JXTACh: A Chord-based JXTA version

Lorenzo Corsani, Carlo Nocentini,
Pilu Crescenzi, Leonardo Lanzi

University of Florence

September 25, 2008

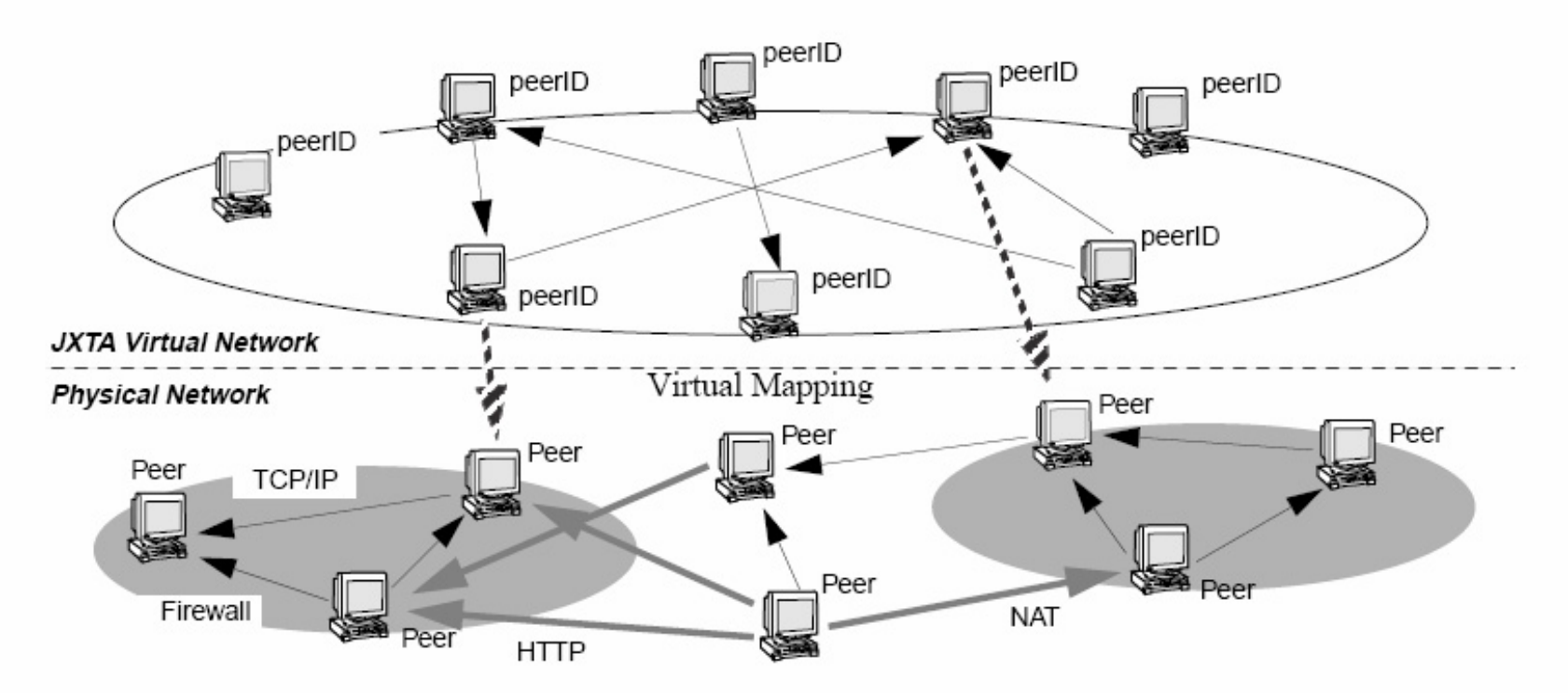
Introduction

- Technology developed by Sun Microsystems to support P2P application development [1]
  - JXTA derive from *juxtaposed*
    - It works side by side the client/server architecture
- Open set of P2P protocols
- Main goals
  - Interoperability
  - Ubiquity
  - Platform independence

Basic concepts

- **Peer**: any device connected to a JXTA network that implements at least one protocol
  - Identified by unique identifier (JXTA ID)
  - Three types
    - Edge peer: simple peer that can send/receive messages, do not route, with/without cache
    - Rendezvous peer: as before plus routing and storing
    - Relay peer: allow communication between peers not directly connected (e.g. because of firewalls or NATs)
- **PeerGroup**: collection of peers that participate to the same application
  - Yields services only to group participants
- **Advertisement**: XML representation of an entity, service or resource
- **Pipe**: virtual communication channel between two peers
  - Unidirectional and asynchronous

JXTA virtual network (picture taken from [2])



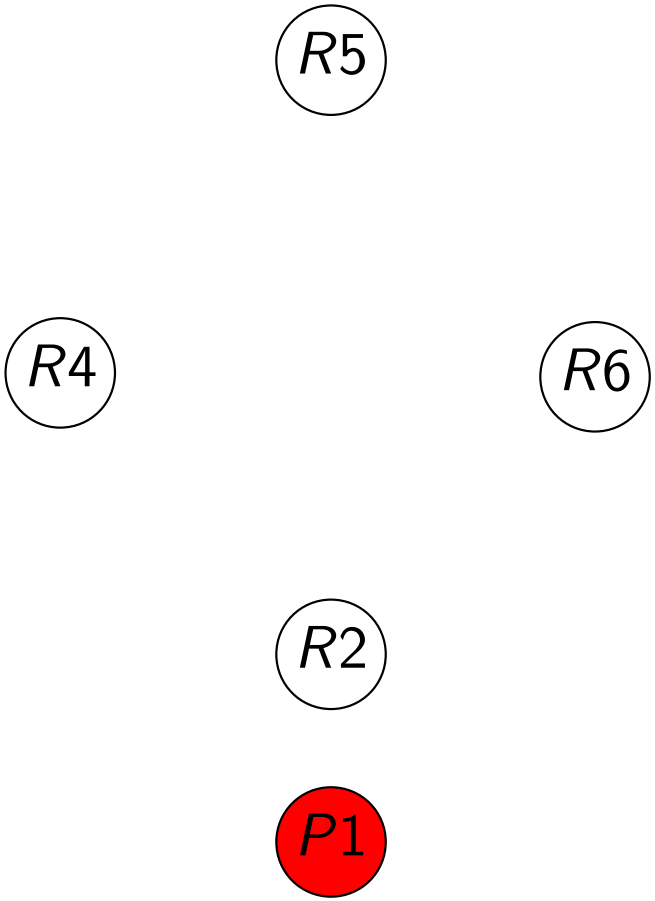
JXTA protocols

- *Core Specification Services*
  - **Peer Resolver Protocol**: dispatcher for all messages directed to the protocols
  - **Endpoint Routing Protocol**: manages the interaction between virtual and physical network and the composition of the routing paths
- *Standard Services*
  - **Peer Discovery Protocol**: allows the publication and search of resources
  - **Peer Info Protocol**: allows monitoring of peer status
  - **Pipe Binding Protocol**: manages the creation of of communication channels between peers
  - **Rendezvous Protocol**: allows connection between peer and its rendezvous peer and manages propagation of messages

Rendezvous Protocol [3]

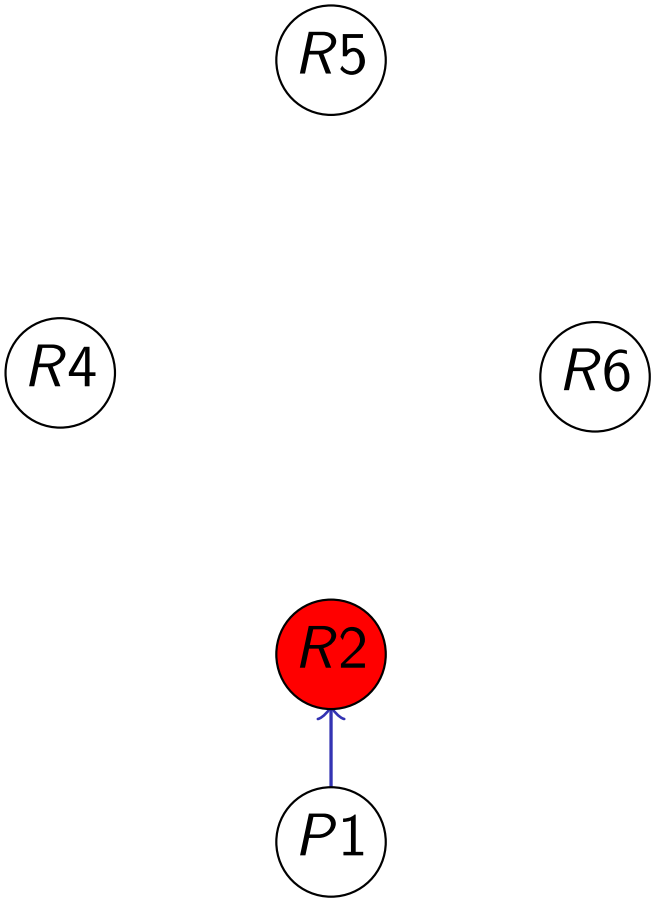
- *Rendezvous Peer View*
  - Rendezvous knowledge of other rendezvous peers in the same group
  - Consistency maintained by periodic message exchanges
- *Shared Resource Distributed Index*
  - Distributed index maintained by rendezvous peers
  - For each advertisement publication
    - Peer sends advertisement SRDI message to its rendezvous R
    - R computes index
    - R computes hash function of index
    - R determines rendezvous in RPV to which index must be sent

Advertisement publication



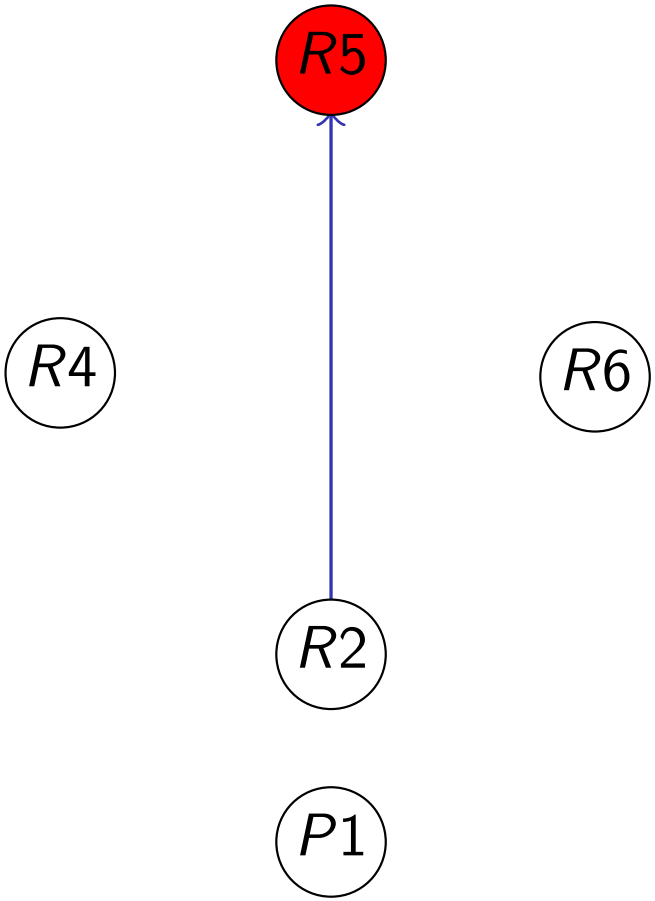
- P1 publish advertisement
- P1 sends SRDI message to R2
- R2 determines rendezvous R5
- R5 stores index in its *SRDI*
- R2 sends index to R5 neighbors (R4 and R6)

Advertisement publication



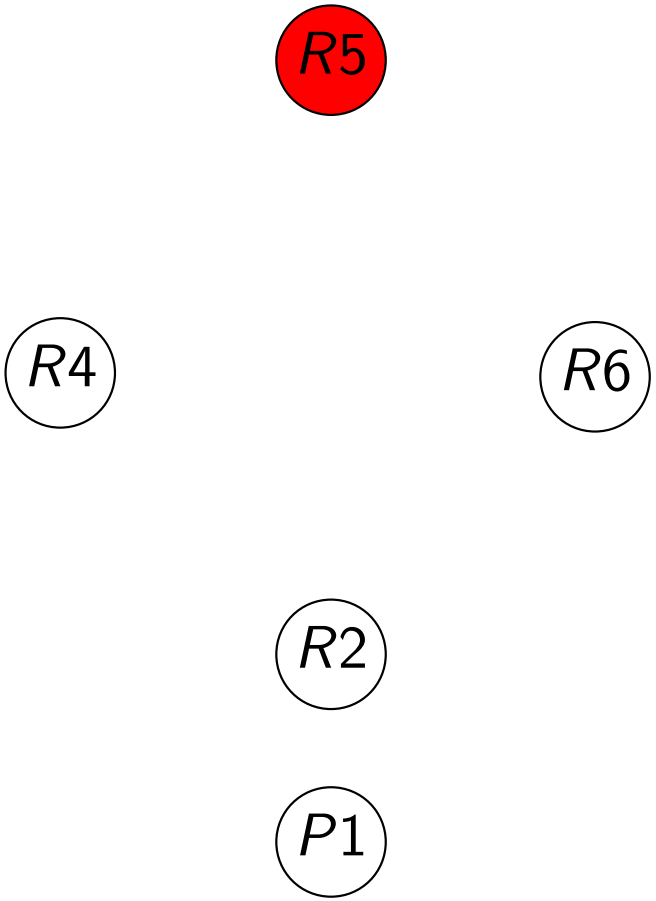
- P1 publish advertisement
- P1 sends SRDI message to R2
- R2 determines rendezvous R5
- R5 stores index in its *SRDI*
- R2 sends index to R5 neighbors (R4 and R6)

Advertisement publication



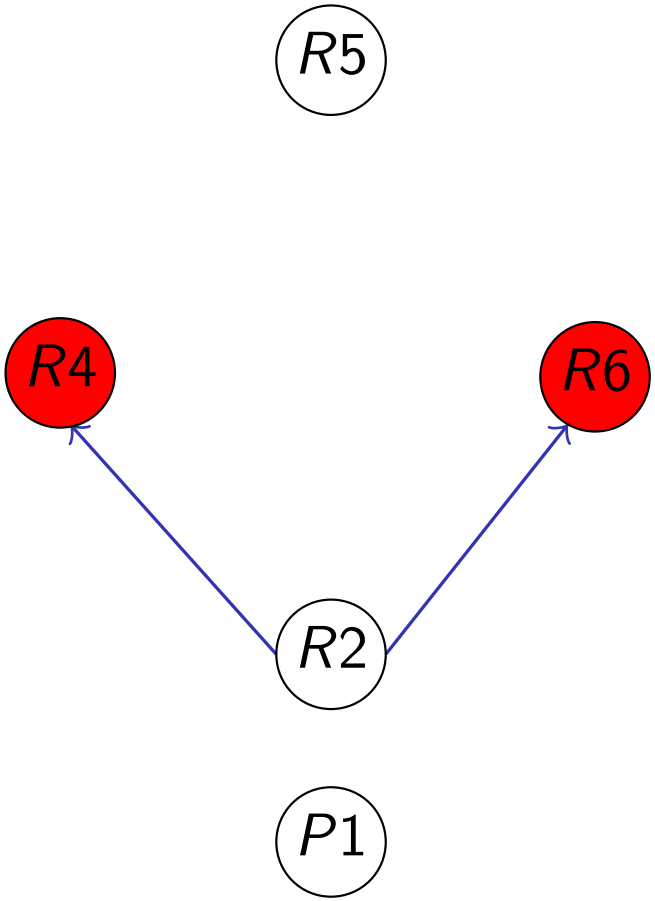
- P1 publish advertisement
- P1 sends SRDI message to R2
- R2 determines rendezvous R5
- R5 stores index in its *SRDI*
- R2 sends index to R5 neighbors (R4 and R6)

Advertisement publication



- P1 publish advertisement
- P1 sends SRDI message to R2
- R2 determines rendezvous R5
- R5 stores index in its *SRDI*
- R2 sends index to R5 neighbors (R4 and R6)

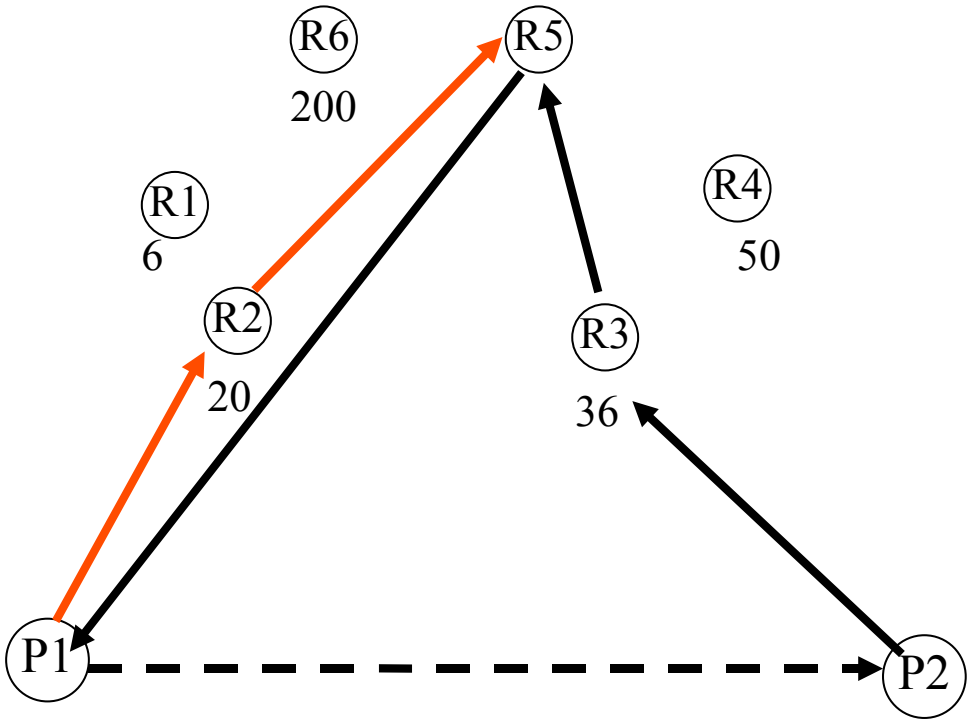
Advertisement publication



- P1 publish advertisement
- P1 sends SRDI message to R2
- R2 determines rendezvous R5
- R5 stores index in its *SRDI*
- R2 sends index to R5 neighbors (R4 and R6)

Search: consistent RPVs case (pictures taken from [4])

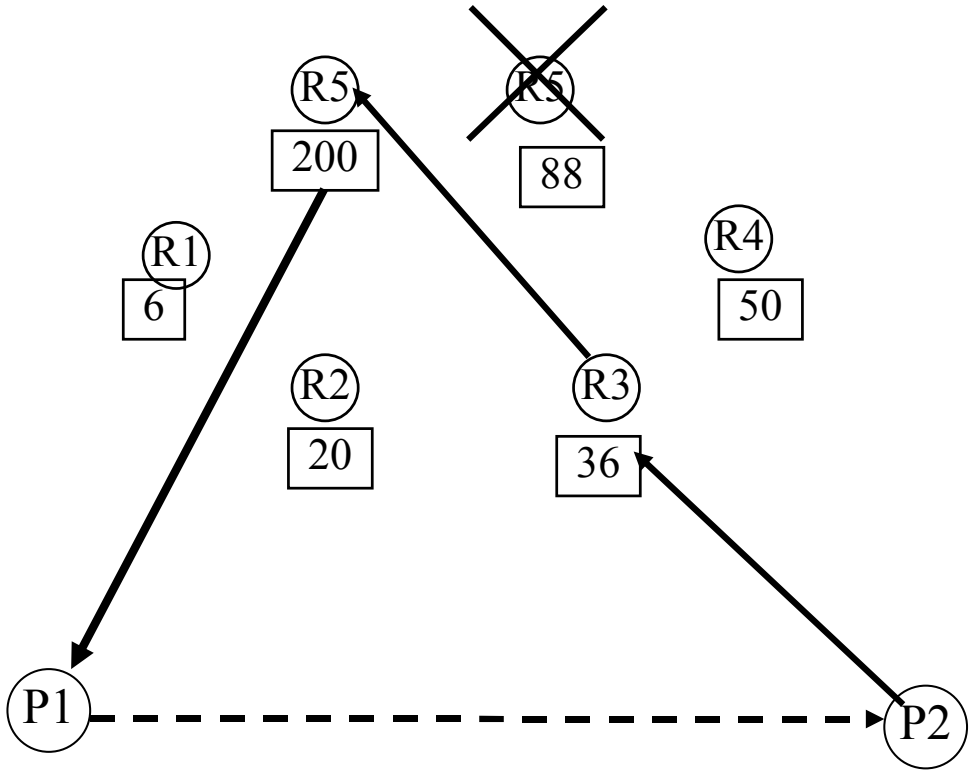
6	R1
20	R2
36	R3
50	R4
88	R5
200	R6



Search: inconsistent RPVs case (pictures taken from [4])

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

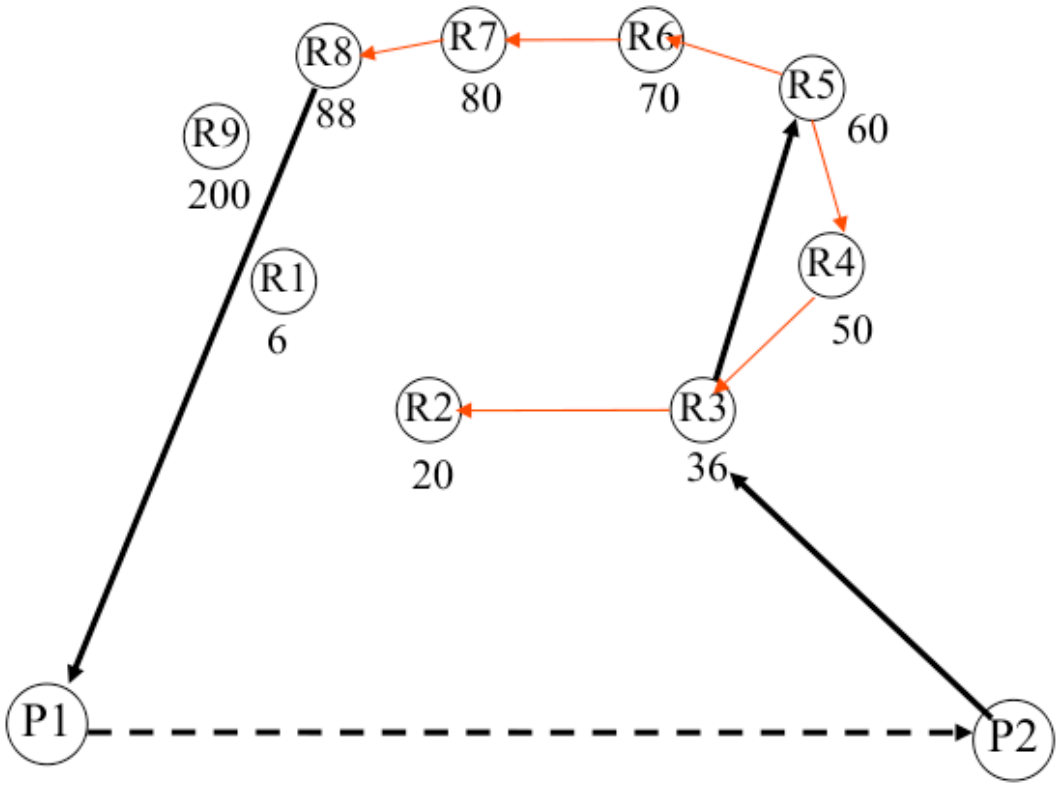
6	R1
20	R2
36	R3
50	R4
200	R5



Search: limited range walker (pictures taken from [4])

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

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



Limited range walker

- Used to walk on rendezvous close to the one returned by the RPV
  - If index is not in target rendezvous, maybe it is close to it
- Bidirectional process: up and down
- Limited process: maximum number of hops is limited
- Justification
  - High maintenance cost required by a real DHT to manage publication/search process

Is it possible to improve the rendezvous protocol performances by replacing the loosely consistent DHT with a real DHT?

Our work

- Integration of Chord protocol (pure DHT) into JXTA architecture
  - First prototype of JXTACh
- Tests and benchmarks on the prototype
  - Test on LAN
  - Test on class C subnets of same class B network
- Test on transparency
  - Test on existing small applications (done)

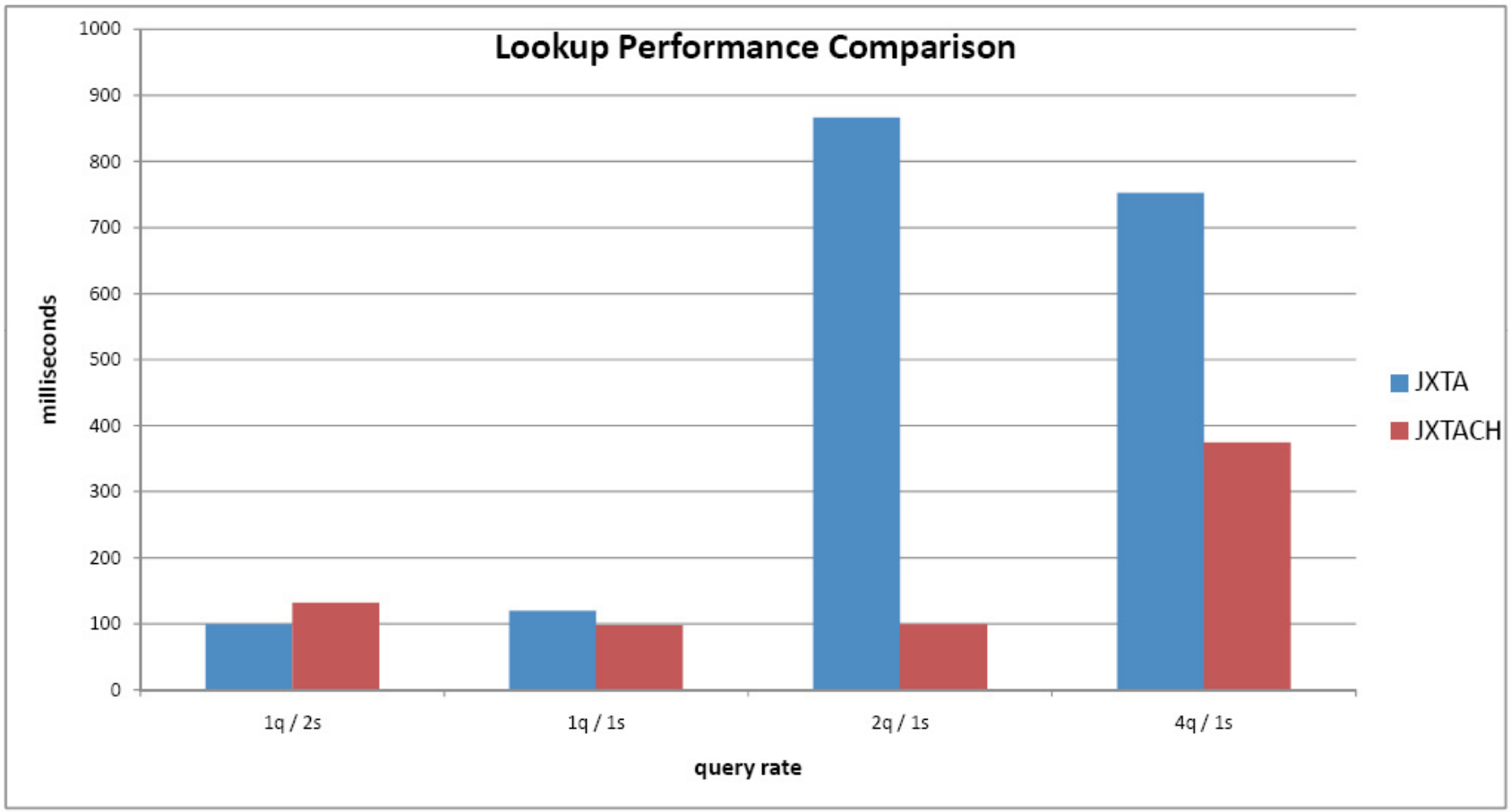
Target of our modification

- JXTA loosely-consistent DHT replaced with Chord DHT policy
- Modification to three main parts
  - Rendezvous Peer View ($\rightarrow$ *Finger Table*)
  - Walker mechanism ($\rightarrow$ *Chord based walker*)
  - Shared Resource Distributed Index ($\rightarrow$ *Chord based SRDI*)

LAN test settings

- Same structure for both protocol versions
- 40 computers in the same LAN
- 1 edge peer publishing 1000 advertisement (fake peer advertisements)
- 1 edge peer looking for each advertisement previously published at a variable query rate
- We measured the average lookup time

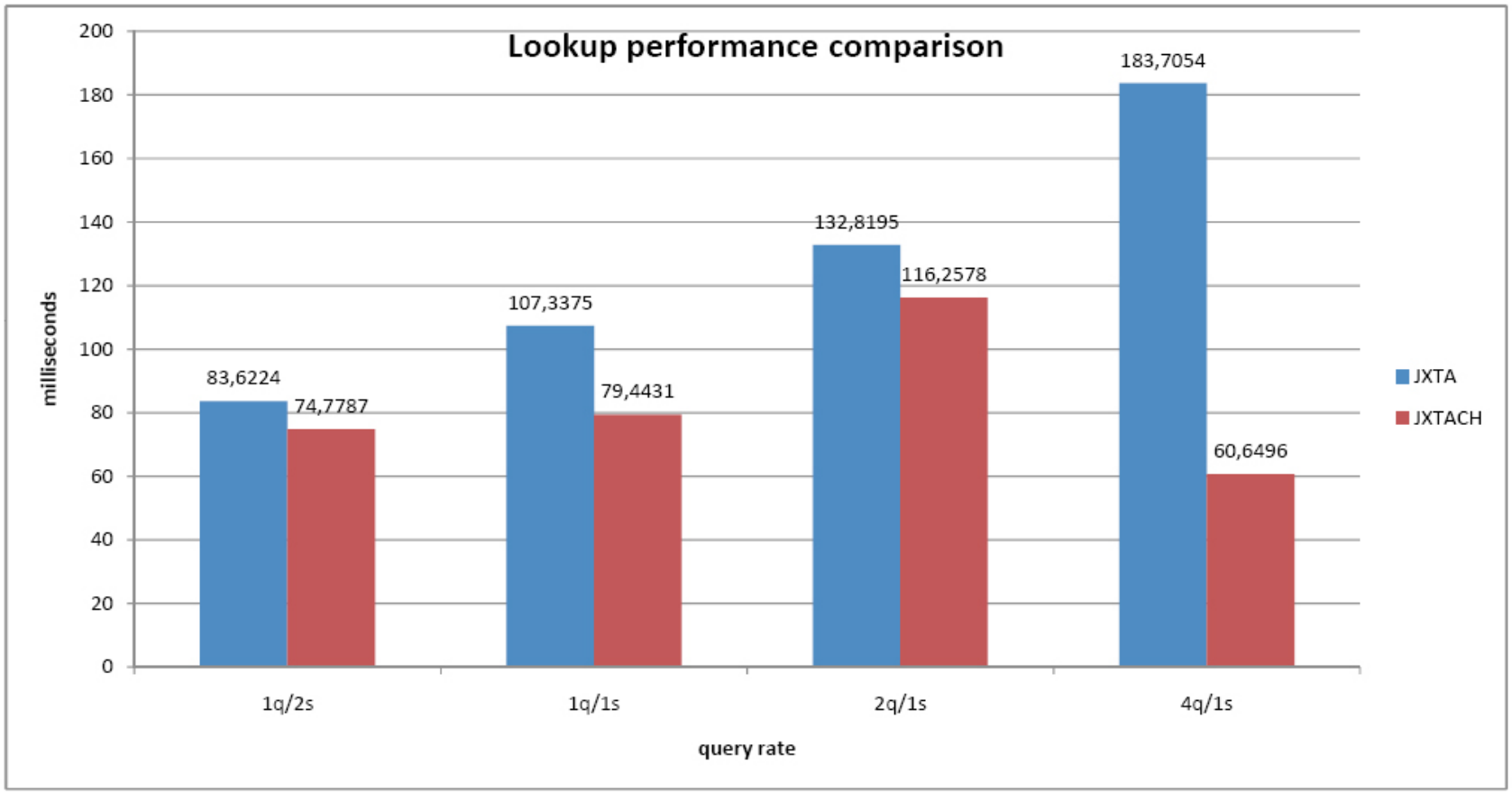
LAN tests results



Class C subnets of a class B subnet test settings

- Same structure for both protocol versions
- 8 computers spread in 3 class C subnets of the same class B net
- 1 edge peer publishing 500 advertisement (fake peer advertisements)
- 1 edge peer looking for each advertisement previously published at a variable query rate
- We measured the average lookup time

LAN tests results



Tests on transparency

- We performed a first transparency test with a simple JXTA application
  - A simple file sharing application
- The application works normally on both JXTA and JXTACh

Future steps

- Apply performance model of [5]
- Experiments with dynamic P2P network
- Same experiments in increasingly large environment
  - Run tests over the global Internet
    - AEOLUS European project
- Heavier transparency test
  - AEOLUS target application

Bibliography

1. JXTA Community Projects. `http://www.jxta.org`.
2. Project JXTA: Three Background Papers. `http://research.sun.com/spotlight/2005_10_12-JXTA.html`.
3. B. Traversat, M. Abdelaziz, E. Pouyoul. Project JXTA: A Loosely-Consistent DHT Rendezvous Walker, 2003.
4. L. Ricci. Peer to peer 2007/2008, `http://www.cli.di.unipi.it/doku/doku.php/p2p/`.
5. E. Halepovic, R. Deters. The JXTA performance model and evaluation, *Future Generation Computer Systems*, 21, 377–390, 2005.