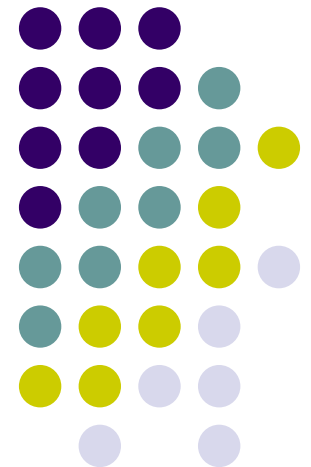


RandPeer: Membership Management for QoS Sensitive Peer-to-Peer Applications

Presented by Yin-Yeh Tseng

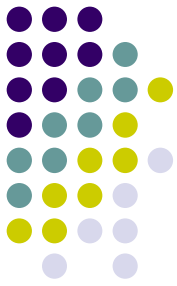
2006/07/20

INFOCOM 2006



Outline

- Introduction
- Application model
- RandPeer service
- Performance Evaluation
- Conclusion
- Discussion





Introduction

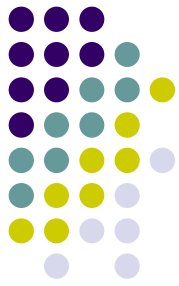
- Many recent P2P applications, including live media broadcasting, and audio conferencing require a high performance overlay structure.
- Building high performance overlays is an important task for these applications.



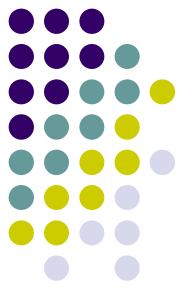
Related work

- Central Server
- Localized Approach
 - Rendezvous point
 - Gossip scheme

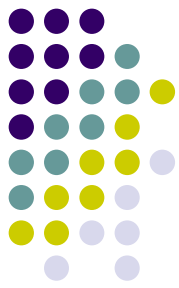
Gossip



- Every peer maintains a small list of other members in the system.
- They periodically exchange membership information with each other.
- The gossip scheme achieves scalability in membership management.

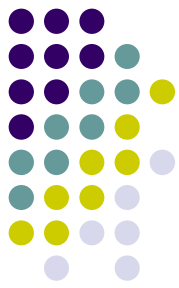


- The gossip style membership management was originally designed for probabilistic data dissemination applications.
- It is not clear how the scheme can be modified to accommodate different application Qos requirements.



RandPeer

- The paper designs a distributed membership service called RandPeer.
- RandPeer clusters peers based on their Qos characteristics.
- By restricting random peer selection within specific Qos clusters, RandPeer achieves Qos awareness in neighbor selection.



Application Model

- RandPeer provides support by decoupling membership management from P2P applications.

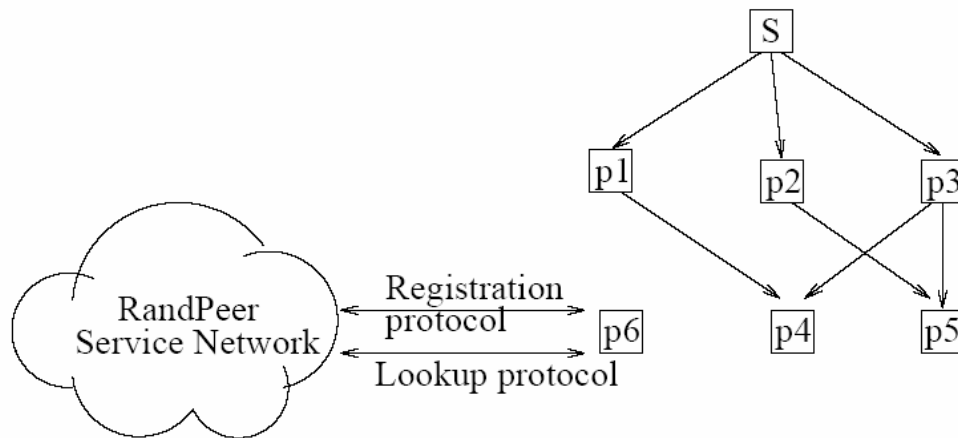
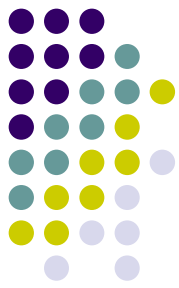
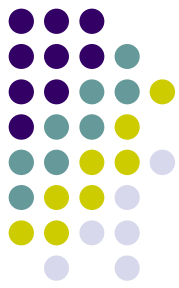


Fig. 1. App. scenario that uses RandPeer



RandPeer Service

- RandPeer uses a distributed trie data structure to organize the membership information, which is in turn mapped to an underlying distributed hash table (DHT).



Membership Trie

- A trie is basically a tree with its nodes labeled with 0, 1 strings.
- Each node in the trie is called a “bin”.
- Peer registration information is only stored at leaf bins.
- Each bin has a label.



- The maximum length of a label is h .
- A peer has a peer id of b bits ($b \geq h$).
- A peer should register with the leaf bin whose label is a prefix of its peer id.

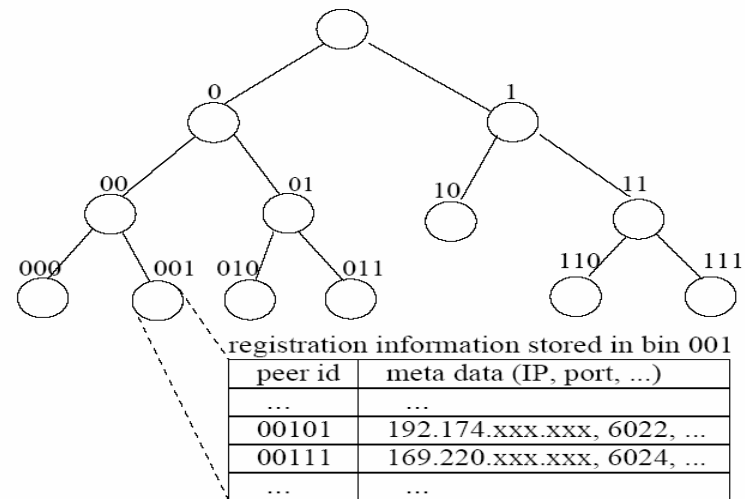
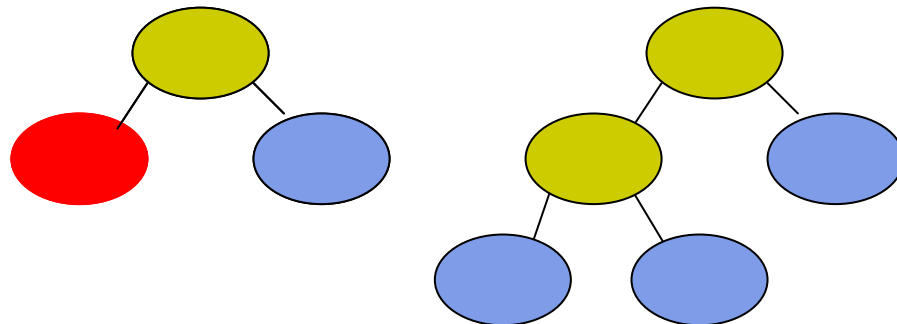
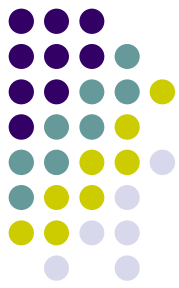


Fig. 2. Membership trie



- Each leaf bin has a capacity B .
- If there are more than B peers registered with a leaf bin (with a label ℓ), the bin can be split into two leaf bins (with labels $\ell 0$ and $\ell 1$).
- If the total number of peers registered with $\ell 0$ and $\ell 1$ drops to below $B/2$, the two leaf bins can be merged.





- RandPeer is built on top of DHT, therefore we use consistent hashing to map the trie to the RandPeer service nodes.

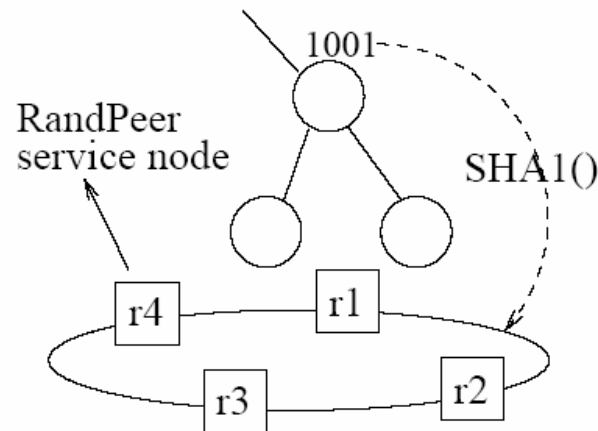
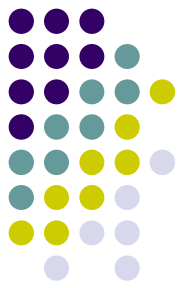


Fig. 3. Mapping from membership trie to RandPeer nodes

Qos Aware Neighbor Selection

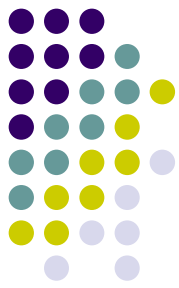


- ReedPeer combines application specific Qos metrics with peer id.
- The id of a peer is divided into a QoS prefix and random suffix.

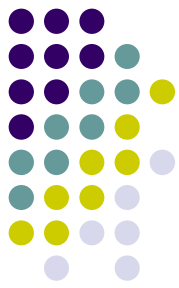
00100

QoS prefix

random suffix

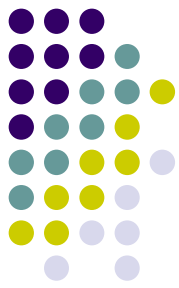


- If two peers have the same QoS characteristics, they will be automatically clustered under the same subtree.
- A peer can find some other peer by generating a random key with the desired QoS prefix.

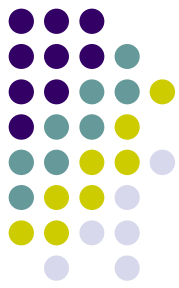


Registration Protocol

- RandPeer takes a soft state approach to membership management.
- A peer send `Register` messages to its leaf bin periodically.
- The leaf bin is expected to send a `RegisterOK` message back.



- Each `Register` message contains the peer id of a peer and some meta information such as its IP address and port number.
- Each `RegisterOK` message contains the meta information previously registered under the peer id and the number of registration in the bin.



Bin Split/Merge

- Each leaf bin has a `leader peer`, which is the peer with the smallest peer id in the bin.
- The leader peer is responsible for initiating the bin split/merge process.

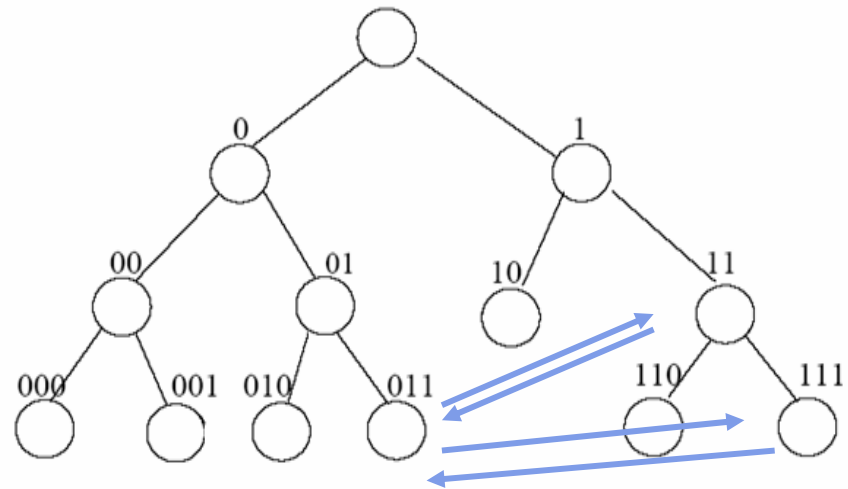


Peer Lookup

- The query peer sends a Lookup message to a bin whose bin label is a prefix of the lookup key and has a length (label_bits) of $h/2$.

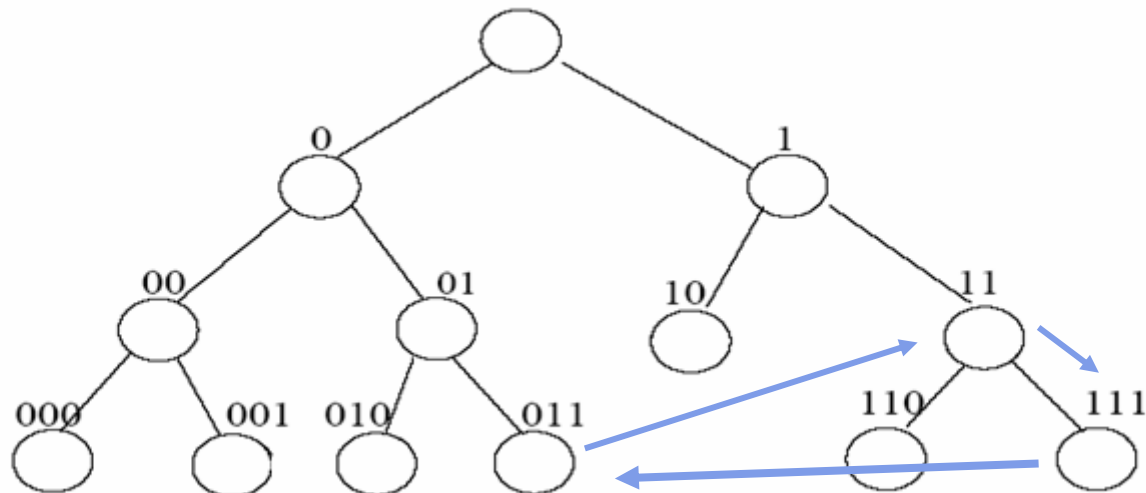
- LookupOK
- LookupGoDown
- LookupGoUp

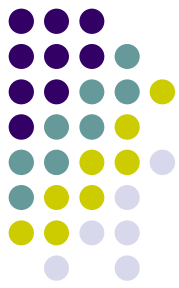
key:1110





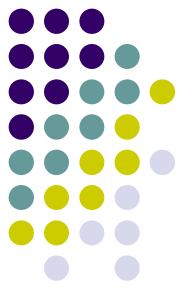
- Let each interior bin record the minimum height of its left and right subtrees.





- A peer is returned if its id immediately follows a lookup key.
- To improve the randomness of the lookup result, we lookup the m peers whose ids immediately follow the lookup key, we then randomly choose one from the m peers to be the final random peer.

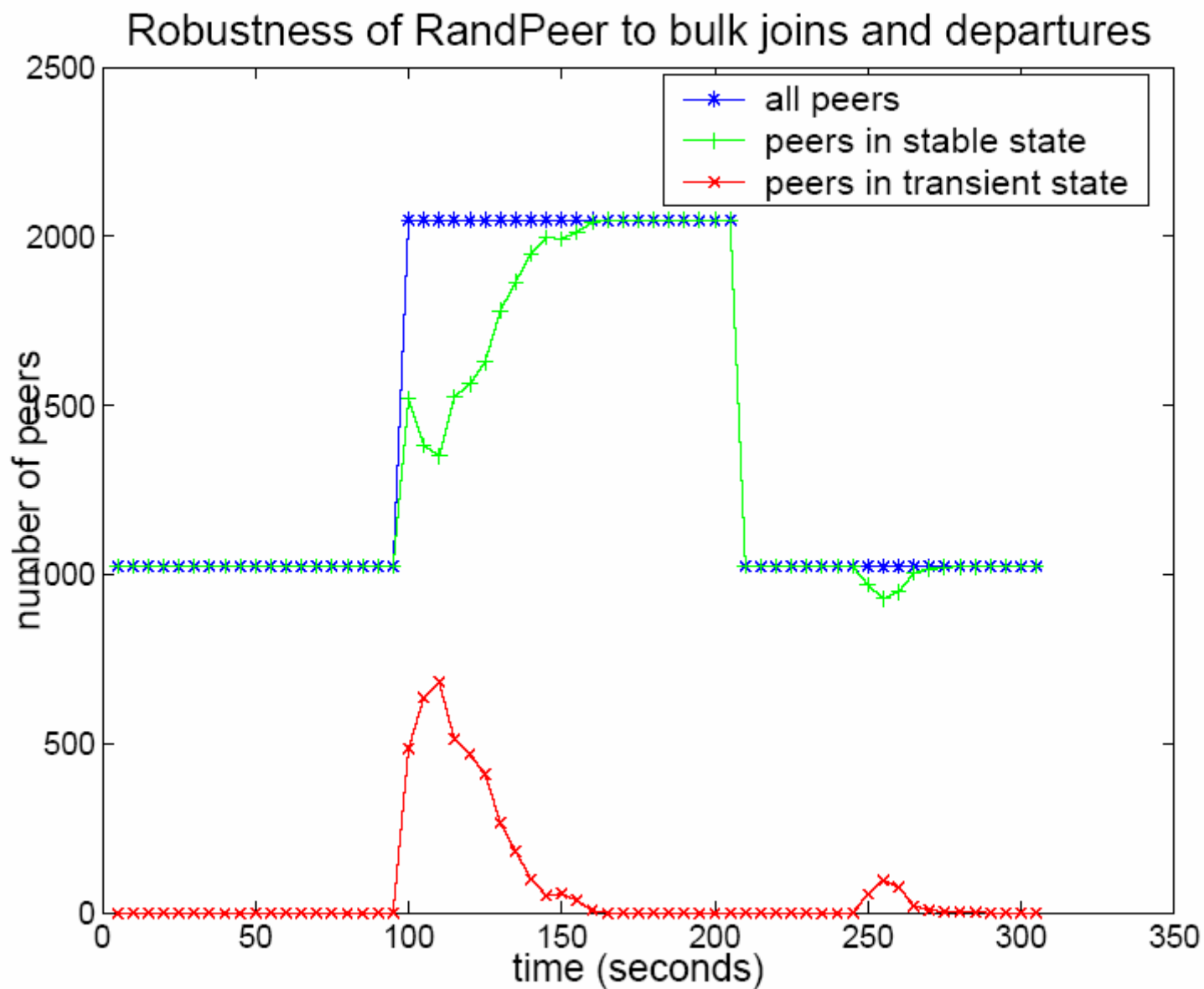
m : random ratio

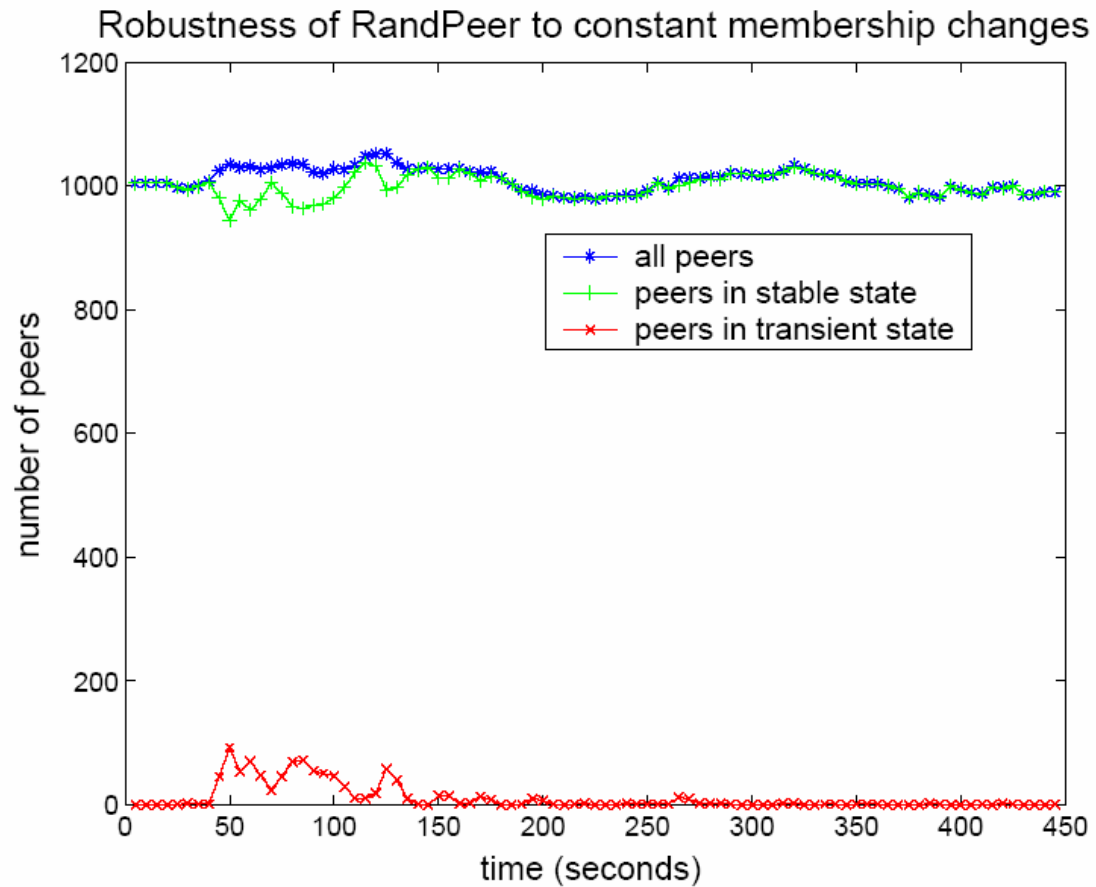
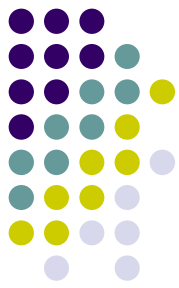


Performance Evaluation

- robustness of RandPeer to highly dynamic P2P membership
- response time
- the impact of Qos aware neighbor selection on the performance of P2P overlays.

- This paper implemented RandPeer on top of the Chord.





Robustness of RandPeer to constant joins and departures (churns)

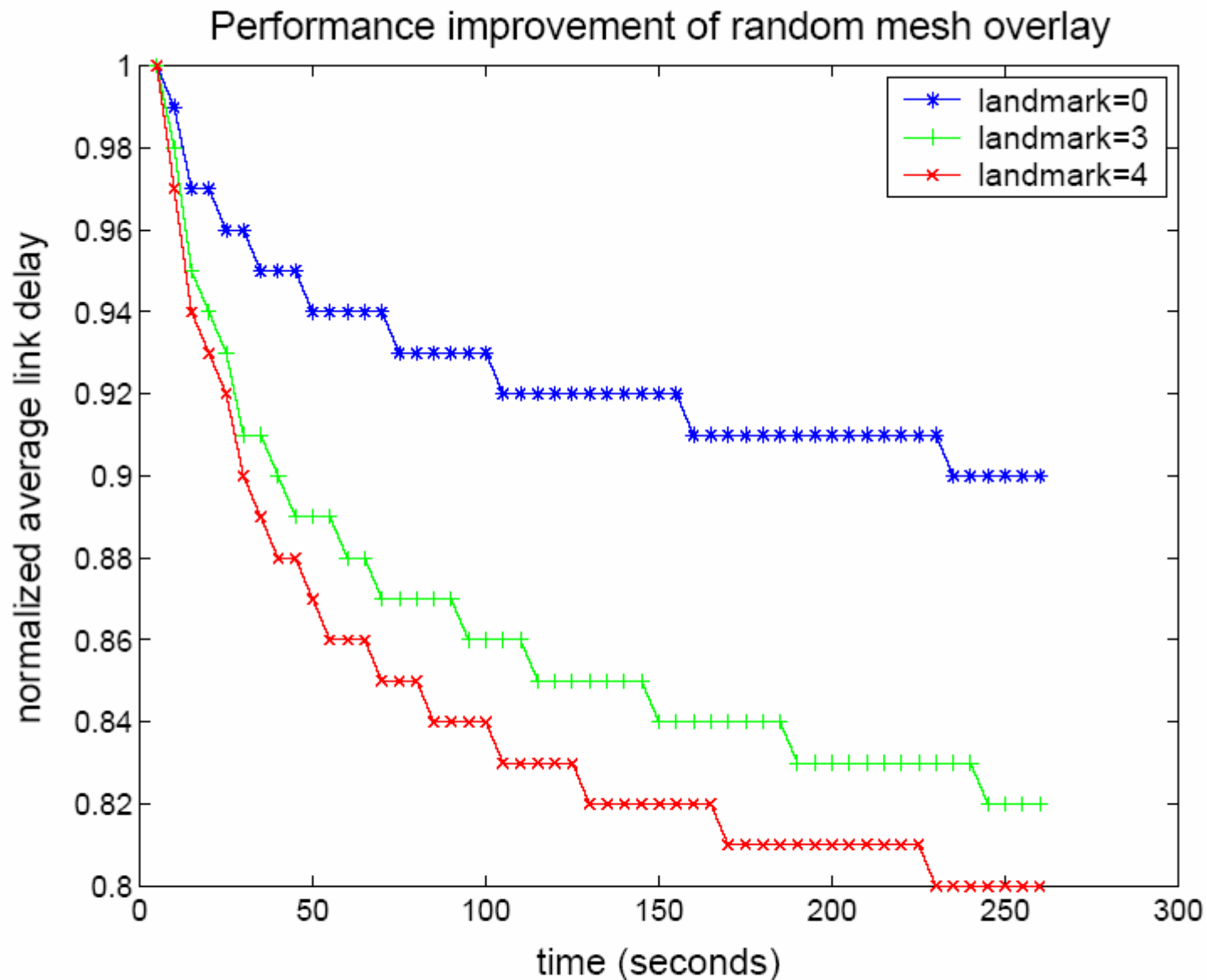
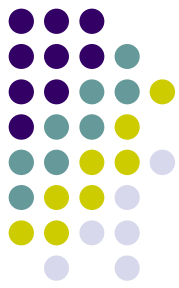


Fig. 13. Performance improvement for an application with 4096 peers



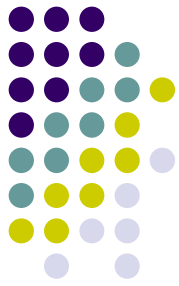
Conclusion

- RandPeer achieves scalable and decentralized membership management by using a trie data structure and map the trie to a DHT.
- It enables Qos aware neighbor selection by clustering application peers based on their Qos characteristics.



Discussion

- Decoupling membership management from P2P applications.
- Integrating RandPeer service networks with P2P application networks.



Thank you 😊