port: 1220 Destination port: 1234
2366	41.373435	192.168.0.2	230.1.2.3	UDP	Source port: 1220 Destination port: 1234
2367	41.389046	192.168.0.2	230.1.2.3	UDP	Source port: 1220 Destination port: 1234
2368	41.404676	192.168.0.2	230.1.2.3	UDP	Source port: 1220 Destination port: 1234
2369	41.410244	192.168.1.2	192.168.1.1	DVMRP	V3 Prune
2370	44.436255	192.168.1.1	224.0.0.4	DVMRP	V3 Probe
2371	45.440313	192.168.1.2	224.0.0.4	DVMRP	V3 Report

Messaggio di prune inviato dall'ultimo sender della sottorete verso il router

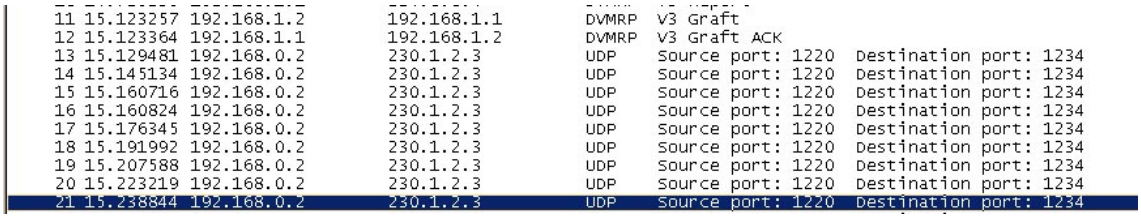
Come previsto nella catura si può vedere un messaggio di *prune*, per avvisare il router che anche l'ultimo receiver della sottorete ha smesso di trasmettere, quindi come prevede il *Troncate Riverse Path Tree* il traffico non verrà più inoltrato verso la rete 192.168.1.0.

5.5 Analisi del traffico di situazioni di transitorio

Un aspetto importane dell'analisi del traffico riguarda l'analisi di questo in situazioni di transitorio. E' interessante vedere il comportamento del demone Mrouted nell'istante in cui più receiver entrano a far parte di più gruppi di multicast differenti, contemporaneamente. In una situazione di questo tipo ci si potrebbe aspettare che il traffico venga inoltrato a tutte le macchine che fanno parte di un qualsiasi gruppo multicast, in quanto l' Mrouter potrebbe non essere ancora a conoscenza dell'appartenenza ai vari gruppi. Questa situazione si risolverebbe al successivo invio di membership query da parte dell' host interessato.
Per generare una situazione di questo tipo la rete realizzata è la seguente:



I due PC receiver si mettono in ascolto su due gruppi di multicast diversi: 230.1.2.3 e 230.4.5.6, mentre il PC sender trasmette soltanto sul gruppo di indirizzo 230.1.2.3
Il risultato della cattura, in questo caso non è come ci si aspetta:



11	15.123257	192.168.1.2	192.168.1.1	DVMRP	V3 Graft	
12	15.123364	192.168.1.1	192.168.1.2	DVMRP	V3 Graft ACK	
13	15.129481	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
14	15.145134	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
15	15.160716	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
16	15.160824	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
17	15.176345	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
18	15.191992	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
19	15.207588	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
20	15.223219	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234
21	15.238844	192.168.0.2	230.1.2.3	UDP	Source port: 1220	Destination port: 1234

Traffico generato in situazione di transitorio

Il traffico catturato tra i due router è tutto indirizzato verso l'indirizzo 230.1.2.3. Questo significa che esiste qualche forma di controllo che impedisce ai router di inoltrare pacchetti quando non si è sicuri della corrispondenza tra indirizzo mittente e destinatario.

Capitolo 6

Conclusioni

"A questo mondo,
tutto quello che ha un inizio ha anche una fine".

Dal film Matrix Revolutions

In questo lavoro, abbiamo cercato di fare il punto sul concetto di comunicazione multicast, mettendo in evidenza gli aspetti che la differenziano e che la accomunano con la comunicazione di tipo unicast. Abbiamo cercato di fare chiarezza sulle tecnologie software che implementa il multicast spiegando prima il protocollo IGMP (***I****nternet Group* ***M****anagement* ***P****rotocol*), e poi i principali protocolli di routing, che in questo momento vengono usati in via sperimentale. Ogni protocollo di routing ha i suoi pro e contro, e noi li abbiamo elencati, indicando quando è meglio utilizzarne uno piuttosto che un altro. La nostra concentrazione si è poi focalizzata sul routing multicat basato sul protocollo DVMRP. Per verificare l'effettiva efficacia di questo protocollo e per poter toccare con mano quanto esposto nei capitoli 3 , 4 e 5 abbiamo realizzato una rete privata, composta da più sottoreti, interconnesse tra di loro da due router Multicast (Mrouter). Questo lavoro ci ha permesso di analizzare il traffico generato, al fine di provare la veridicità della parte teorica. Le prove pratiche in laboratorio ci hanno anche mostrato come sia semplice realizzare una trasmissione di questo tipo, sfruttando parte delle funzionalità già presenti tra quelle che regolano il traffico unicast.
Molti aspetti interessanti sono stati trascurati o appena accennati, come ad esempio l'accurata analisi delle tecnologie che implementano il Multicast, visto che in questi ultimi tempi questa tecnologia sta prendendo sempre più piede e trova sempre più applicazioni nella vita quotidiana. Un altro aspetto che sarebbe stato interessante sviluppare riguarda l'analisi di routing dinamico con l' impiego del protocollo PIM, che assieme al protocollo DVMRP è quello attualmente più usato. Infine un tema di grande attualità ed interesse riguarda la rete GARR, ovvero una rete utilizzata per lo più in ambito universitario, basata sul sistema di comunicazione multicast, impiegata per videoconferenze, videolezini, ecc.
Vista l'estrema volatilità della materia trattata questo lavoro è probabilmente destinato a diventare obsoleto a breve. Ciononostante, al di là dell'analisi del protocollo trattato (il DVMRP) riteniamo che il messaggio di fondo rimanga comunque valido: la comunicazione multicast sta prendendo sempre più piede, essa rappresenta il futuro per quanto riguarda applicazioni "uno a molti", ovvero applicazioni per le quali la solita

trasmissione Unicast si rivelerebbe inefficace e troppo pesante e dispendiosa in termini di banda.

Bibliografia

G. Malkin. RIP Version 2 Protocol Analysis. RFC 1387, gennaio 1993. URL http://www.ietf.org/rfc/rfc1387.txt?number=1387

B. Whetten, Talarian, L. Vicisano, Cisco, R. Kermode, Motorola, M. Handley, CIRI 9, S. Floyd, ACIRI, M. Luby. Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfert. RFC 3048, gennaio 2001.
URL http://www.ietf.org/rfc/rfc2163.txt?number=2163

D. Estrin, USC, D. Farinacei, CISCO, A. Helmy, USC, D. Thaler, UMICH, S. Deering, XEROX, M. Handley, UCL, V. Jacobson, LBL, C. Liu, USCP. Sharma,USC, L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. Giugno 1997 RFC 2117. URL http://www.ietf.org/rfc/rfc2117.txt

D. Estrin, USC, D. Farinacei, CISCO, A. Helmy, USC, D. Thaler, UMICH, S. Deering, XEROX, M. Handley, UCL, V. Jacobson, LBL, C. Liu, USCP. Sharma,USC, L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. RFC 2362. giugno 1998 URL http://www.ietf.org/rfc/rfc2362.txt

D. Provan, Novell, Inc. Tunneling IPX Traffic through IP Networks RFC 1234. giugno 1991 URL http://www.ietf.org/rfc/rfc1234.txt

D. Thaler, Microsoft, M. Handley, ACIRI, D. Estrin, ISI. The Internet Multicast Address Allocation Architecture RFC 2908 settembre 2000
URL http://www.ietf.org/rfc/rfc2908.txt

S. E. Deering, D. R. Cheriton, Stanford University Host Groups: A Multicast Extension to the Internet Protocol RFC 966. dicembre 1985.
URL http://www.ietf.org/rfc/rfc0966.txt

D. Waitzman, C. Partridge, BBN STC, S. Deering, Stanford University. Distance Vector Multicast Routing Protocol. RFC 1075 novembre 1988
URL http://www.ietf.org/rfc/rfc1075.txt

J. Moy, Proteon, Inc. MOSPF: Analysis and Experience RFC 1585. marzo 1994 URL http://www.ietf.org/rfc/rfc1585.txt

A. Ballardie, Consultant, Core Based Trees (CBT) Multicast Routing Architecture RFC 2201 settembre 1997 URL http://www.ietf.org/rfc/rfc2201.txt

A. Ballardie, Consultant. Core Based Trees (CBT version 2) Multicast Routing RFC 2189 settembre 1997 URL http://www.ietf.org/rfc/rfc2189.txt

D. Meyer, University of Oregon. Administratively Scoped IP Multicast RFC 2365 luglio 1998 URL http://www.ietf.org/rfc/rfc2365.txt
D. Thaler, Microsoft, Interoperability Rules for Multicast Routing Protocols RFC 2715. ottobre 1999. URL http://www.ietf.org/rfc/rfc2715.txt

W. Fenner, Xerox PARC. Internet Group Management Protocol, Version 2 RFC 2236. novembre 1997 URL http://www.ietf.org/rfc/rfc2236.txt

Linux Notes URL http://www.dancres.org/bjspj/docs/docs/linux.html

Mbone URL http://telemat.die.unifi.it/book/MBone/mbindex.htm

Building and install a custom kernel URL http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig-building.html

Routing Information Protocol
URL http://it.wikipedia.org/wiki/Routing_Information_Protocol

Rete GARR URL http://www.garr.it/

Indice analitico