Lezione n.13

JXTA: RENDEZVOUS PROTOCOL SRDI

Materiale didattico sulla pagina del corso

Laura Ricci

RENDEZVOUS SUPERPEERS

- Classificazione dei peers
  - Edge Peers
  - Rendez-Vouz Peers
  - Relay Peers

- Rendezvous Peers: peers che si sono resi disponibili ad effettuare

l 'instradamento degli advertisements sulla rete

- Un prerequisito per diventare rendezvous peer per un gruppo è essere membro di quel gruppo

- Al momento del bootstrap un rendez vous peer deve inizializzare la propria RPV (Rendezvous Peer View)

RPV= conoscenza che un rendezvous ha degli altri rendezvous dello stesso gruppo

SEEDING RENDEZVOUS

- Seeding Rendezvous
  - Rendez vous che sono stabilmente attivi sulla rete
  - Un rendezvous peer è configurato mediante una lista predefinita di seeding rendezvous
  - insieme di rendezvous che permettono il bootstrap di un rendezvous peer sulla rete

- I seeding rendez vous agiscono da rendezvous solo per NetPeerGroup

- All'inizio il rendezvous conosce solamante un insieme di seeding rendezvous

- dopo il bootstrap, i seeding rendezvous vengono utilizzati come punto di distribuzione di advertisements di rendezvous che si sono connessi alla rete

RENDEZVOUS SUPERPEERS

- E' compito dell'utente definire i rendezvous peer per gruppi diversi dal NetPeerGroup

- Come un peer P può diventare rendezvous peer:

  - Invocazione esplicita al metodo startRendezvous( )
  - Con Invocazione del metodo setAutoStart(true, period) P dichiara la sua disponibilità a diventare un rendezvous.
  - A seconda del rapporto edge/rendezvous peer presenti nella rete in quel momento, P può assumere o meno la funzione di rendez vous peer
  - Lo stato della rete viene ricontrollato ogni period millisecondi. JXTA quindi decide dinamicamente se 'promuovere' un peer da edge a rendez vous o farlo 'regredire' da rendezvous ad edge peer

RENDEZVOUS SUPERPEERS

- Un peer che è diventato un rendez vous mediante il meccanismo di autostart( ) può ritornare automaticamente alla condizione di edge peer
- di solito JXTA mantiene al massimo 5 rendezvous per ogni gruppo, anche nel caso in cui tutti i partecipanti al gruppo siano rendezvous
- se esistono più di 5 rendezvous nella rendezvous peer view del peer, il peer può decidere di tornare allo stato di edge peer. La decisione è influenzata anche dal numero di edge peers connessi
- I peers serviti da quel rendezvous devono trovare un altro rendezvous a cui connettersi
- Per connettersi ad altri rendez vous peers utilizzano i rendezvous advertisements reperiti mediante il rendezvous a cui erano precedentemente connessi

RENDEZVOUS SUPERPEERS

- Quando un peer diventa Rendez-Vous, pubblica un proprio Advertisement, al cui interno sono contenute le informazioni necessarie agli altri peer al fine di stabilire una connessione con esso.
- Formato di un advertisement

```
<jxta:RdvAdvertisement xmlns:jxta="http://jxta.org">
    <RdvGroupId> . . . </RdvGroupId>
    <RdvPeerId> . . . </RdvPeerId>
    <Name> . . . </Name>
    <RdvServiceName> . . . </RdvServiceName>
    <RdvRoute> . . . </RdvRoute>
</jxta:RdvAdvertisement>
```

- Un edge peer che desidera collegarsi ad un rendezvous peer, effettua una ricerca di questi advertisements
- La ricerca avviene a partire dai rendezvous peer già esistenti (all'inizio a partire dai seeding rendezvous)

CONNESSIONE EDGE/RENDEZVOUS PEER

- un edge peer è connesso in ogni istante ad un unico rendesvouz peer
- un edge peer ricerca continuamente (in background) ulteriori rendes vouz peer advertisements e li memorizza nella propria cache locale
- i rendezvous advertisment sono mantenuti nella cache locale del peer al momento della sua disconnessione dalla rete e possono quindi essere ritrovati in esecuzioni successive
- Un edge peer, per connettersi ad un rendezvous peer esegue, nell'ordine, i seguenti passi
  - ricerca nella propria cache locale
  - invia richieste di advertisement sulla rete
  - prova a connettersi ai seeding rendezvous
  - diventa esso stesso un rendezvous peer

CONNESSIONE EDGE/RENDEZVOUS

- La ricerca degli advertisements dei rendezvous peer avviene mediante il discovery service

- Per connettersi ad un rendezvous
  - connectToRendezVous (PeerAdvertisement adv) il parametro indica l'advertisement di un rendez vous scoperto mediante il Discovery Service
  - connectToRendezVous (EndpointAddress addr). Il parametro indica l'indirizzo del rendez vous

- L'invocazione di questi metodi genera un lease request message.

- Lease request message: viene spedito da un peer ad un rendez-vous per chiedere il permesso a collegarsi ad esso, di poterlo usare per propagare i propri messaggi e di essere inseriti nella lista dei peer in grado di ricevere messaggi propagati da altri.

GESTIONE CONNESSIONI AI RENDEZVOUS

- Per disconnettersi da un rendez vous

  voiddisconnectFromRendezVous(ID peerID)

- L'invocazione del metodo genera un lease cancel message
- Per la gestione dei messaggi di connessione/disconnessione è possibile utilizzare un listener, il RendezvousListener.
- Il RendezvousListener definisce un unico metodo

  rendezvousEvent(RendezvousEvent event)

che deve essere implementato e poi registrato presso il rendez-vous service mediante il metodo addListener

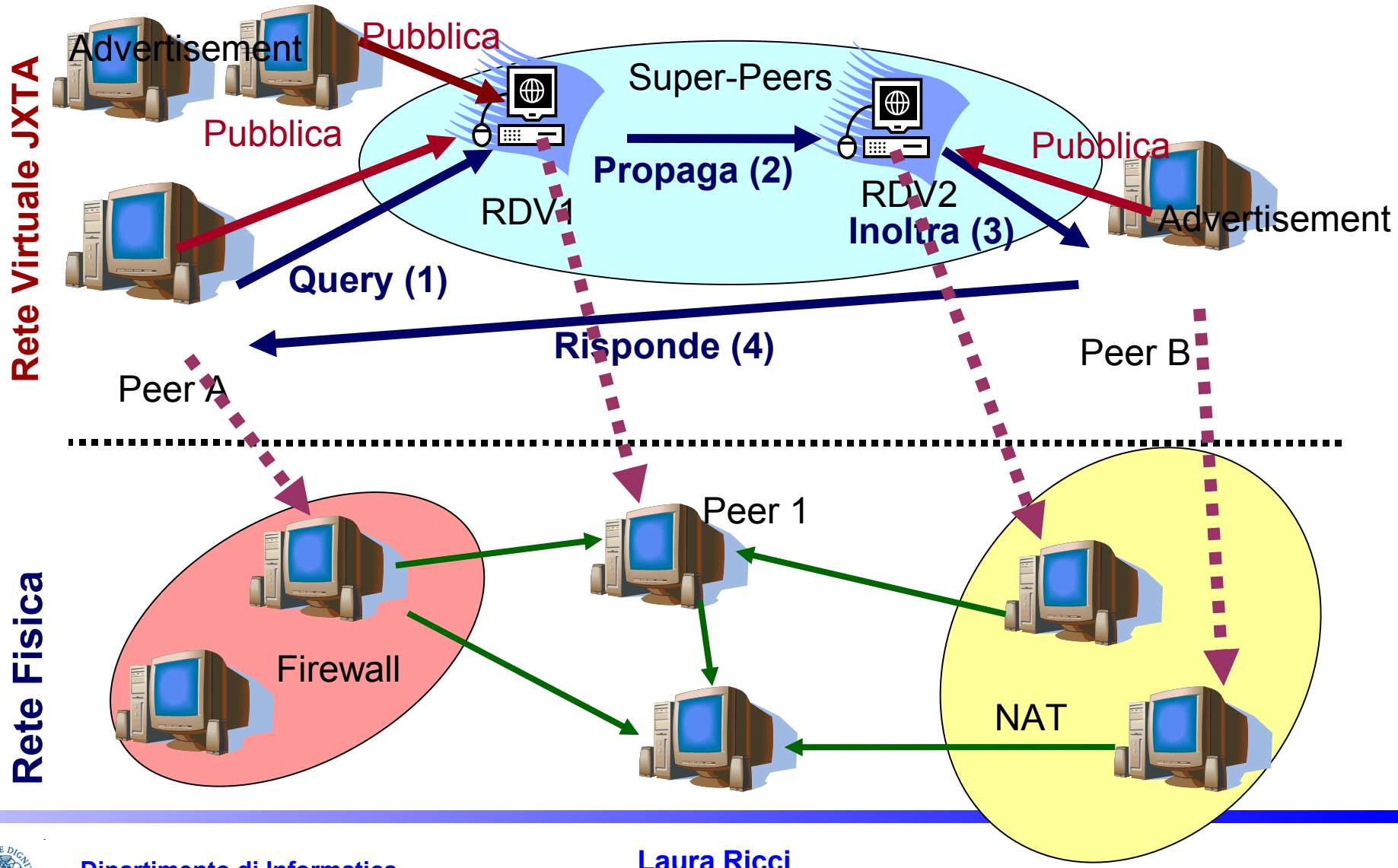
- ogni messaggio di connessione/disconnessione ricevuto dal rendez-vous service provocherà il risveglio del listener, con la notificazione di uno specifico evento.
- previsti diversi possibili eventi: RDVCONNECT, RDVDISCONNECT , RDVFAILED : BECAMERDVOUS, BECAMEEDGE,...

RENDEZ VOUS SUPERPEERS

Funzioni di un Rendez Vouz peer

- memorizza nella propria cache un indice che contiene un insieme di chiavi di advertisements pubblicati in edge peers del gruppo
- gli advertisments sono memorizzati unicamente negli edge peers, non nei rendezvous peers
- gli edge peers inviano al rendez-vouz peer le chiavi degli advertisment pubblicati

UN ESEMPIO


Rete Virtuale JXTA
Advertisement
Pubblica
Pubblica
Super-Peers
Propaga (2)
RDV1
RDV2
Inoltra (3)
Pubblica
Advertisement
Query (1)
Risponde (4)
Peer A
Peer B
Rete Fisica
Peer 1
Firewall
NAT

RENDEZVOUS PEER VIEW

- Ogni rendezvous deve mantenere aggiornata la lista degli altri rendezvous presenti sulla rete
- Global Rendezvous PeerView (GRPV) di un gruppo: insieme di tutti i rendezvous peers di quel gruppo
- La GRPV non viene mantenuta in uno specifico peer del gruppo
- Ogni RVP mantiene una vista locale delle GRPV
- Rendezvous Peer View(RPV) di un peer P: elenco di tutti i rendez-vous conosciuti da P , ordinato per ID crescente.
- Le diverse RPV non vengono mantenute costantemente consistenti (loosely consistent)
- PeerView Protocol cerca di far convergere le diverse viste locali dei diversi peer ad un'unica visione consistente

RENDEZVOUS PEER VIEW

- Periodicamente ogni rendezvous peer estrae dalla propria RPV un numero casuale di peer dalla loro RPV ed invia i loro advertisement a tutti i rendezvous noti

- I rendesvouz inviano anche periodicamente un heartbeat ai rendesvouz vicini (quelli che occupano le posizioni +1 e –1 nella RPV)

- I rendesvouz
  - aggiornano le loro tabelle quando ricevono informazioni dai vicini
  - eliminano dalle proprie tabelle i rendezvous da cui non ricevono heartbeat per più di un certo intervallo di tempo
  - possono inviare un ping ai vicino per verificare se sono connessi

- Le RPV dei superpeer tendono a convergere mediante l'invio di messaggi successivi

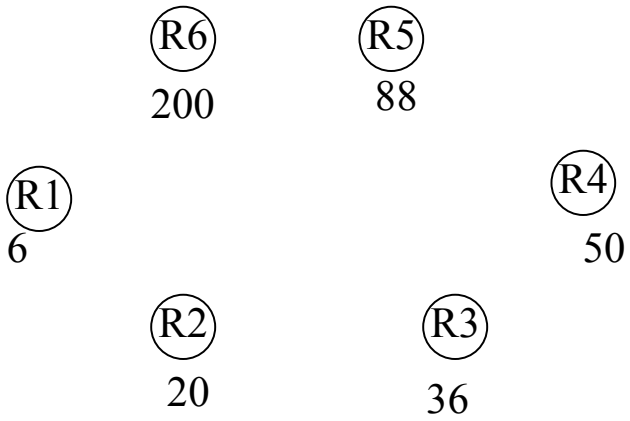
RENDEZVOUS PEER VIEW

- Rendezvous Peer View (RPV)= lista di rendezvous peer conosciuti ordinata in base al loro Peer ID

- ogni rendezvous peer RV mantiene una propria rendezvous peer view che contiene alcuni rendezvous peer che RV ha identificato sulla rete

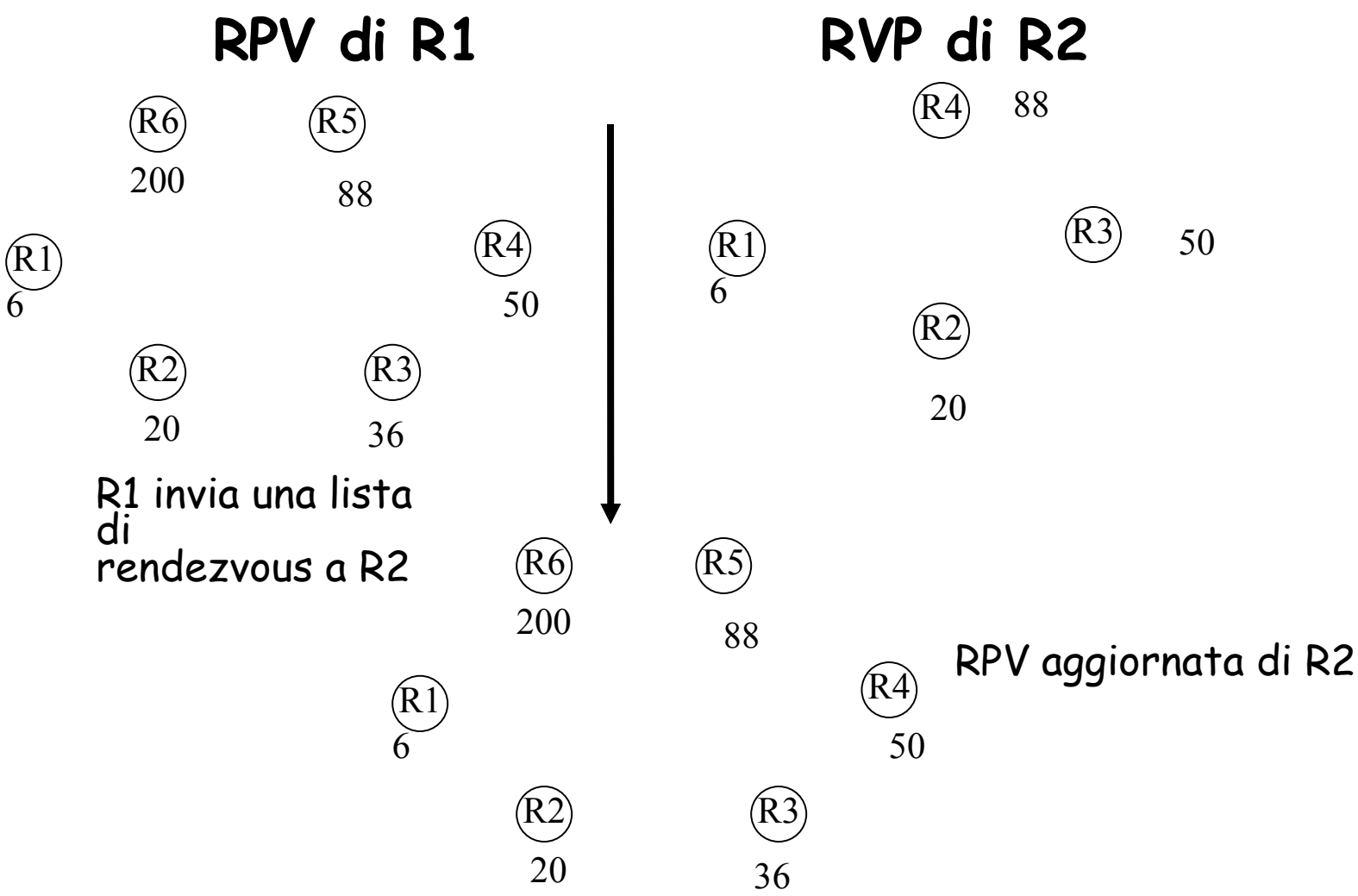
6	R1
20	R2
36	R3
50	R4
88	R5
200	R6

RPV di R2



- Ogni rendezvous nella RPV è identificato da un valore intero che corrisponde alla sua posizione nell RVP

RENDEZ-VOUZ PEER VIEW: CONVERGENZA



RENDEZVOUS PROTOCOL

Ogni peer, quando diventa un rendezvous peer, esegue le seguenti operazioni

- crea e pubblica localmente un rendezvous advertisement A con un lifetime corrispondente al tempo medio di connessione del peer alla rete
- pubblica A su tutti i seeding rendezvous peers
- queste operazioni vengono ripetute quando il ciclo di vita dell'advertisement si esaurisce

Quindi, ad intervalli di tempo regolari, ogni rendez vous peer

- ricerca sui seeding rendezvous peer ulteriori rendesvouz advertisements
- scambia la propria RPV con i rendez vous peer vicini
- invia un ping ai rendezvous peers nella propria RPV. I Rendezvous peers che non rispondono vengono eliminati dalla RPV

RENDEZVOUS PROTOCOL

- i seeding peers introducono un elemento di centralizzazione del protocollo
- i seeding peers devono essere utilizzati prevalentemente in fase di bootstrap
- una volta che un rendezvous peer conosce altri rendezvous, può contattare questi peers direttamente
- durante l'esecuzione della applicazione, i seeding rendezvous peer possono accellerare la convergenza della RPV

USO DELLE RVP

- un edge peer EP identificato da EdgePeerID pubblica un advertisement
- EP invia (push) al proprio rendezvouz RV peer alcune chiavi che identificano l'advertisement
- RV associa, mediante una funzione hash, un identificatore all'advertisement. La funzione hash è applicata alle chiavi dell'advertisement

UTILIZZO DEI RPV

- Quando un peer pubblica un advertisement, ne invia le chiavi al proprio rendez-vous.
- In realtà questo rendezvous procederà anche ad inoltrare l'informazione relativa a quell'advertisement anche ad altri rendez-vous.
- Supponiamo che la funzione hash applicata all'advertisement abbia restituito l' intero n.
- Il rendez vous estrae dalla propria RPV il rendez-vous che occupa la posizione n e vi replica una copia dell'advertisement ricevuto.

DISTRIBUZIONE SPAZIO DEGLI IDENTIFICATORI

Definizione del Mapping M

- M: advertisement $\rightarrow$ Identificatori di RVP
- Esempio: mapping "a la Chord". Un advertisement viene mappato sul rendezvous peer che ha identificatore (posizione nella RPV) uguale all'hash dell'advertisement
- Possibili Inconsistenze:
  - il mapping può restituire valori diversi su rendezvous diversi, poichè le RPV di possono contenere RPV inconsistenti (non contengono gli stessi valori)

UTILIZZO DELLE RVP

- Un edge peer EP
  - pubblica un advertisement
  - lo memorizza nella cache locale
  - invia (push) le chiavi al proprio rendez-vous peer
  - l'invio al proprio rendezvous peer può essere sincrono/asincrono
- il rendez vouz peer
  - calcola la funzione hash H
  - mappa H su uno dei rendezvous peer MRV della rete
  - invia l'indice dell'advertisement a MRV

UTILIZZO DELLE RVP

- Quando un peer ricerca un advertisement , invia al suo rendez vous le chiavi che identificano l'advertisement
- il suo rendez-vous applica la funzione di hash alle chiavi ricevute: ed ottiene un valore di n
- Interroga il rendez-vous che occupa la posizione n nella propria RVP.
- Se la RVP è uguale per tutti, allora l'advertisement trovato sarà quello effettivamente ricercato.
- Altrimenti sono necessari ulteriori passi di ricerca

UTILIZZO DELLE RVP

- E' importante che le diverse RPV convergano allo stesso insieme
- Se un rendez-vous viene rimosso da una RVP locale, o se altri rendez vous si aggiungano alla rete, si possono generare inconsistenze, la funzione hash può portare alla ricerca di un advertisement su un peer che non lo possiede
- Per rendere ovviare a queste inconsistenze
  - si introduce un certo livello di replicazione degli advertisements
  - Limited-Range Walker: tecnica utilizzata per ricercare l'advertisement non solo sul peer individuato dalla funzione hash, ma anche su alcuni peer 'vicini'

LOOSELY CONSISTENT DHT

- meccanismi di memorizzione/ricerca di informazioni in reti P2P
  - basati su DHT(Distributed Hash Tables): ricerca efficiente, alto costo per la gestione dell'indice distribuito
  - basati su flooding (Gnutella): ricerca inefficiente, non richiede la gestione di indici distribuiti

- JXTA 2.x adotta un approccio ibrido, basato sulla definizione di Loosely Consistent DHTs

- la consistenza assoluta delle DHT memorizzate sui diversi rendezvous peer non viene garantita
  - il costo di gestione delle DHT viene così limitato

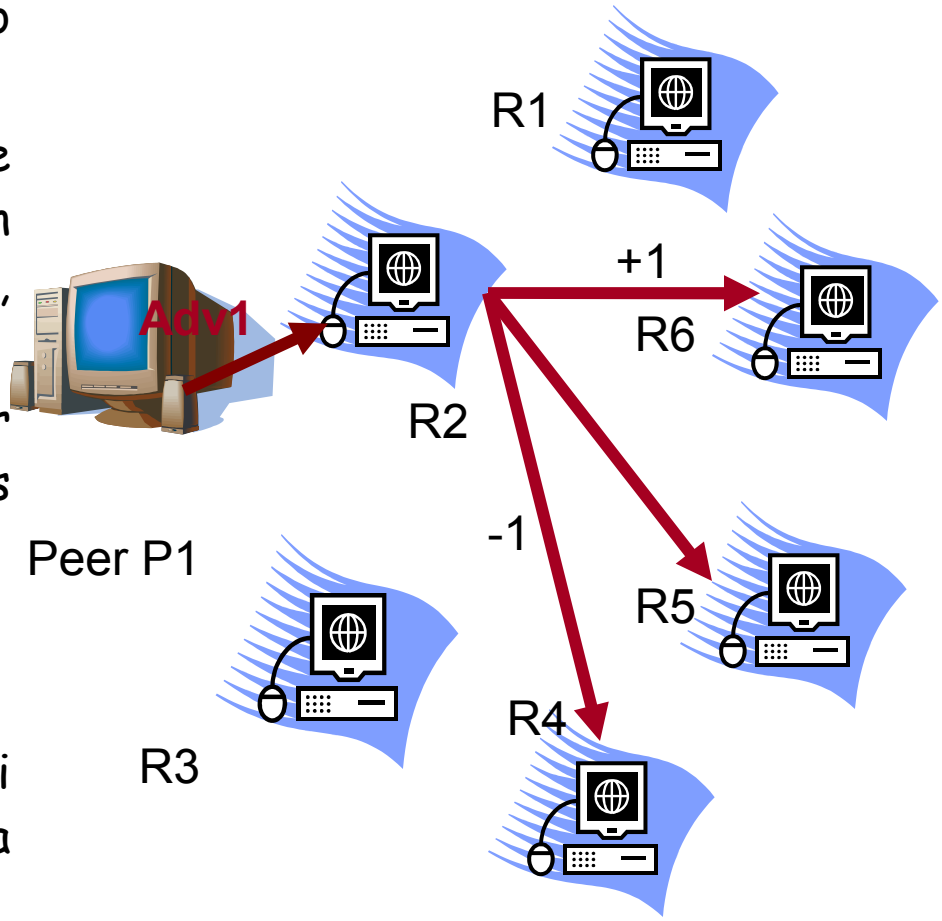
- il servizio Shared-Resource Distributed Index (SRDI) supporta la gestione degli indici

IL SERVIZIO SRDI

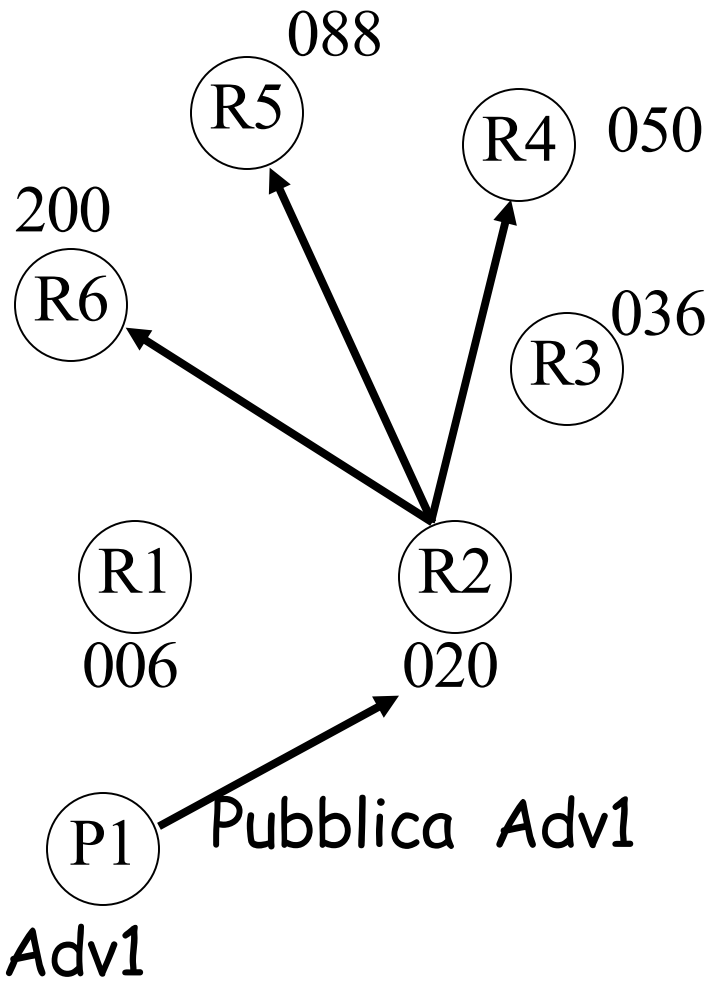
- la ricerca è basata sul servizio SRDI
- l'utente può specificare diversi tipi di queries
  - getRemoteAdvertisements(peerId,Type,attribute,value,threshold)
- l'edge peer invia la query al proprio rendezvous peer
- solo i rendezvous peer sono coinvolti nella propagazione delle queries
- una query viene inviata ad un edge peer P solo se la query individua un advertisement pubblicato da P
- questa strategia riduce in modo significativo il traffico di rete per la ricerca degli advertisements

PUBBLICAZIONE DEGLI ADVERTISEMENTS

- Il peer P1 pubblica un nuovo advertisement Adv1 sul suo rendezvous peer R2 via SRDI
- Ad ogni advertisement viene associato un indice usando un numero predefinito di chiavi (nome, ID,…)
- R2 usa una funzione DHT per associare l'indice ad un rendezvous peer nella sua RPV
- Supponiamo che DHT restituisca R5
- R2 invia l'indice a R5
- L'indice viene replicato sui vicini (nella RPV) di R5 (+1 e –1 nella lista ordinata della RPV)



PUBBLICAZIONE DEGLI ADVERTISEMENTS

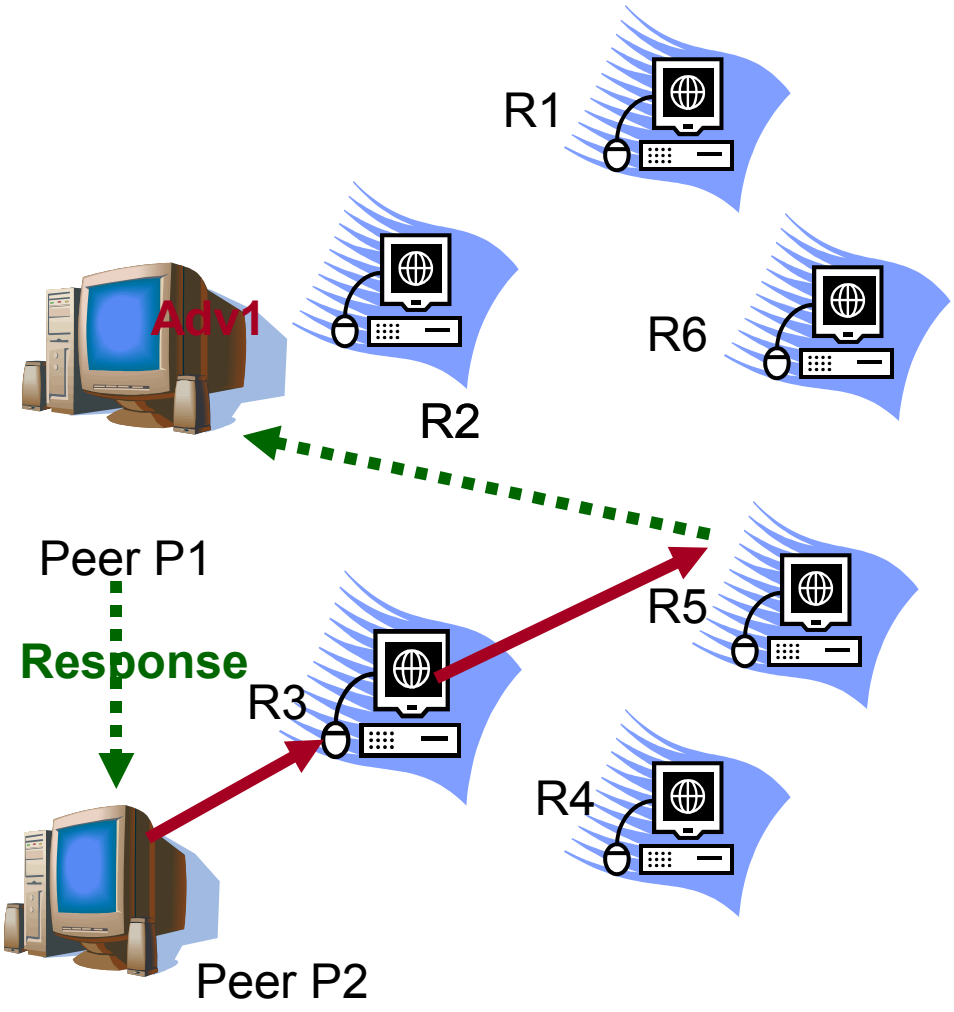


Tipo dell'advertisement : peer
Nome attributo : name
Valore attributo : P1

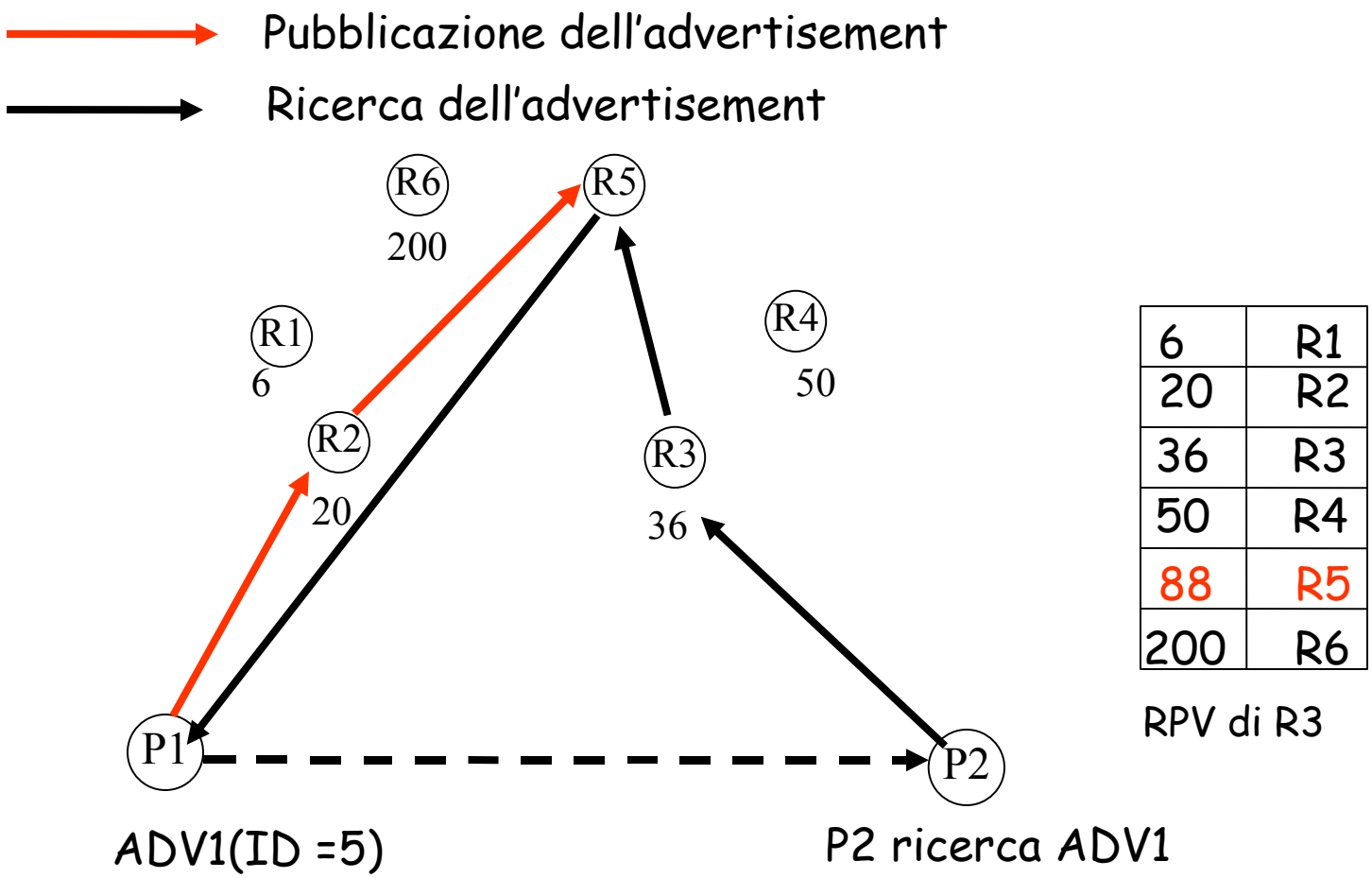
- H(peer,name,P1) = 7
- DHT(7)= R5
- L'indice viene
  - pubblicato su R5 =
  - replicato su R4 e su R6

RPV CONSISTENTI:PROPOGAZIONE DELLE QUERIES

- supponiamo che l'edge peer P2 stia ricercando l'advertisement Adv1
- P2 invia una query al suo rendezvous peer R3
- il servizio SRDI su R3 calcola la chiave usando la RPV di R3
- se le RPV su R2 e su R3 sono consistenti (contengono gli stessi rendez vous), la funzione DHT restituisce lo stesso rendezvous R5
- R3 invia la richiesta ad R5 che la inoltra a P1
- P1 invia la risposta direttamente a P2
- condizione per il corretto funzionamento: consistenza delle RPV di R2 e di R3



RPV CONSISTENTI:PROPAGAZIONE DELLE QUERIES

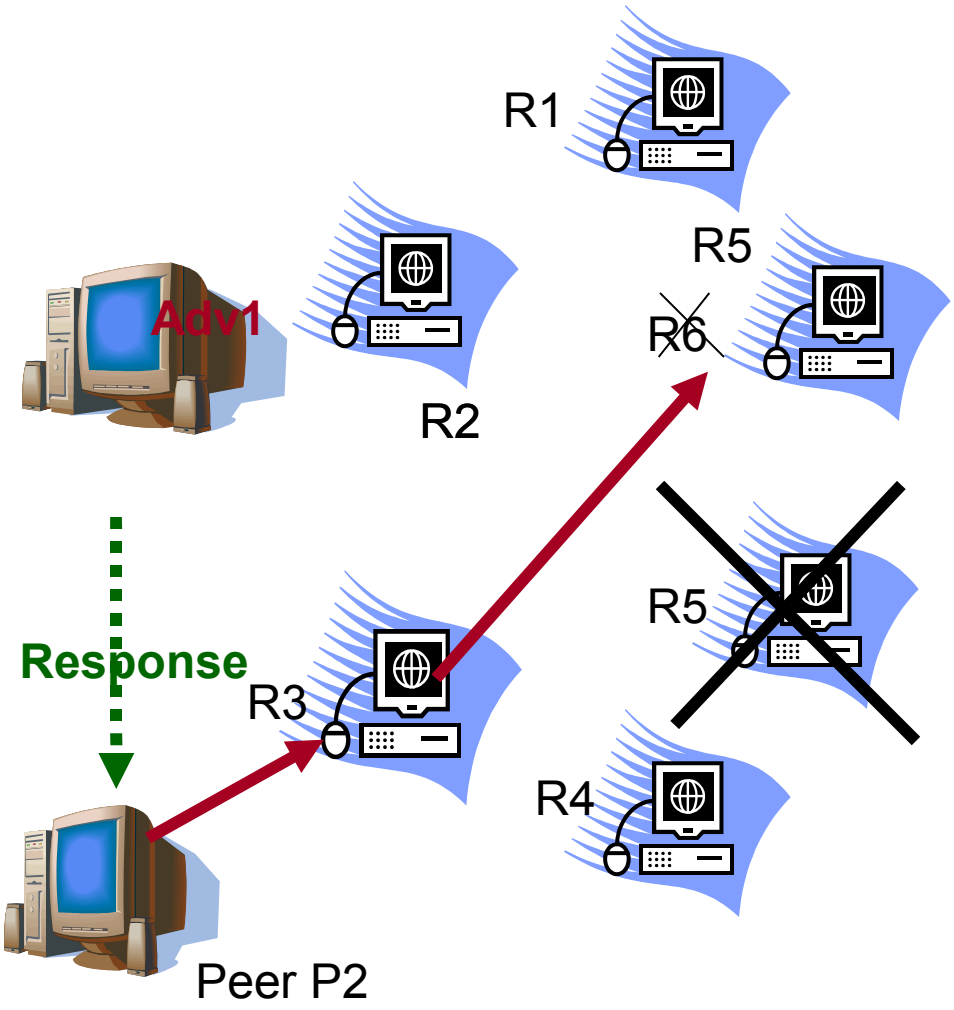


6	R1
20	R2
36	R3
50	R4
88	R5
200	R6

RPV di R3

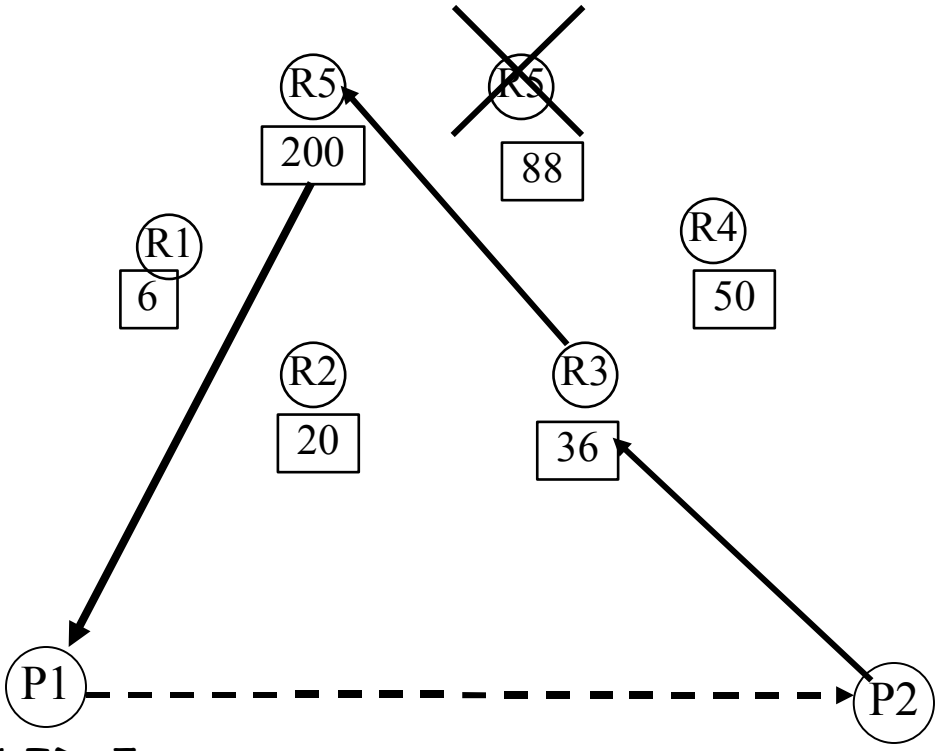
RPV INCOSISTENTI:PROPOGAZIONE DELLE QUERIES

- R5 si disconnette : R3 aggiorna la propria RPV
- R3 scopre che R5 si è disconnesso quando gli invia il messaggio e non riceve risposta
- R3 sposta di una posizione le entrate della propria RPV
- il numero di entrate nella RPV di R3 è ora 5
- Dopo aver ricevuto la richiesta di P2, R3 calcola la funzione M che restituisce R5 (il vecchio R6)
- Poichè l'indice è stato pubblicato anche su R6, la ricerca da esito positivo



RPV INCONSISTENTI:PROPAGAZIONE DELLE QUERIES

P2 è connesso ad R3



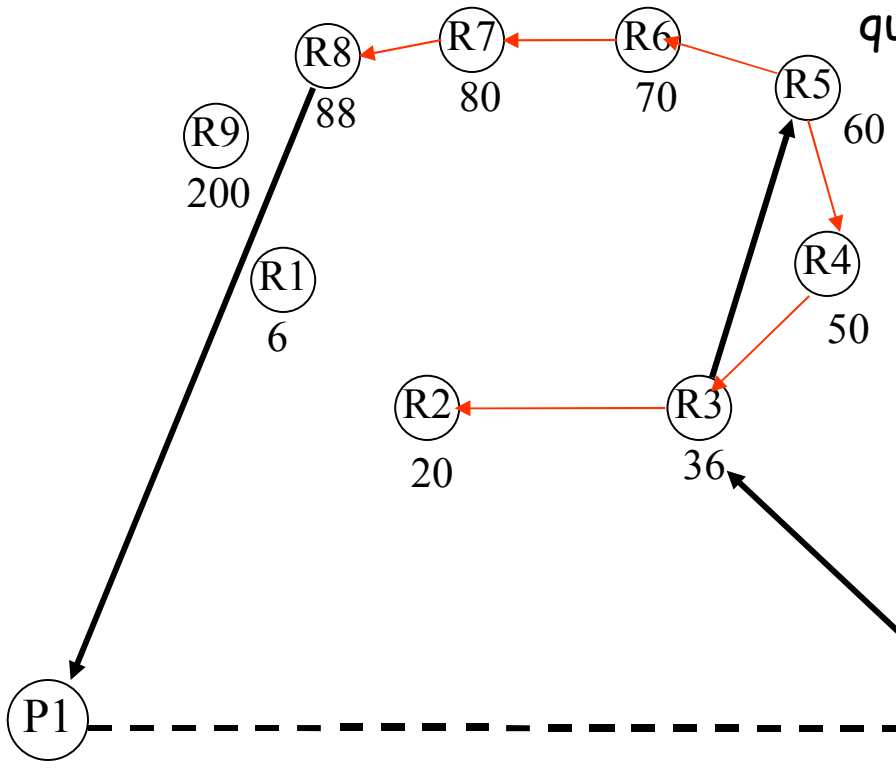
6	R1
20	R2
36	R3
50	R4
200	R5

RPV di R3

- ADV1 ID=5
- l'advertisement è memorizzato sul 'vecchio' R5 (identificatore 88)
- viene ricercato sul 'nuovo' R5 (identificatore 200)

RPV CONSISTENTI: LIMITED RANGE WALKER

R3 effettua uno shift delle entrate della propria RPV quando inserisce nuovi rendezvous



ADV1(ID =5)

6	R1
20	R2
36	R3
50	R4
88	R5
200	R6

RPV di R3

P2 ricerca ADV1 ID=5
P2 è connesso ad R3

6	R1
20	R2
36	R3
50	R4
60	R5
70	R6
80	R7
88	R8
200	R9

RPV aggiornata di R3

LIMITED RANGE WALKER

- Limited Range Walker = utilizzato per "camminare" sui rendezvous vicini al rendenzvous restituito dal mapping M (rendez vous target)
- se l'indice non è individuato nel rendesvouz target, è probabile che venga individuato nelle vicinanze
- il processo di walking viene effettuato nelle due direzioni, up e down
- si definisce un hop count per indicare il massimo numero di passi da effettuare (rendezvouz da visitare)
- in una direzione il processo di ricerca si ferma quando l'indice è individuato, nell'altra direzione dopo hop passi

PROPAGAZIONE DELLE QUERIES

- Se lo shift effettuato sulla RPV rientra nella distanza di replicazione (+1,-1), l'indice viene individuato

- replicare l'indice nelle vicinanze del target iniziale aumenta la possibilità di ritrovare l'indice anche in presenza di inconsistenze temporanee

- la distanza di replicazione
  - può essere incrementata (+2 or +3) per aumentare la probabilità di individare l'indice
  - dipende dalla dimensione della RPV

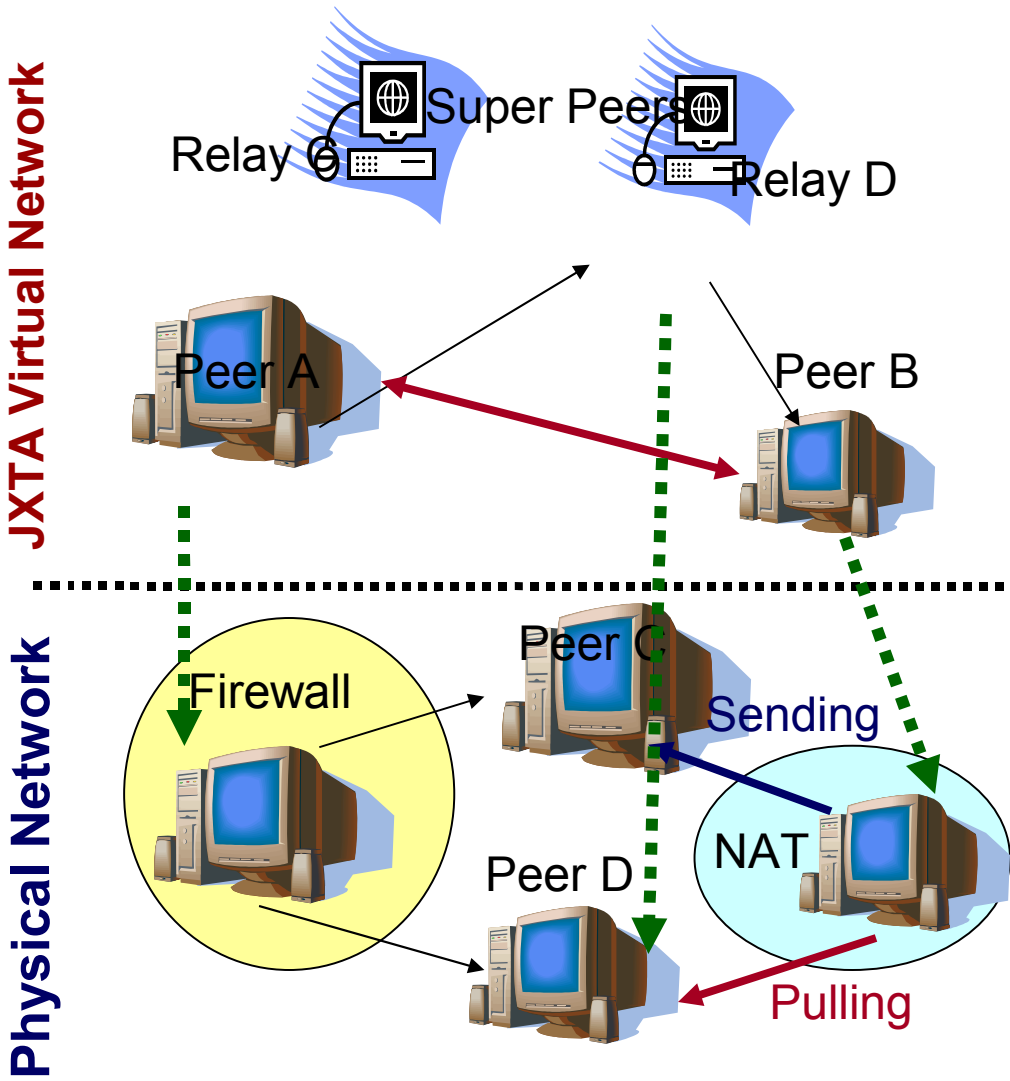
- soluzione di compromesso: diminuisce il costo per la gestione distribuita della RPV

RELAY SUPER PEERS

- Alcuni super peers possono diventare relay peers
- Relay peers= mettono in comunicazione peers che non hanno una connessione fisica diretta (NAT, firewalls)
- Ogni peer può diventare un relay peer se possiede le caratteritistiche necessarie (banda, connettività diretta con la rete,…)
- I relay peer possono memorizzare i messaggi diretti ad host che non possono essere raggiunti o che sono momentaneamente irraggiungibili
- Ogni rely peer possiede una visione degli altri rely peers della rete
- Loosely Consistent Rely Peer View

RELAY SUPER PEERS

- Supponiamo che il peer a voglia spedire un messaggio al peer B
- A non può inviare un messaggio direttamente a B
- B usa D per accedere alla rete
- Un peer può definire, mediante un peer advertisements, l'insieme di relay peers prescelti
- Ogni peer tenta sempre una connessione diretta con gli altri peers prima di utilizzare un relay peers
- Il routing effettuato mediante i relay peers viene effettuato in modo trasparente in maniera dinamica mediante la rete virtuale JXTA



RELAY SUPER PEERS

- Gli edge peers stabiliscono una connessione verso un relay peer (leased – connection, connessione presa in affitto). I relay peers allocano delle code di messaggi per ogni peer collegato.

- L'associazione relay/edge peers è transitoria: un edge peer può cambiare il proprio relay peer

- Di tanto in tanto gli edge peers possono cambiare relay peer per ottimizzare la propria visibilità o per migliorare la propria connettività alla rete

RELAY SUPER PEERS

- Come i rendezvous peer, ogni relay peers mantiene una propria vista dei relay peers disponibili
- Viene utilizzato un insieme di seeding relays per effettuare il bootstrap dei relay peers disponibili
- Gli edge peers mantengono una lista di advertisements di relay peers nella propria cache locale. Quando il peer si disconnette dalla rete, la lista rimane disponibile per successivi reboots
- I seeding relays sono utilizzati solo quando non sono disponibili ulteriori informazioni
- Nel caso di disconnessione di un relay peers, l'edge peer si riconnette ad un altro relay peer in modo trasparente rispetto all'applicazione

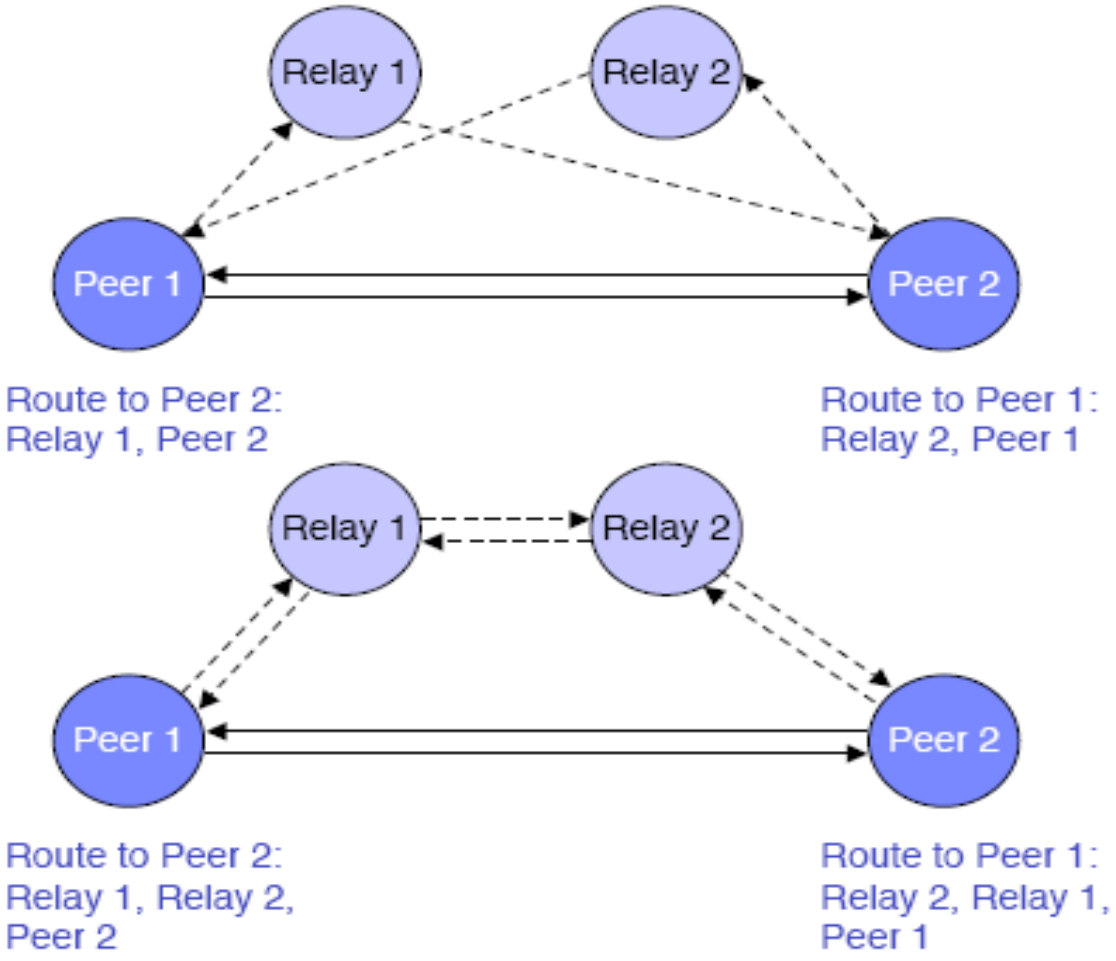
ENDPOINT ROUTING PROTOCOL

- Endpoint Routing Protocol(ERP) Core Protocol di JXTA
- Gestisce il routing dei messaggi, ossia di trovare una sequenza di nodi intermedi (hops) che consenta ad un pacchetto dati di giungere dal mittente fino al destinatario
- Protocollo necessario per gestire una rete P2P distribuita e dinamica, in cui i peer possono
  - connettersi/sconnettersi dinamicamente dalla rets
  - usare firewall/NAT che rendano impossibile una comunicazione bidirezionale
  - non possedere canali di comunicazione diretti fra di loro.
- Questa dinamicità rende impossibile l'utilizzo di tabelle di routing statiche e note a tutti i peer.
- ERP è un core protocol di JXTA ed è utilizzato dai protocolli a più alto livello
- Le sue funzionalità vengono invocate all'interno di metodi a più alto livello, l'utente non interagisce direttamente con l'ERP

ENDPOINT ROUTING PROTOCOL

- In JXTA le informazioni per effettuare il routing sono rappresentate mediante route advertisements
- un route advertisment descrive il percorso necessario per raggiungere un peer come una sequenza ordinata di hops
- ogni hop è identificato da un peer ID con una sequenza opzionale di endpoints per quel peer.
- Un hop corrisponde in generale ad un relay peer, ma può corrispondere anche ad un edge peer (ad esempio in una ad hoc network)
- Su Internet, in generale, solo un relay peer è necessario per comunicare con un peer che si trova a monte di un firewall o di un NAT
- In una rete mobile, un edge peer può implementare un hop per effettuare routing di messaggi tra peers che non sono direttamente connessi

ENPOINT ROUTING PROTOCOL



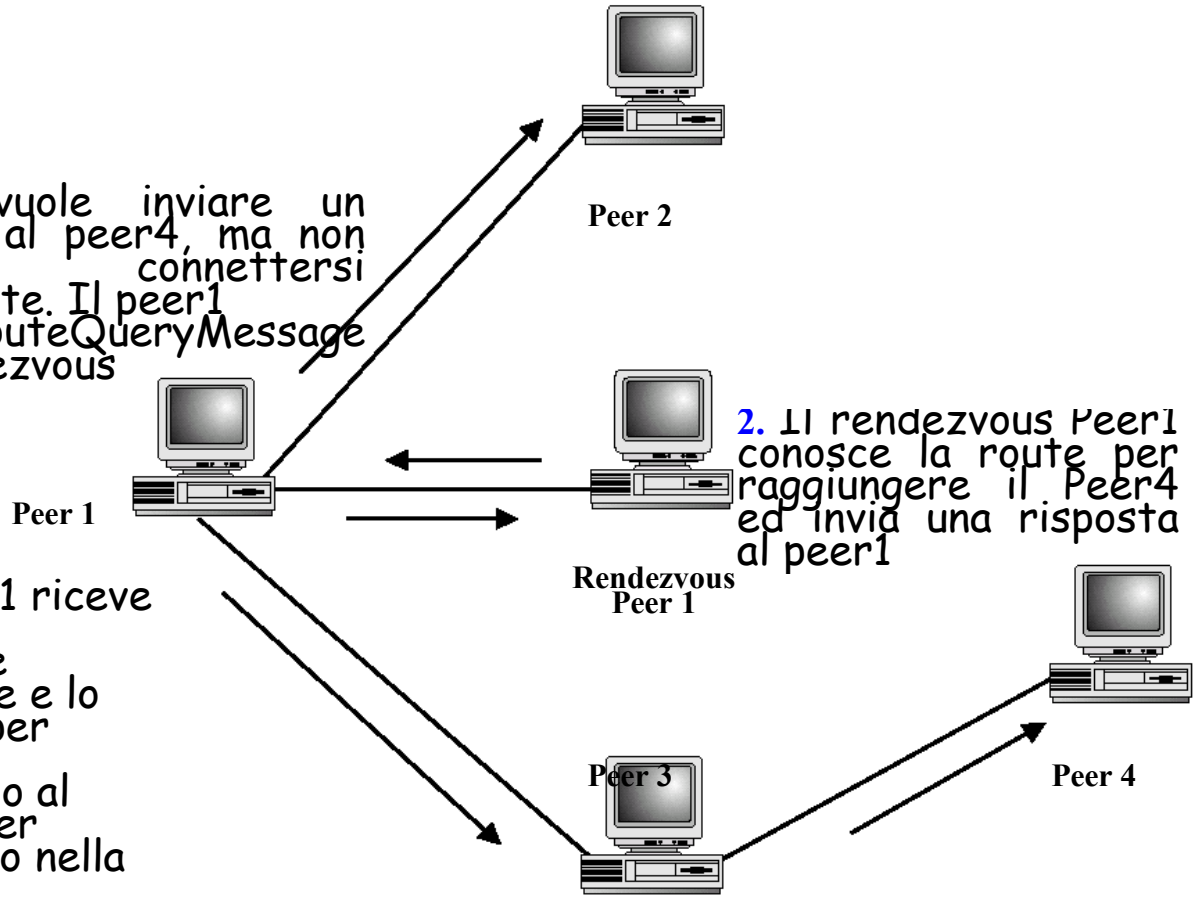
ENPOINT ROUTING PROTOCOL

- Le routes sono indipendenti dal mittente del messaggio
- Esempio: un peer route advertisement per il peer A può contenere una sequenza di hops < Peer B, Peer C, Peer D>
- Il route advertisement può essere utilizzato da diversi mittenti. Ogni mittente può utilizzare una porzione del cammino descritto nell'advertisement
- La descrizione di un cammino contenuta in un peer advertisement può essere utilizzata da diversi mittenti
- Ad esempio, se l'advertisement per il peer A contiene la seguente lista di hops < Peer B, Peer C, Peer D>, ed il peer F può comunicare direttamente con C, F usa solamente la porzione <Peer B, Peer C> del cammino
- Come ogni advertisement, i route advertisements hanno un ciclo di vita limitato (default 15 minuti)

ENDPOINT ROUTING PROTOCOL

- Ogni peer P
  - Conosce come raggiungere alcuni altri peer
  - memorizza nella local cache alcune informazioni che riguardano il routing verso quei peer
- Se il destinatario di un messaggio non rientra fra i peer conosciuti da P, perchè ad esempio è protetto da un firewall o usa un protocollo di trasporto sconosciuto si invia una RouteQuery al proprio rendesvouz
- Protocolli di trasporto supportati:TCP, HTTP, TLS, Beep, ServletHttp, Bluetooth
- Il rendezvous può rispondere con una RouteResponse, in cui vengono indicati gli hops da percorrere per giungere fino al destinatario.
- il RVP può inoltrare la richiesta ai RVP a lui noti
- La Route Response viene memorizzata nella cache locale di P

ENDPOINT ROUTING PROTOCOL



1. Peer vuole inviare un messaggio al peer4, ma non può connettersi direttamente. Il peer1 invia un RouteQueryMessage ai sui rendezvous

2. Il rendezvous Peer1 conosce la route per raggiungere il Peer4 ed invia una risposta al peer1

3. Il peer1 riceve il Route Response Meassage e lo utilizza per inviare il messaggio al primo peer contenuto nella route

4. Il peer3 riceve l'endpoint router message e determina ilprossimo peer da contattare e

5. il peer4 riceve il mesaggio e si identifica come destinatario finale del messaggio

JXTA E LA GRIGLIA: PROPOSTE DI TESI

- Utilizzare JXTA per il grid computing
- JXTA permette la scoperta dinamica di risorse e di servizi
- Le risorse della griglia vengono pubblicate mediante advertisements
- Quando un utente sottomette un job per l'esecuzione, richiede le risorse necessarie per l'esecuzione del job, ad esempio pubblicando un job advertisement
- I peer che possiedono le risorse opportune, rispondono, descrivendo le proprie risorse
- Un peer viene scelto per l'esecuzione
- Codice + dati del job trasmessi mediante una pipe
- Riallocazione dinamica del job nel caso in cui si rendano disponibili peers con risorse più adatte all'esecuzione del job

JXTA E LA GRIGLIA: PROPOSTE DI TESI

- Supporto di applicazioni basate su paradigmi tipo farm computing
- Architettura possibile: Definizione di diversi peer groups per workers, task dispatchers, repository, monitors
  - Worker Group: gruppo di peers responsabili per l'elaborazione di particolari jobs
  - Task Dispatcher Group: gruppi di peers che distribuiscono i jobs ai workers
  - Repository Group: Cache per codice e dati
  - Monitor Group: Top level group, monitora l'attività dell'intero sistema
- I worker comunicano con tutti i task dispatchers di un gruppo
- Mccanismi di bilanciamento del carico tra i dispatchers
- Memorizzazione distribuita dei dati nei repositories
- Monitor :decide se attivare un numero meggiore di dispatchers, dove memorizzare codice e dati, etc....

JXTA VS.SOAP:PROPOSTE DI TESI

Simple Object Access Protocol (SOAP)

- Permette l'accesso a Servizi Remoti
- Usa XML

JXTA

- Permette l'accesso a Servizi remoti
- Definisce un ricco insieme di protocolli (Peer Discovery, Peer groups , Peer pipes, meccanismi di comunicazione flessibile)

- SOAP messages can be send thru JXTA