

Lezione n.13

JXTA:

**SHARED RESOURCE
DISTRIBUTED INDEX (SDRI)**

**Materiale didattico
sulla pagina del corso
Laura Ricci**

RENDEZVOUS SUPERPEERS

- JXTA introduce
 - Edge Peers
 - Rendez-Vouz Peers (Relay Peers)
- **Rendezvous Peers**: peers che si sono resi disponibili ad effettuare l'instradamento degli advertisements sulla rete
- un Rendez Vouz peer
 - memorizza nella propria cache un indice che contiene le chiavi degli advertisements pubblicati in alcuni edge peers
 - gli advertisements sono memorizzati unicamente negli edge peers, non nei rendezvous peers
 - gli edge peers inviano al rendez-vous peer le chiavi degli advertisement pubblicati

LOOSELY CONSISTENT DHT

- meccanismi di memorizzazione/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

ADVERTISEMENTS PUSH

- un edge peer EP identificato da EdgePeerID pubblica un advertisement
- EP invia (push) al proprio rendezvous 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

$$H = \text{Hash}(\text{Type} + \text{Attribute} + \text{Value})$$

Type = tipo dell'advertisement

Attribute = un attributo dell'advertisement

Value = valore associato a quell'attributo

- indice dell'advertisement

$(H, \text{EdgePeerID})$

in base al valore di H, l'indice viene mappato su un rendezvous peer MRV

ADVERTISEMENT PUSH

- l'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 vous peer

calcola la funzione hash H

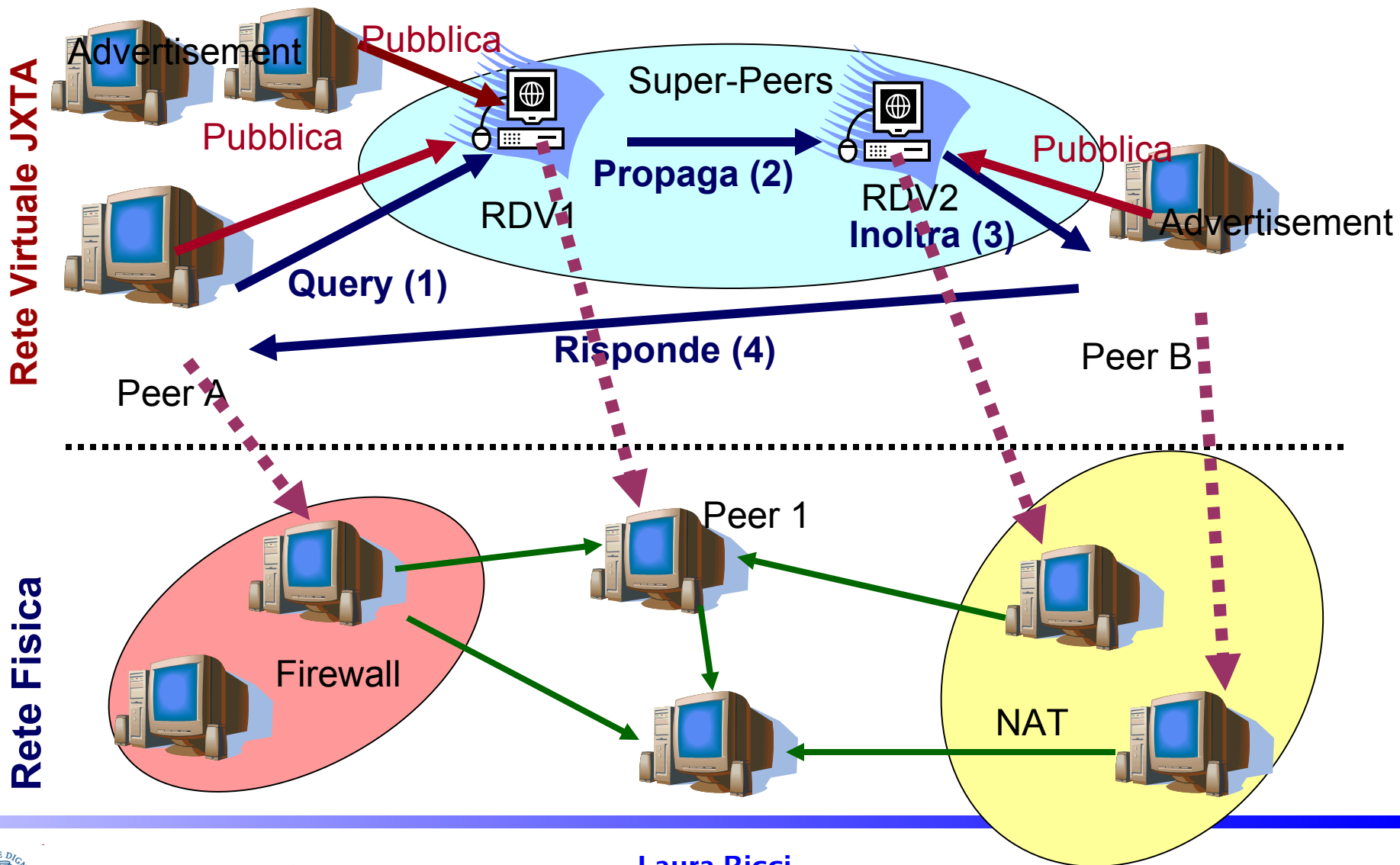
mappa H su uno dei rendezvous peer MRV della rete

invia l'indice dell'advertisement a MRV

RICERCA DEGLI ADVERTISEMENTS

- 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 rendez-vous 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

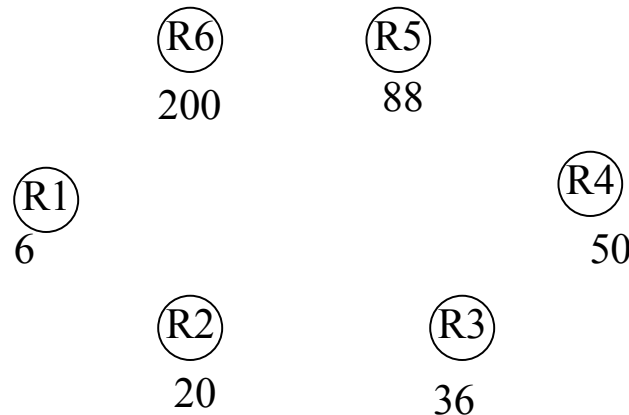
UN ESEMPIO



RENDEZVOUS PEER VIEW

- Rendezvous Peer View (RPV)= lista di rendezvous peer 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

- le RPV di rendezvous peer diversi appartenenti allo stesso gruppo possono risultare inconsistenti

DISTRIBUZIONE SPAZIO DEGLI IDENTIFICATORI

Definizione del Mapping M

- M: identificatori di advertisement \rightarrow Identificatori di RVP
- Esempio: mapping "a la Chord". Un advertisement identificato dall'identificatore AI viene mappato sul primo rendezvous peer che ha identificatore maggiore di AI
- Inconsistenza:

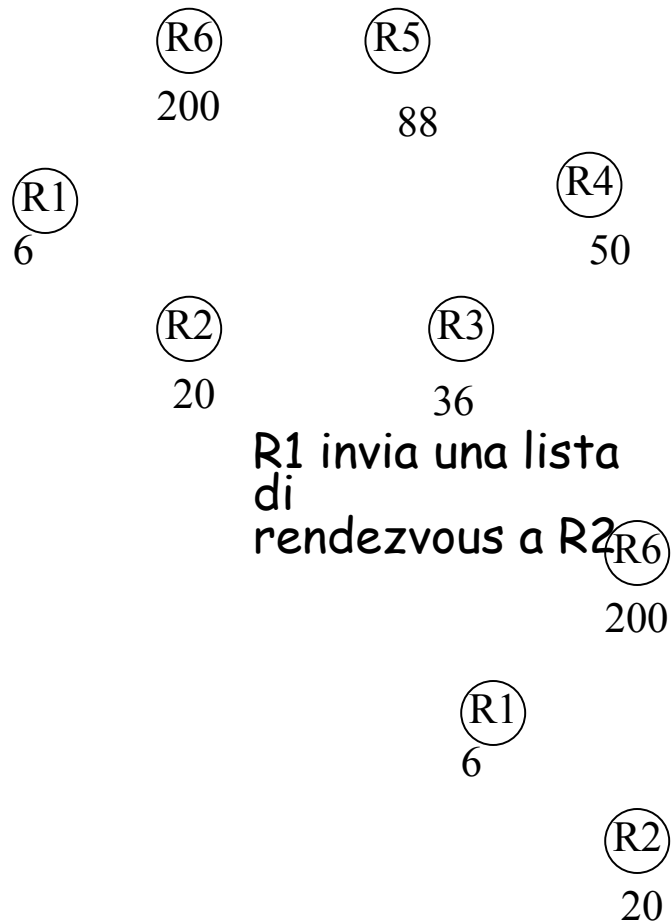
il mapping può restituire valori diversi su rendezvous diversi, poichè le RPV di possono contenere puntatori a rendezvous diversi (RPV inconsistenti)

RENDEZVOUS PEER VIEW

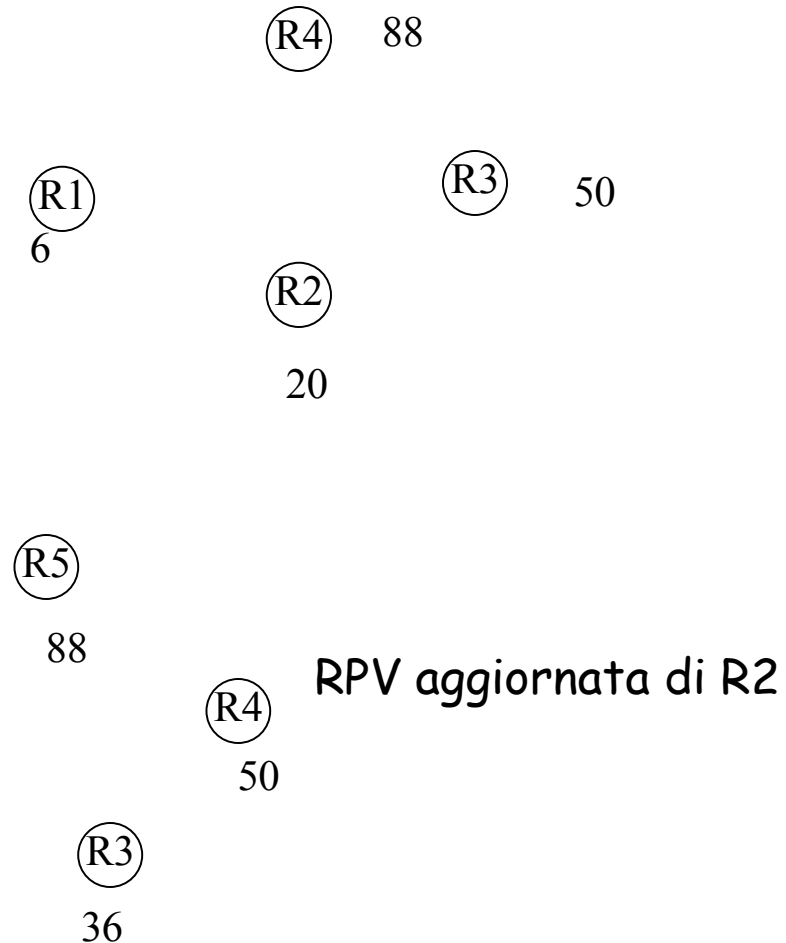
- I rendezvous selezionano periodicamente un numero random di peer dalla loro RPV ed inviano a loro una lista random dei rendezvous che conoscono
- I rendezvous inviano anche periodicamente un **heartbeat** ai rendezvous vicini (quelli che occupano le posizioni +1 e -1 nella RPV)
- I rendezvous
 - aggiornano le loro tabelle quando ricevono informazioni dai vicini
 - ed eliminano dalle proprie tabelle i rendezvous da cui non ricevono heartbeat per più di un certo intervallo di tempo
 - possono inviare un ping al vicino per verificare se sono connessi
- Le RPV dei superpeer **tendono a convergere** mediante l'invio di messaggi successivi

RENDEZ-VOUZ PEER VIEW: CONVERGENZA

RPV di R1



RVP di R2



SEEDING RENDEZVOUS

- Al momento del bootstrap un rendez vous peer deve inizializzare la propria RPV (Rendezvous Peer View)
- All'inizio il rendezvous conosce solamente un insieme di **seeding rendezvous**
- Seeding Rendezvous = insieme di rendezvous che permettono il bootstrap di un rendezvous peer sulla rete
- dopo il bootstrap, i seeding rendezvous vengono utilizzati come punto di distribuzione di advertisements di rendezvous che si sono connessi alla rete
- Ogni peergroup può definire un proprio insieme di seeding rendezvous peers
- Un rendezvous peer è configurato mediante una lista predefinita di seeding rendezvous

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 rendezvous peer

- ricerca sui seeding rendezvous peer ulteriori rendezvous advertisements
- scambia la propria RPV con i rendezvous 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 accelerare la convergenza della RPV

CONNESSIONE EDGE/RENDEZVOUS PEER

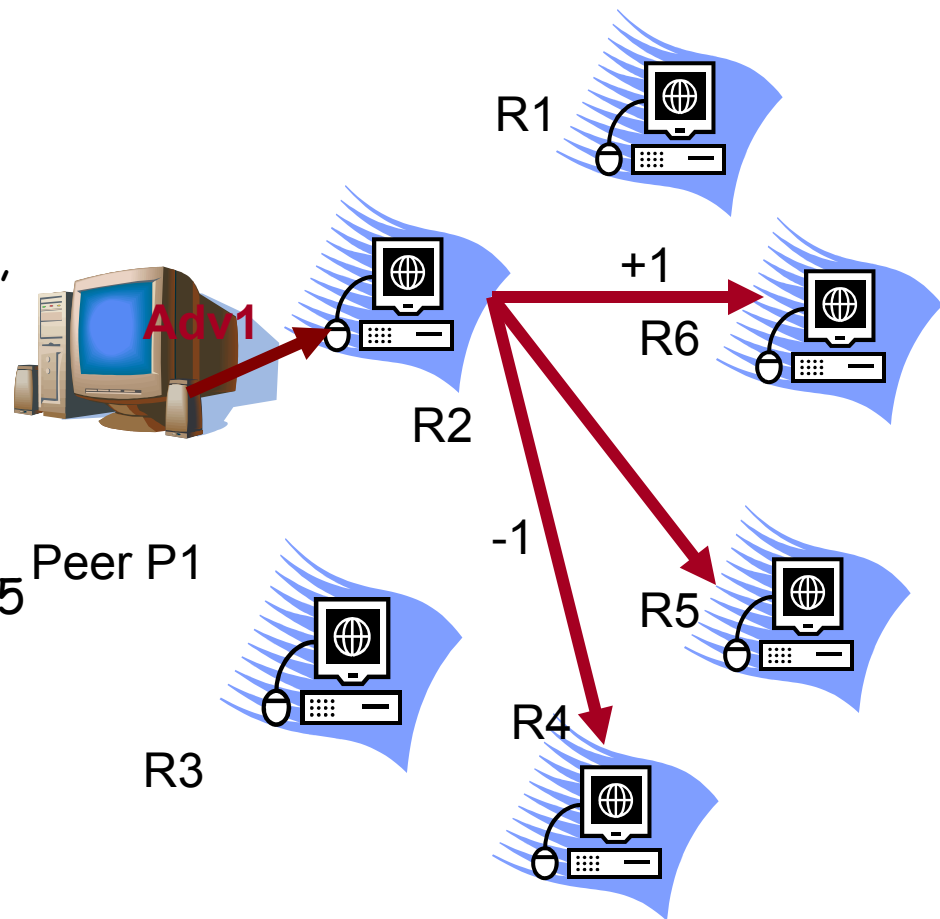
- un edge peer è connesso in ogni istante ad un unico rendezvous peer
- un edge peer ricerca continuamente (in background) ulteriori rendezvous peer advertisements e li memorizza nella propria cache locale
- i rendezvous advertisement sono mantenuti nella cache locale del peer al momento della sua disconnessione 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 PEER

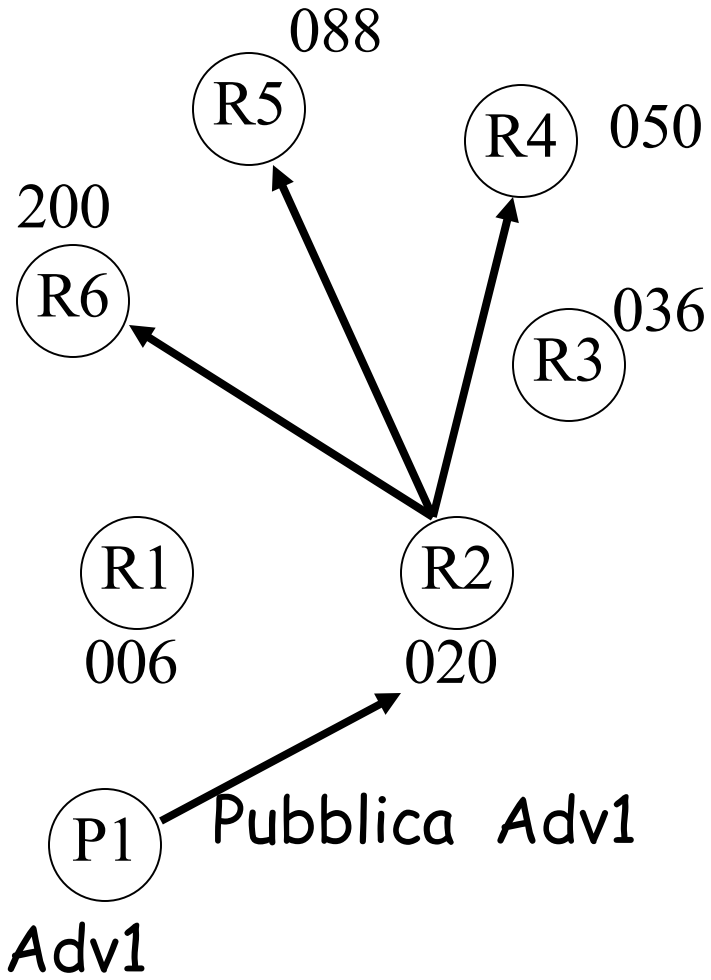
- ricerca di rendezvous peers:
 - gli advertisement vengono ordinati in gruppi
 - vengono inviate sulla rete gruppi di richieste
- se non viene stabilita una connessione entro un certo intervallo di tempo (es: 30 secondi) l'edge peer invia una richiesta mediante servizi di livello più basso (multicast, pipes)
- se dopo un ulteriore periodo (30 sec.), non viene individuato alcun rendezvous peer, si contatta uno dei seeding rendezvous
- se nessun seeding rendezvous risponde dopo un periodo di tempo (5 minuti), l'edge peer diventa un rendezvous peer
non appena riesce a contattare un rendezvous peer, diventa nuovamente un edge peer

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



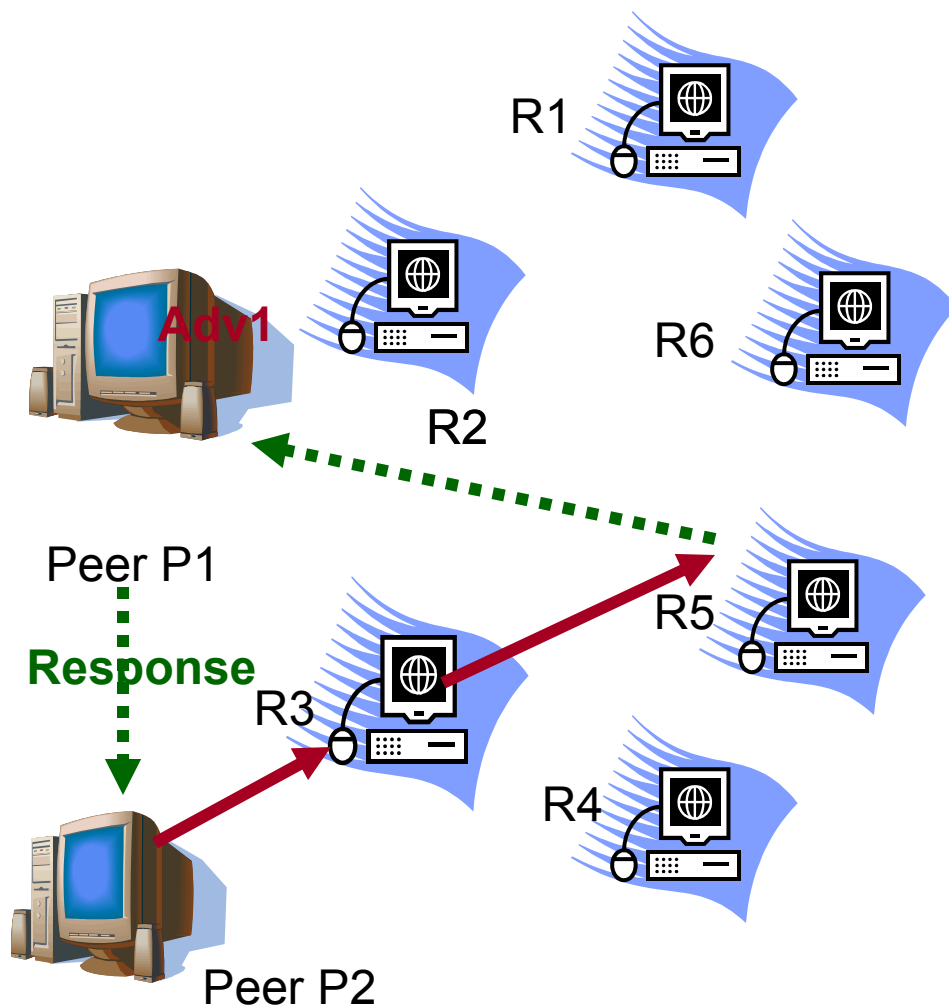
Tipo dell'advertisement : peer
Nome attributo : name
Valore attributo : P1

- $H(\text{peer}, \text{name}, P1) = 87$
- $DHT(87) = R5$
- L'indice viene
 - pubblicato su R5 = (87, P1)
 - replicato su R4 e su R6



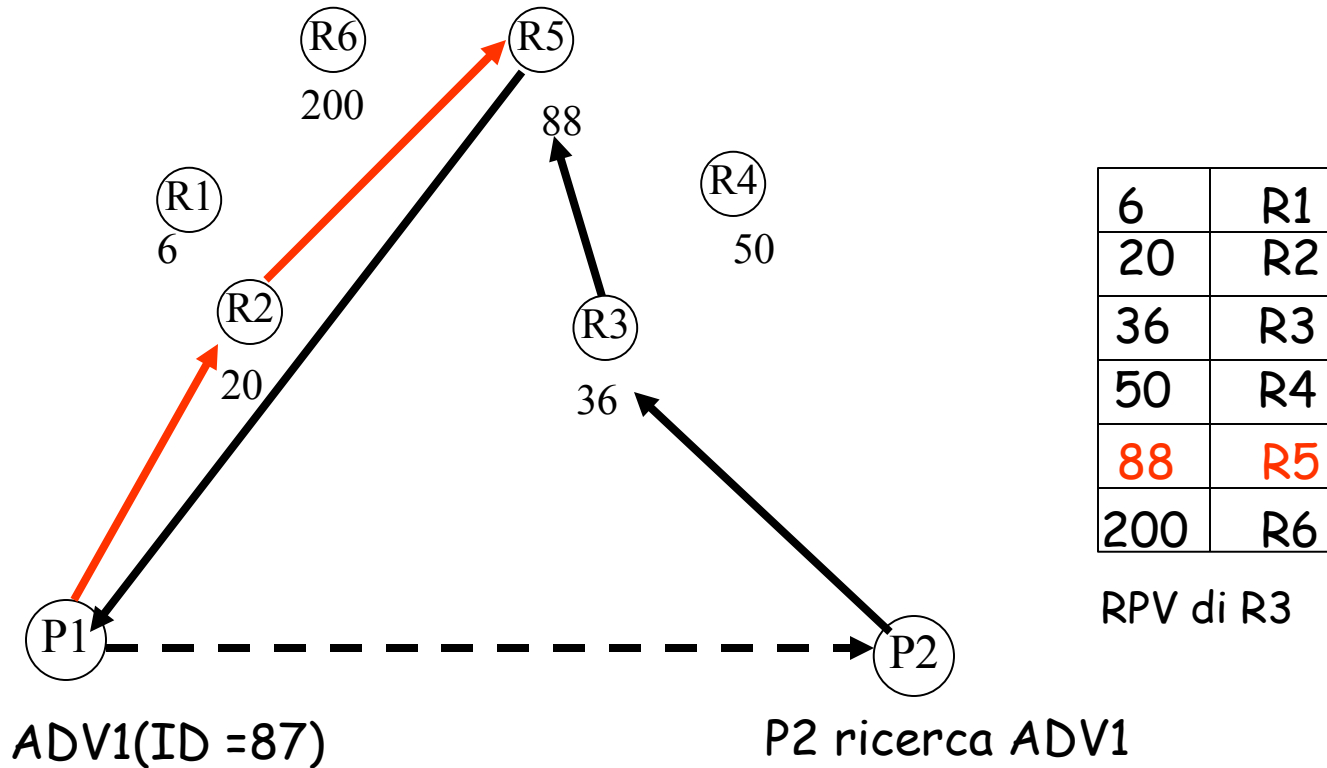
RPV CONSISTENTI: PROPOGAZIONE DELLE QUERIES

- 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



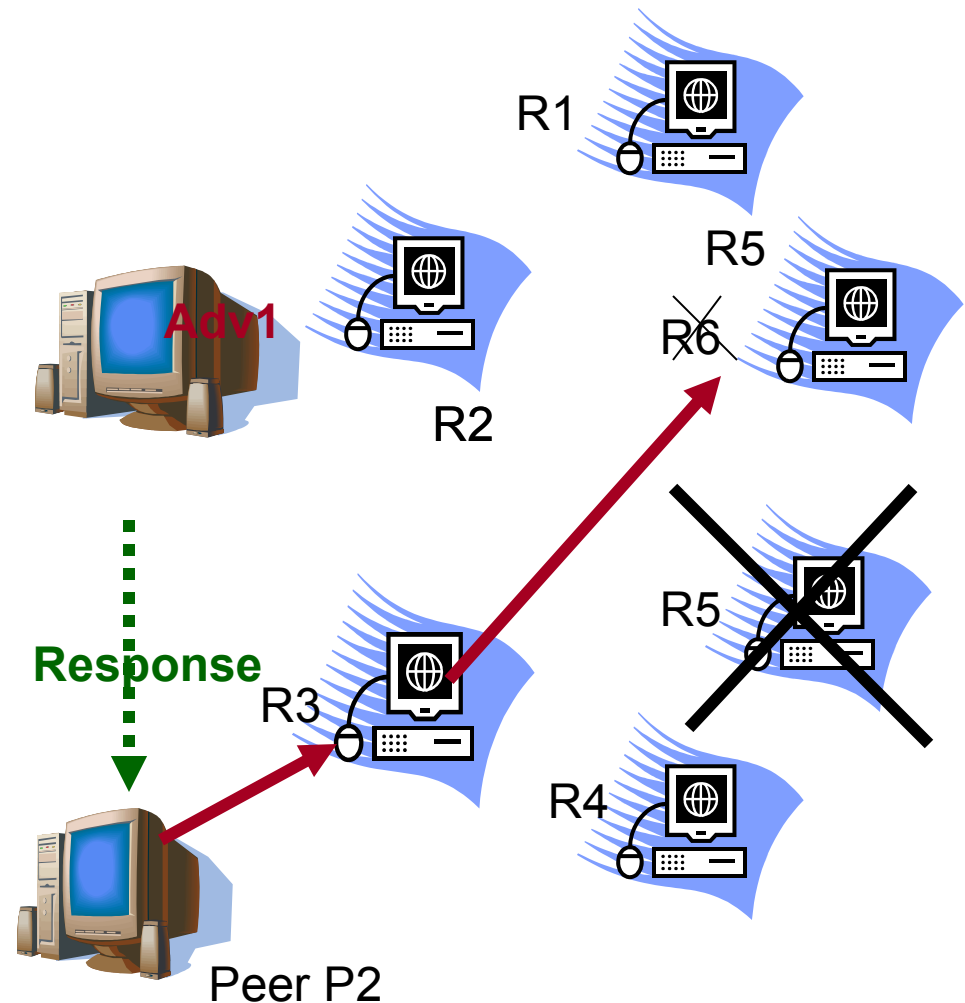
RPV CONSISTENTI: PROPAGAZIONE DELLE QUERIES

- Pubblicazione dell'advertisement
- Ricerca dell'advertisement



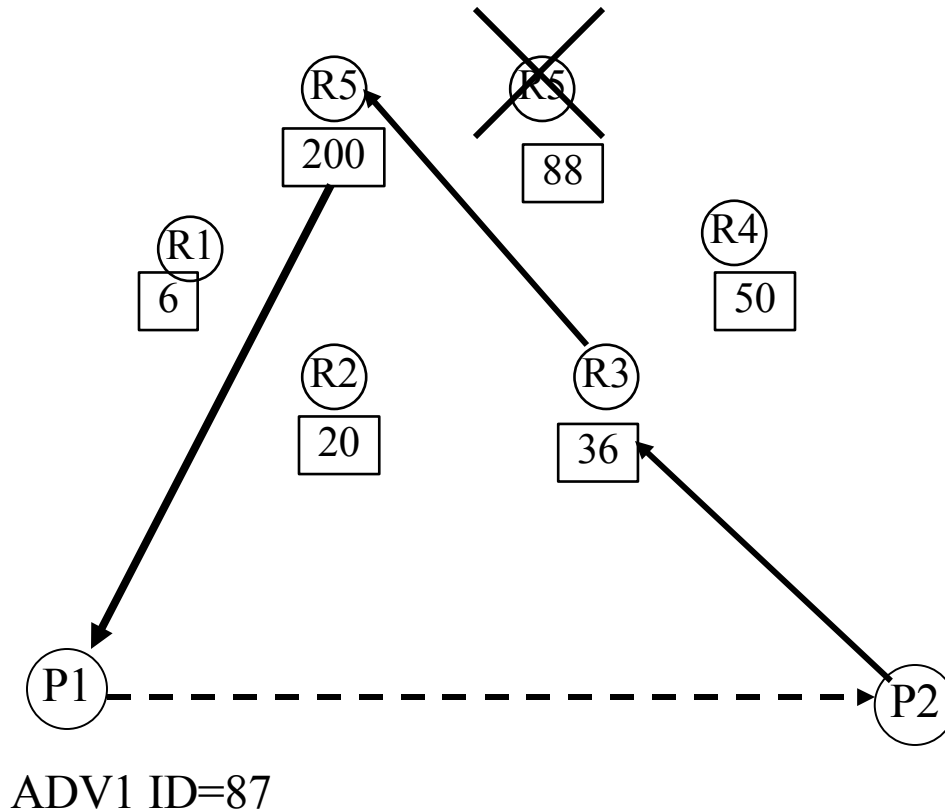
RPV INCOSISTENTI: PROPOGAZIONE DELLE QUERIES

- R5 si disconnette : R3 aggiorna la propria RPV
- R3 scopre che R5 si e' 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)
- Poiche' l'indice e' stato pubblicato anche su R6, la ricerca da esito positivo



RPV INCONSISTENTI: PROPAGAZIONE DELLE QUERIES

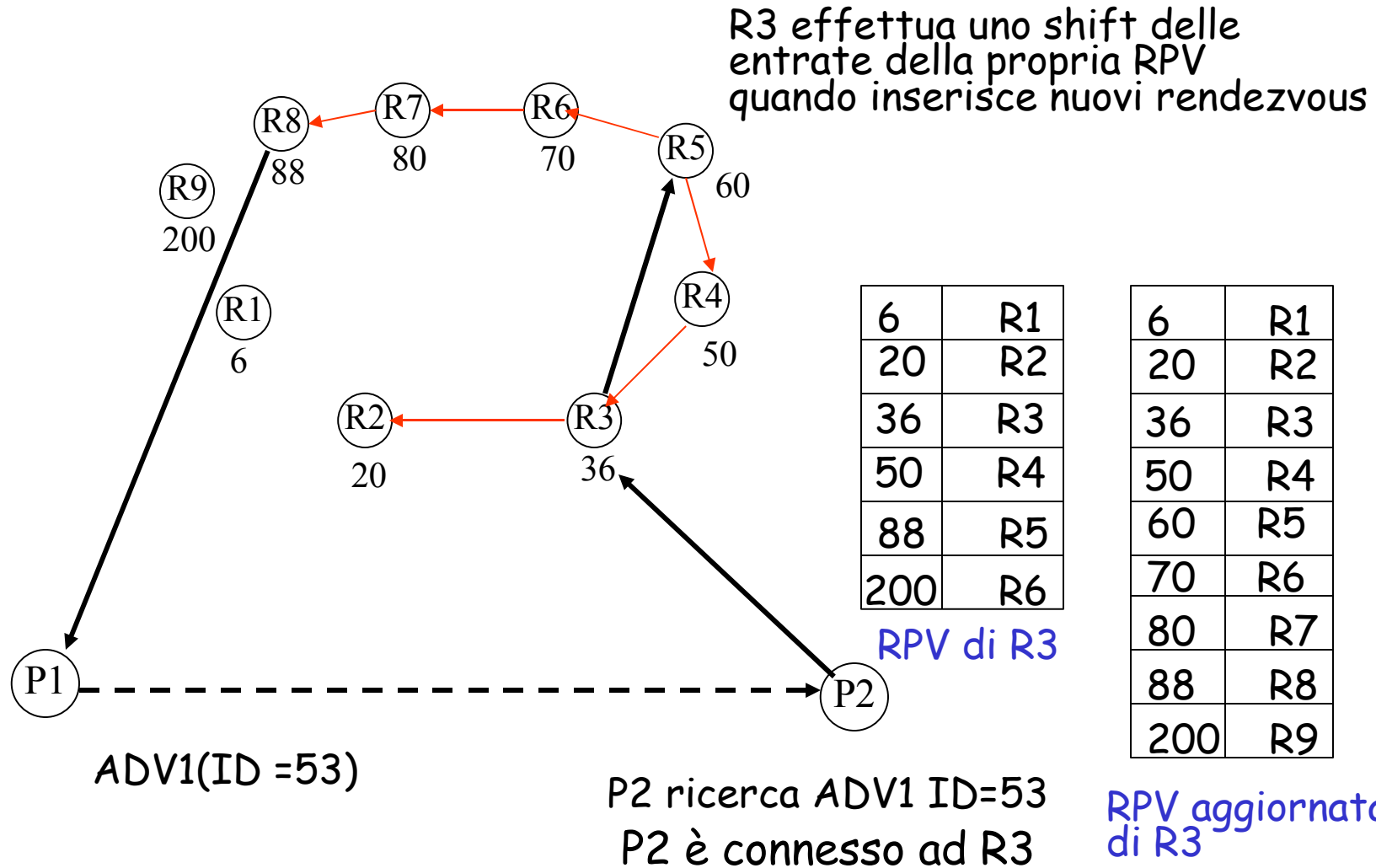
P2 è connesso ad R3



6	R1
20	R2
36	R3
50	R4
200	R5

RPV di R3

RPV CONSISTENTI: LIMITED RANGE WALKER



LIMITED RANGE WALKER

- *Limited Range Walker* = utilizzato per "camminare" sui rendezvous vicini al rendezvous restituito dal mapping M (rendezvous target)
- se l'indice non è individuato nel rendezvous 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 (rendezvous 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 individuare l'indice
 - dipende dalla dimensione della RPV
- soluzione di compromesso: diminuisce il costo per la gestione distribuita della RPV

RELAY SUPER PEERS

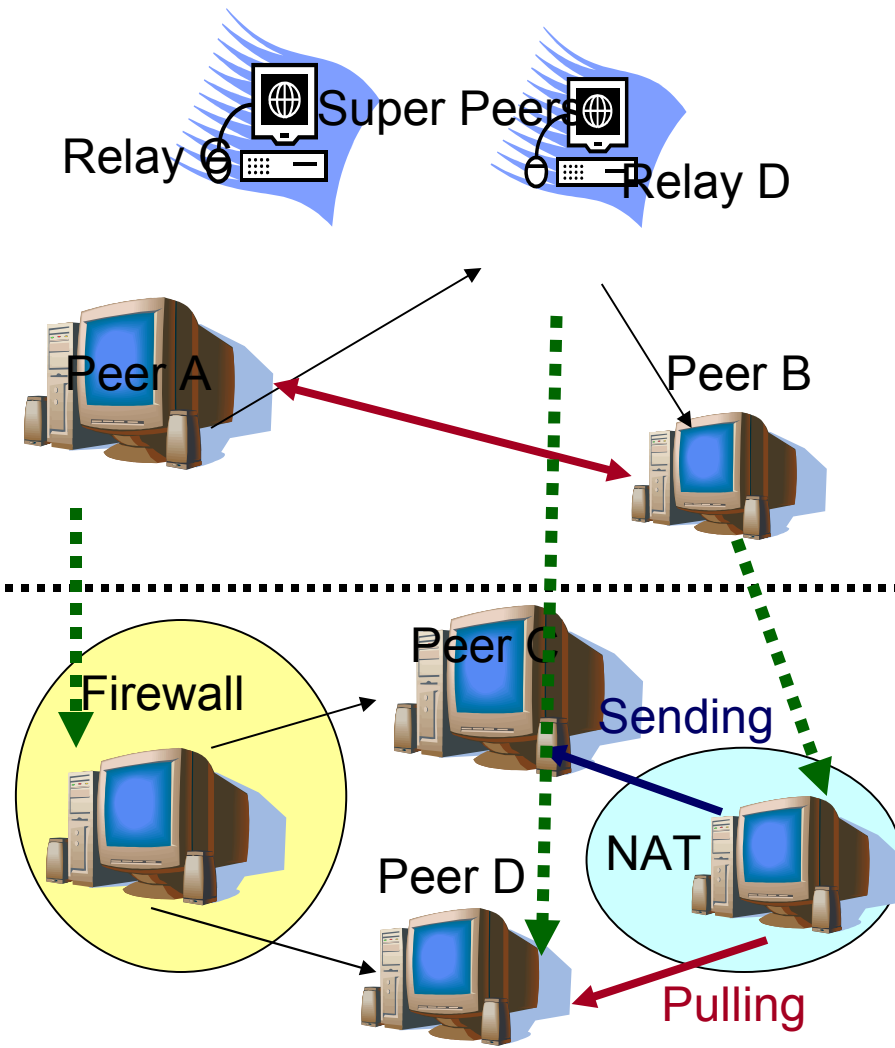
- Alcuni super peers possono diventare relay peers
- Relay peers= mettono in comunicazione peers che non hanno una connessione fisica diretta (NAT, firewalls)
- Ogni peer può diventare un relay peer se possiede le caratteristiche necessarie (banda, connettività diretta con la rete,...)
- I relay peer possono memorizzare i messaggi diretti ad host che non possono essere raggiunti o che sono momentaneamente irraggiungibili

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

JXTA Virtual Network

Physical Network



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

RELY SUPER PEERS

- In JXTA le informazioni per effettuare il routing sono rappresentate mediante *route advertisements*
- un route advertisement 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
- 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
- Gli edge peers
 - memorizzano i routing advertisements nella propria cache locale
 - inviano queries sulla rete per scoprire routing advertisements verso una certa destinazione

RELY SUPER-PEERS

- Esempio: un peer route advertisement per il peer A può contenere una sequenza di hops $\langle \text{Peer B}, \text{Peer C}, \text{Peer D} \rangle$
- 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 $\langle \text{Peer B}, \text{Peer C}, \text{Peer D} \rangle$, ed il peer F può comunicare direttamente con C, F usa solamente la porzione $\langle \text{Peer B}, \text{Peer C} \rangle$ del cammino
- Come ogni advertisement, i route advertisements hanno un ciclo di vita limitato (default 15 minuti)