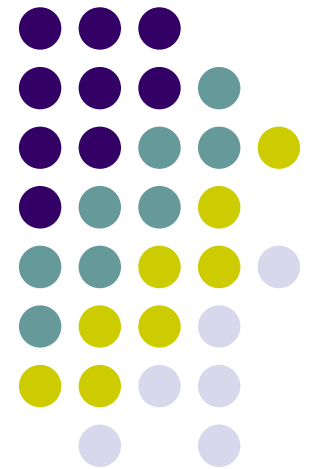
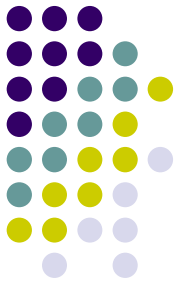


A Fair Resource Allocation Algorithm for Peer-to-Peer Overlays

Presented by 曾胤燁
IEEE INFOCOM2005



Outline



- Introduction
- Fair resource allocation algorithm
 - Fairness index
- Performance evaluation
- Conclusion
- Discussion



Introduction

- Fair resource allocation
 - Load balance
 - Data distributed allocation
- Advantage:
 - Improve resource usage
 - Minimize network latencies



Unfair??

- Freerider :user that consume resources of the system without contributing anything in return.
- Problem:
 - Decrease system performances
 - Bottleneck



- How can we implement a fair resource allocation scheme to distribute the data in an unstructured peer-to-peer system?
- Distribute the data in a fair and fully decentralized manner



Approach overview

- I_{pi} the load of object i : the total volume of data (in bytes) transmitted by p .
- I_p the load of peer p : the sum of the loads $\sum_i I_{pi}$
- P_{neig} : the neighborhood of peer p : the nodes that are up to 1 hop away from p
- Every node keeps the load of each peer in its neighborhood.

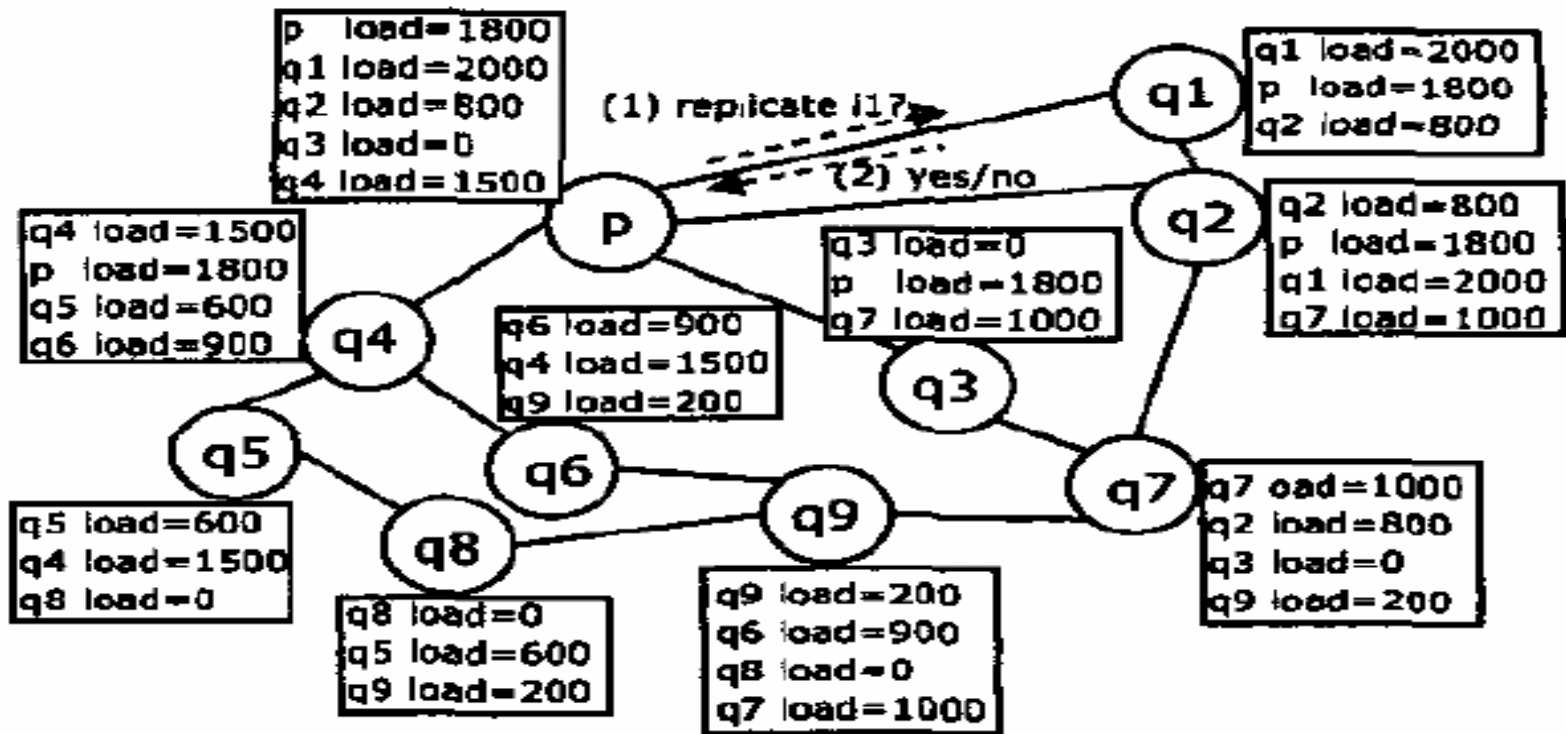
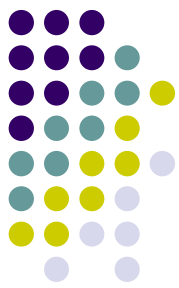


Fig. 1. The model of our system



Fairness index

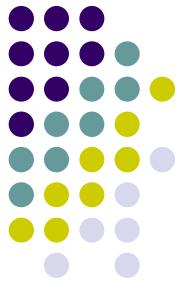
- Use the fairness index to measure “how equal” are the loads of the peers.

$$\mathcal{F}(\bar{l}) = \frac{(\sum_{p \in N} l_p)^2}{|N| \cdot \sum_{p \in N} l_p^2}$$



- The fairness index value ranges between values 0 and 1.
- The higher the value of the fairness index, the more fair the distribution is.

$$\mathcal{F}(\bar{l}) = \frac{(\sum_{p \in N} l_p)^2}{|N| \cdot \sum_{p \in N} l_p^2}$$



- The challenge in our scheme is to achieve a fair load distribution using only local information and making only local decisions at the nodes.



- Compute the load distribution $\mathcal{F}(\bar{l})$ and the average load $avg_{\bar{l}}$ in their neighborhoods.

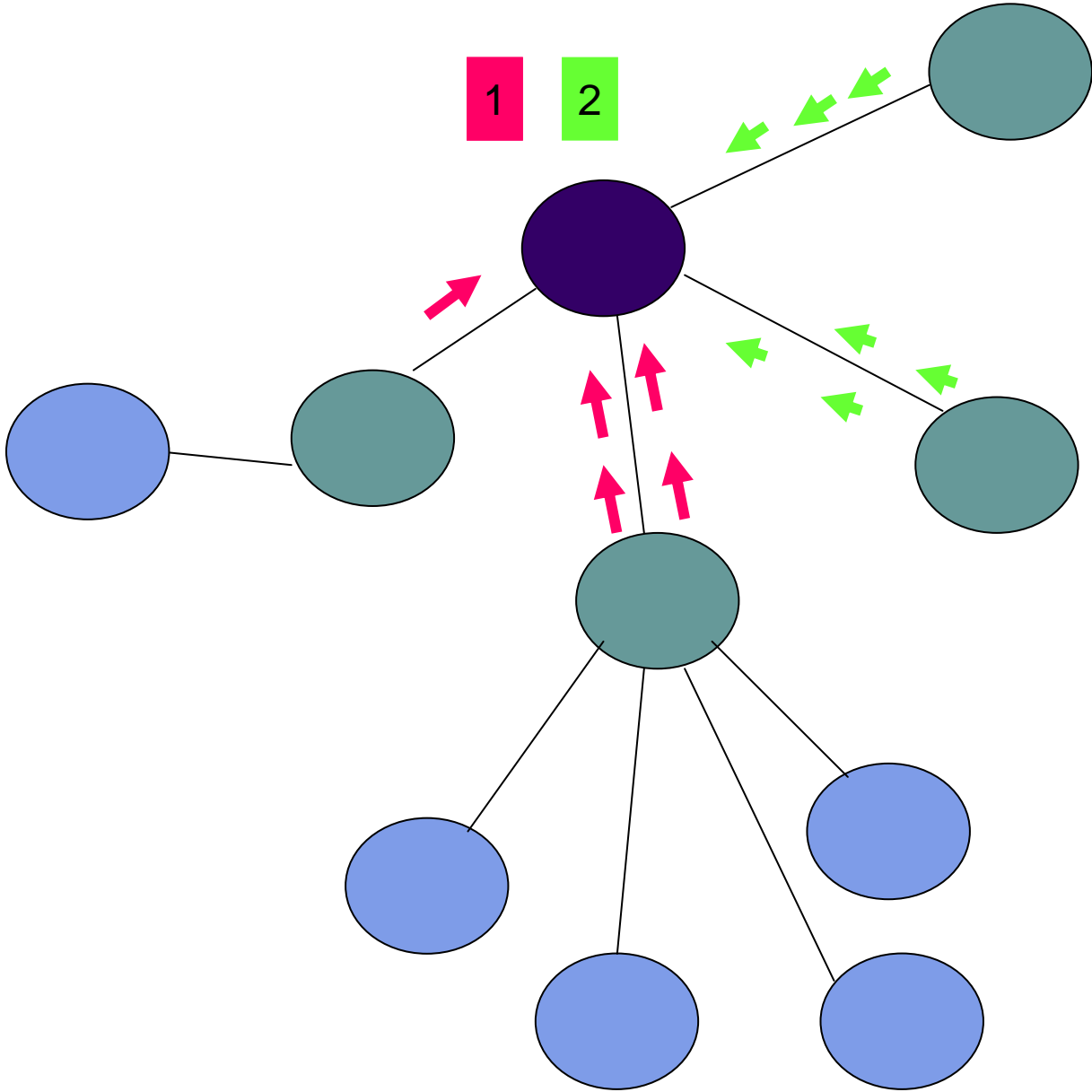
$$\mathcal{F}(\bar{l}) = \frac{(\sum_{q \in p_{neigh}} l_q)^2}{|p_{neigh}| \cdot \sum_{q \in p_{neigh}} l_q^2}$$

$$avg_{\bar{l}} = \frac{\sum_{q \in p_{neigh}} l_q}{|p_{neigh}|}$$



Fair resource allocation algorithm

- Compute the load $F(\ell)$ and the average load avg_i of its neighborhood.
- If the $F(\ell)$ is lower than a threshold τ_{neigh} and node's p load is higher than the average load of the neighborhood, node p selects an item to replicate to another node.



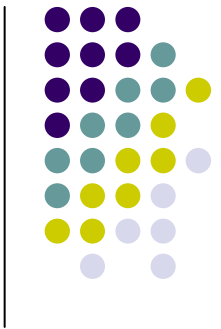
1

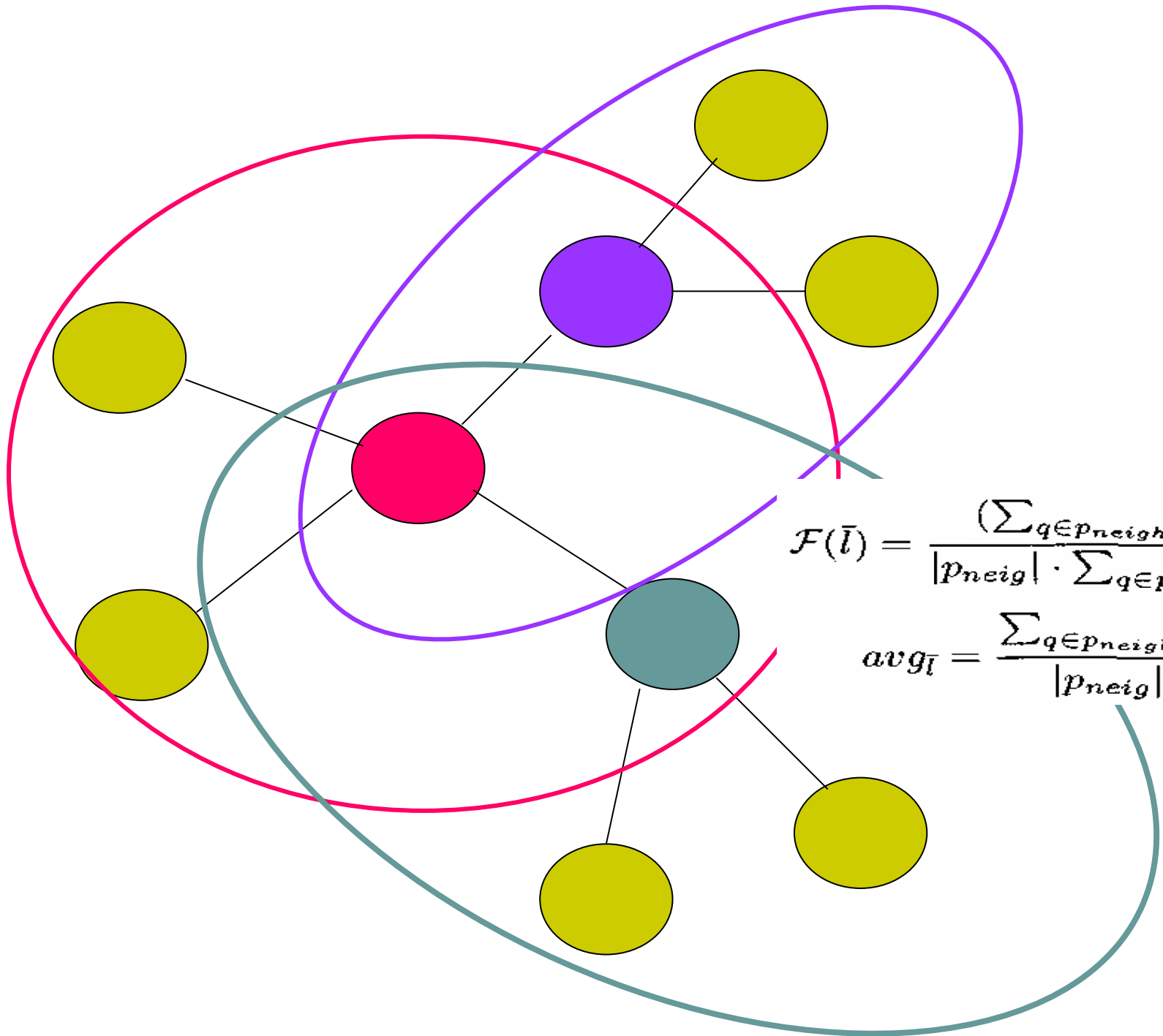
2

1

2

5 : 7





$$\mathcal{F}(\bar{l}) = \frac{(\sum_{q \in p_{neigh}} l_q)^2}{|p_{neigh}| \cdot \sum_{q \in p_{neigh}} l_q^2}$$

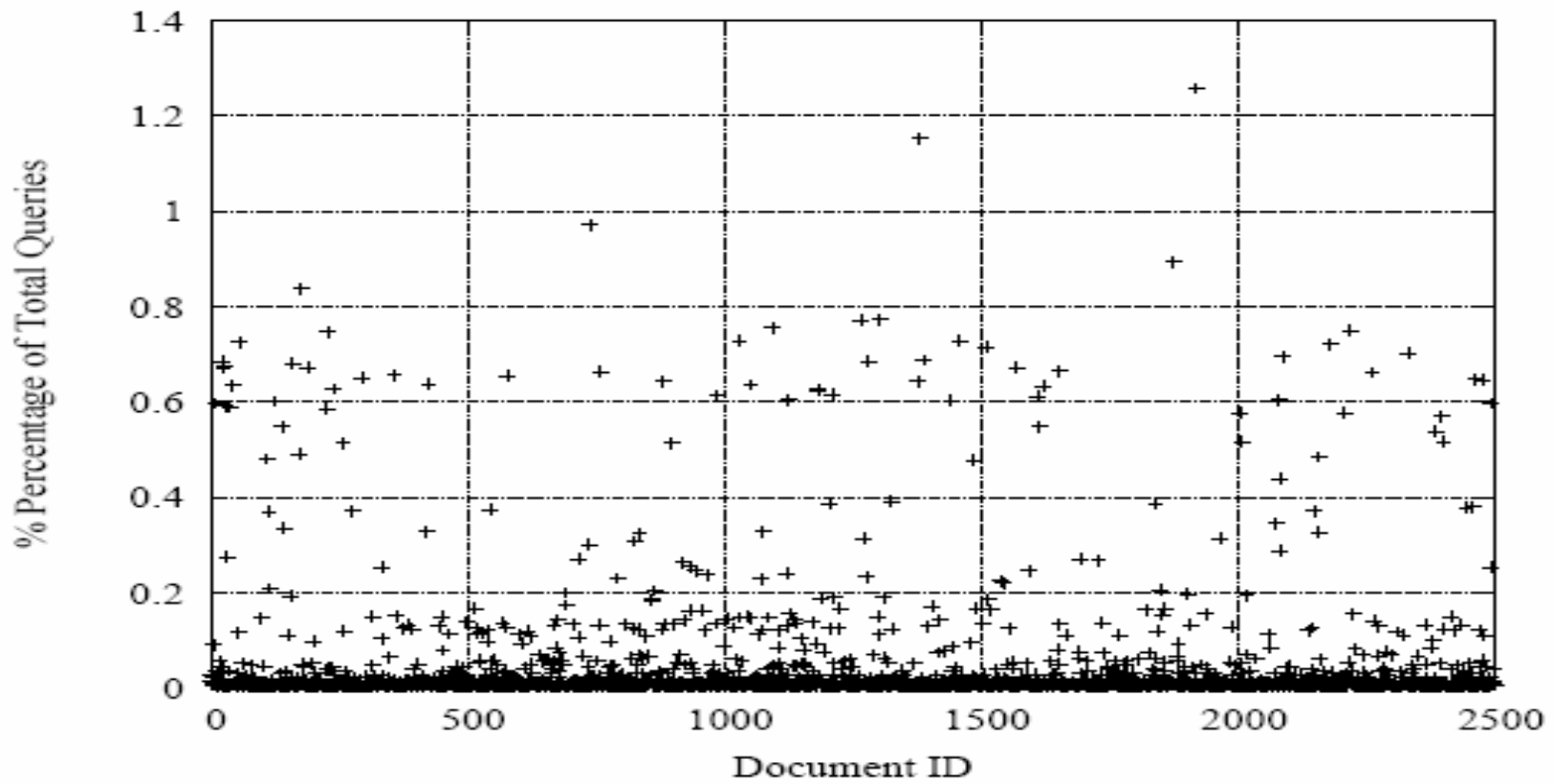
$$avg_{\bar{l}} = \frac{\sum_{q \in p_{neigh}} l_q}{|p_{neigh}|}$$



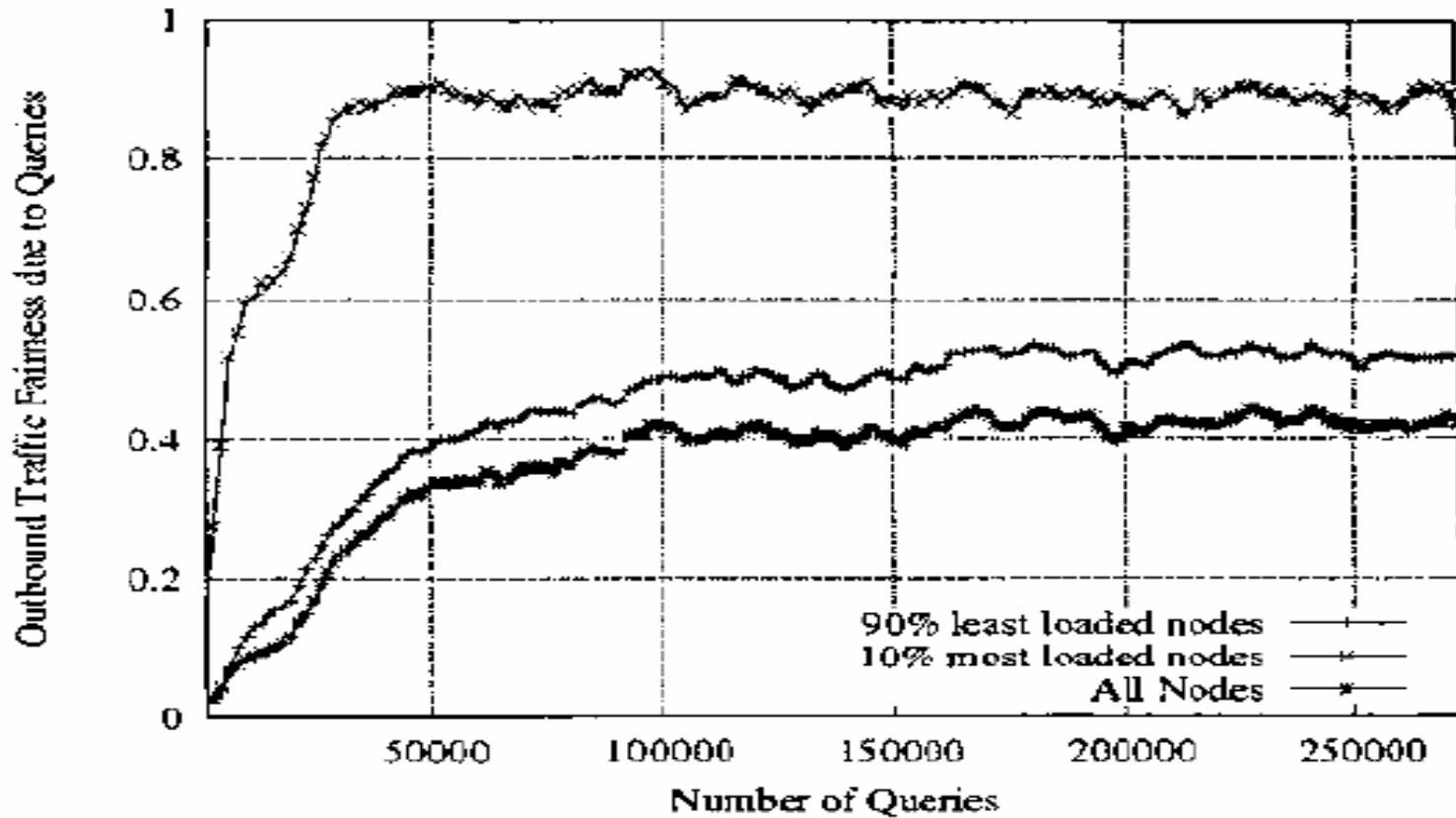
Performance evaluation

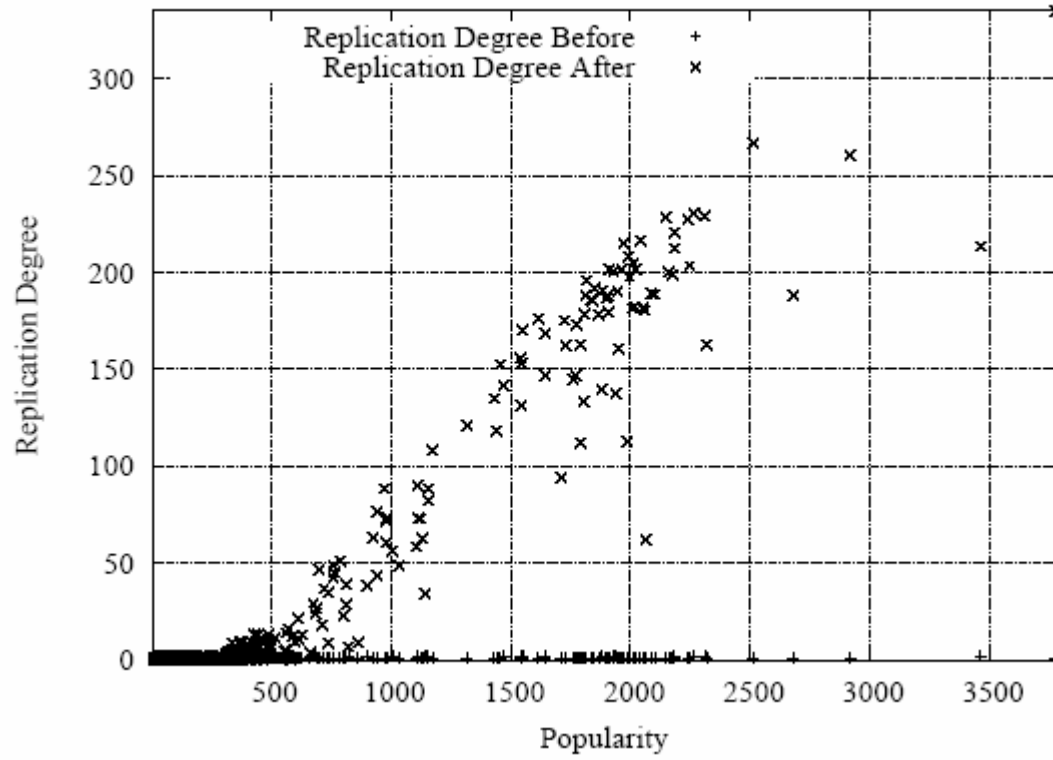
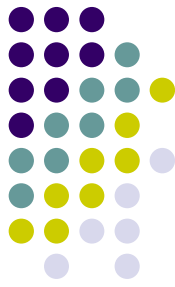
Environment :

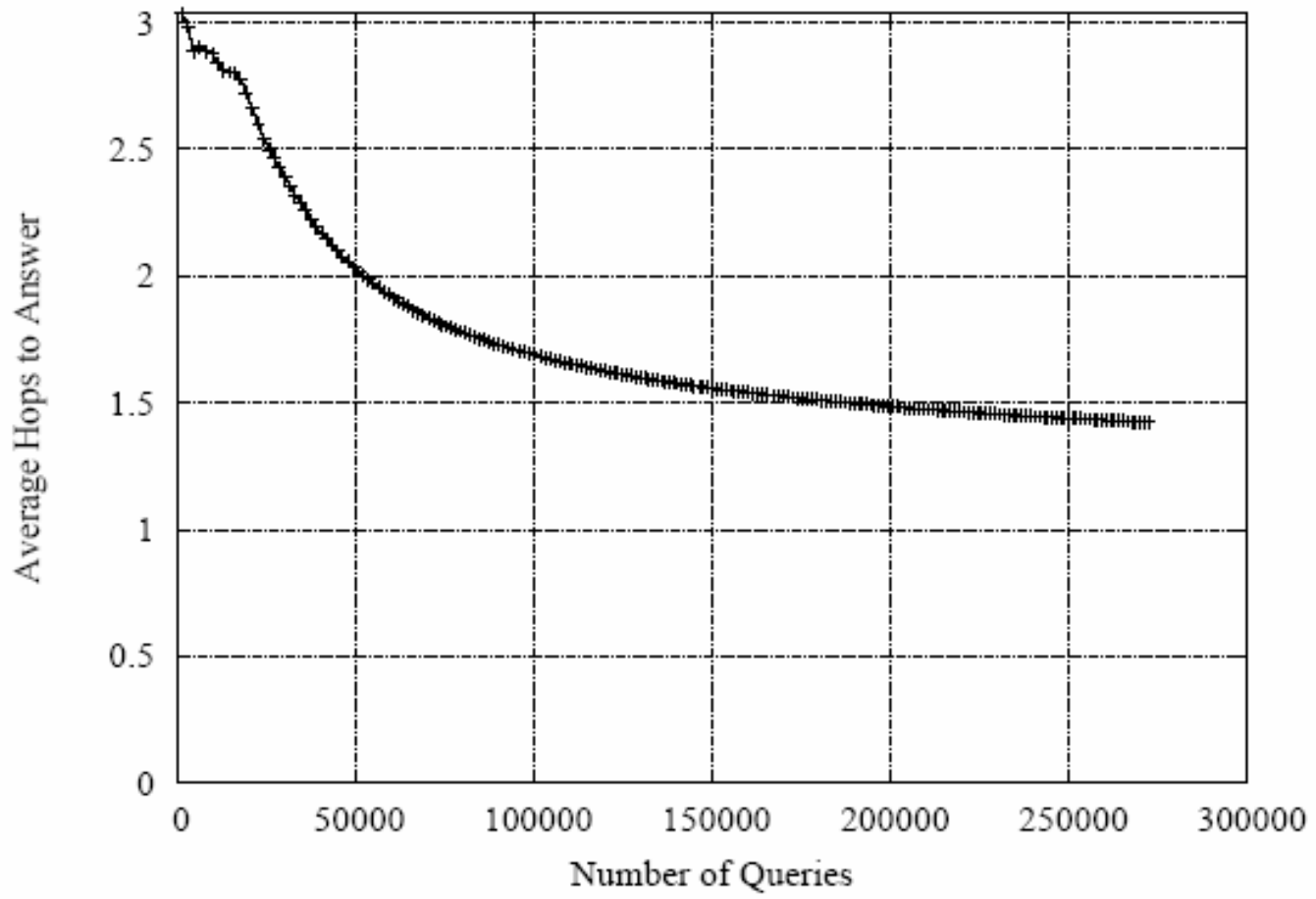
- 1000 nodes
- 2500 documents
- 280000 requests
- Random graph
- Average node capacity : 50KB
- The sum of the size of all items: 20MB
- The threshold τ_{neigh} :0.45



Performance evaluation









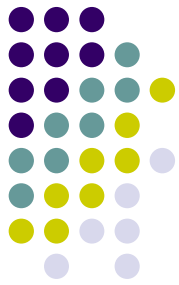
Conclusion

- In this paper they presented a technique that solves fair resource allocation problem in a decentralized manner.
- The simulation results show that the technique achieves fairness in load balance distribution, reduces the number of hops to find the documents and highly replicates the popular documents in the system.



Discussion

- Dynamic peer connections and disconnections
- Better replication selection scheme (more hops away?)
- Load balance for p2p applications.



Thank you 😊