An Interest-based Architecture for Peer-to-Peer Network System

Presented by Chi-Hong Chao

Outline

- Introduction
- Related Work
- System Architecture
- Simulation
- Conclusion

Introduction

- Peer-to-peer file sharing system.
  - Centralized (Napster)
  - Decentralized
    - Unstructured (Gnutella)
      - Flooding based
    - Structured (CAN, Chord, Pastry)
      - DHT (Distributed Hashing Table) based

Introduction

- Flooding based
  - Trivial
  - Not scalable
- DHT based
  - Scalable
  - Sensitive to node failure
  - Hard to support keyword search.

Related Work

- Data replication
- Selective search
- Cluster
- Interest group

Related Work – Data Replication

- Data replication is a technique to improve the effectiveness of flooding, because sharing files are replicated among the peer-to-peer network system so that the flooding range is reduced.
- With these algorithms, the search scope can be reduced, because of its explicit control of placement of data items that can be easily located.

Related Work – Selective Search

- Random walks
  - forwards a query message to randomly chosen k neighbors instead of sending out all
- Routing Indices
  - Routing Indices (RIs) are used to guide the queries toward where the queries are more likely to be satisfied.

Related Work – Cluster

- Cluster is a simple and useful method to restrict the flooding of query messages.
- The number of peers in a cluster is often limited in order to restrict searching range. If the number of peers in a cluster is too large, it is divided into several clusters.
- Each cluster selects a cluster header by a header selection algorithm. And the cluster headers together can further be connected. Hence the whole peer-to-peer network has a hierarchical structure with this cluster strategy.

Related Work – Interest Group

- In [13], the locality embedded in human interests effectively guides search queries.
- The set of peers satisfied with the same guide rule should contain data items that are similar.
- In [14], metadata are used to describe and represent documents that nodes share with others. Metadata can be simply defined as data about data.

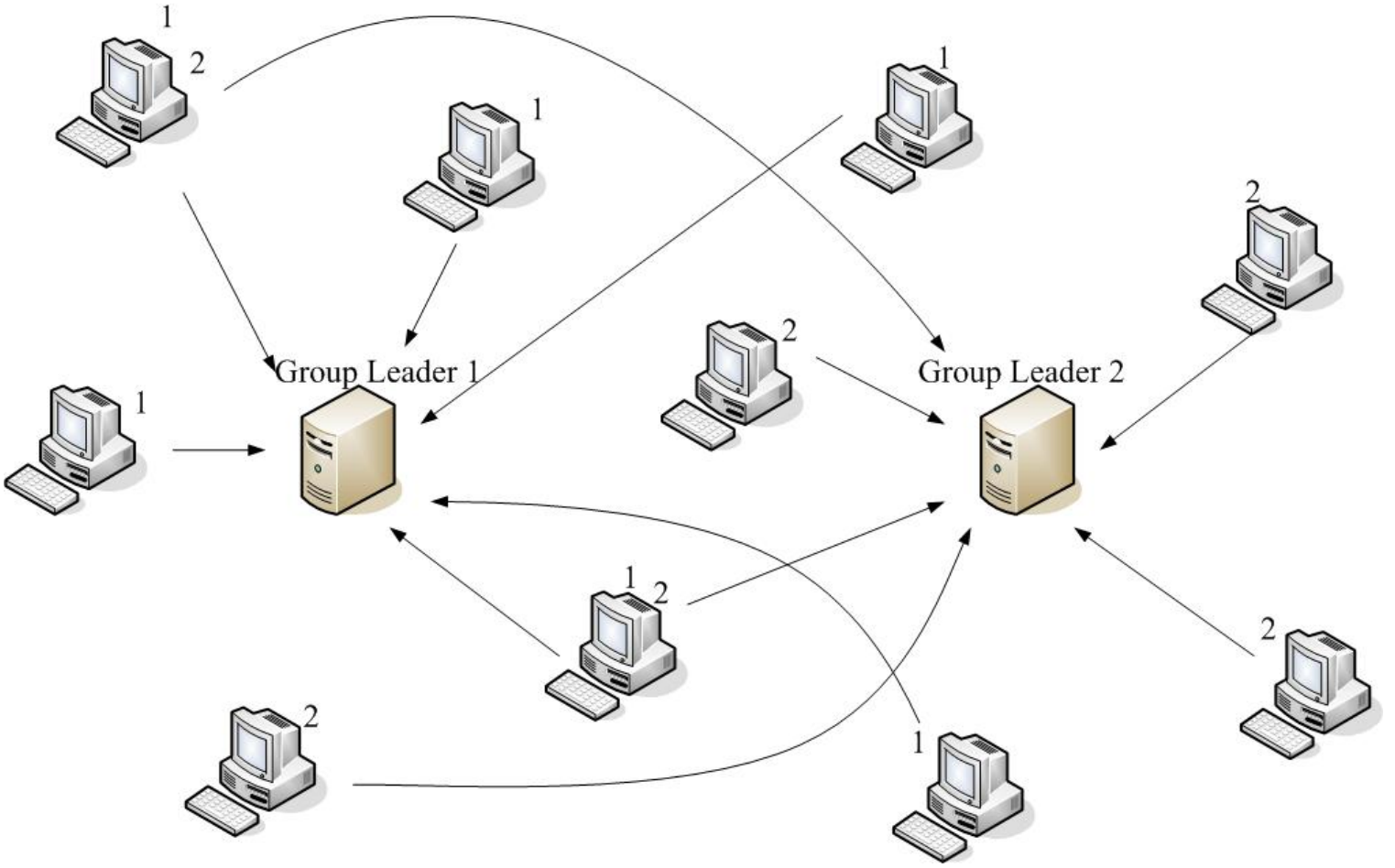
System Architecture

- Our proposed system is constructed as a group-based architecture.
- The locality of user interests is the key on grouping peers in the system.
- What a peer is interested in depends on both the types of shared files and the interest profile, which is explicitly configured by the users.
- Peers whose interests are similar form an interest group and each group in this system is an unstructured overlay.

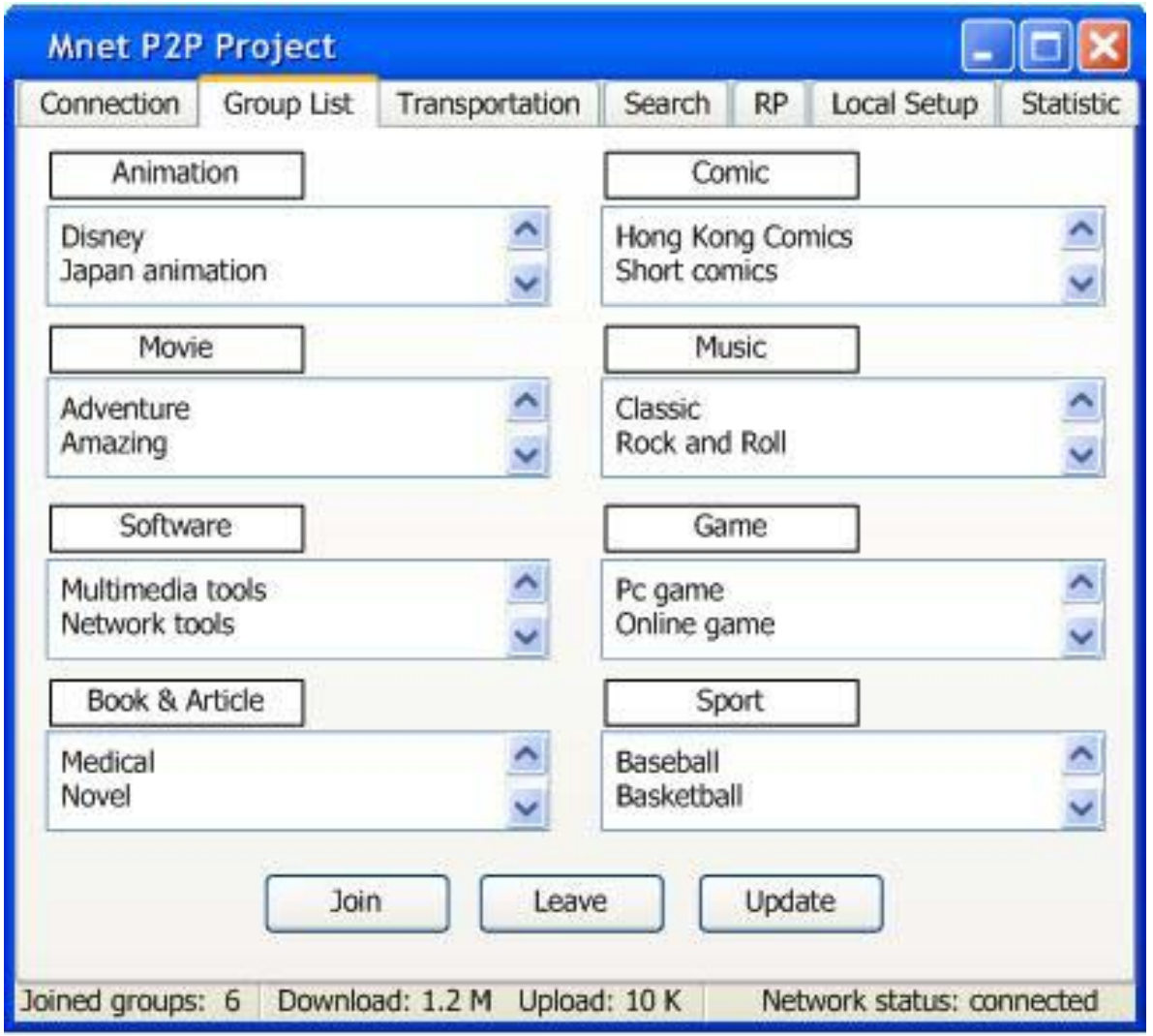
System Architecture

- A peer with capable capacity on computation and bandwidth is selected as the group leader.
- Group leaders maintain connection status of peers in the group and handle the join process of new peers.
- A backup leader is selected and synchronized with the primary leader for robustness purpose.
- Group leader also plays an important role in the searching process.

System Architecture



System Architecture - Interests



System Architecture - Interests

- All files shared by a peer are categorized into most suitable categories automatically based on the metadata in the files.
- A peer joins groups according to the number of shared files in each category.
- A peer may join several groups, and the groups in our proposed architecture are overlapped.
- Although the groups are always overlapped, all the operations such as join, leave, search and download are performed in different groups simultaneously and independently.

System Architecture – Group Leaders Selection

- A peer with capable capacity on computation and bandwidth is selected as the group leader or the backup leader.
- The criterion of scoring peer's capacity is defined by this equation:

$$CB = CPU \times 1 + MEM \times 2 + BAND \times 7$$

CB: the Computation and Bandwidth score

CPU: is computational ability of a peer

MEM: is size of memory of a peer

BAND: is network bandwidth of a peer

System Architecture – Group Leaders Selection

- 1). P_1 is the first peer joins the group, and P1 becomes the group leader undoubtedly.
- 2). P_2 is the second peer joins the group and gets the CB value of group leader. After comparing the CB value between leader and P2, the peer with highest CB score becomes the group leader and the other one becomes the backup leader.
- 3). P_i is the i-th peer joins the group and gets the CB value of group leader and backup leader. After comparing the CB value among group leader, backup leader and Pi, the peer with highest CB score becomes the group leader. The second becomes the backup leader and the other one becomes the normal peer.

System Architecture – Peers join and connection management

- Each peer has a unique identifier, usually its IP address, and maintains two caches: a neighbor cache (nCache) and a member cache (mCache).
- nCache contains a list of neighboring peers in each group it joins, while mCache contains partial list of the peers in each group it joins.
- Rendezvous Point (RP) is employed in the bootstrapping process of new peer in the interest group peer-to-peer network system.
- RP records all the information of interest groups, such as the IP address of the group leaders and the descriptions of the groups.

System Architecture – Peers join and connection management

- While a new peer P_i is to join the interest group peer-to-peer network system, it first sends a query message to RP to obtain a list of group leaders.
- Suppose that P_i is interested in group j, and L_j being the group leader of group j. The join procedure is as follows:

System Architecture – Peers join and connection management

- 1). P_i first sends a query message to RP to get the Group_Leader_List that contains IP addresses of group leaders in this system.
- 2). P_i sends joining messages to the group leader (L_j) of groups that P_i wants to join. If group j that P_i wants to join has no member, P_i becomes the group leader of group j and jump to step 5.
- 3). L_j randomly selects some deputy nodes from its nCache and redirects P_i to these deputies.
- 4). P_i gets lists of neighbor candidates from these deputies' mCache, and selects some of them as its neighbors to establish connections.
- 5). End of joining procedure.

System Architecture – Peers join and connection management

- To accommodate overlay dynamics, each peer in a group periodically floods an alive message to announce its existence.
- When a peer receives an alive message which is not sent by its direct neighbors, it will update the information in its mCache.
- If a peer finds that any of its neighbors doesn't send the alive message after a period, it removes that neighbor from its nCache and tries to find a new neighbor from its mCache.
- nCache always keeps alive peers as neighbors, while mCache only caches peers that refresh their existence recently and providing a second chances for a peer to find other peers in the network.

System Architecture – Peers Disconnect and Recovery

- Regular Departure
  - normal peer

    It sends a DEPARTURE message to all connected peers. On receiving a DEPARTURE message, the neighbors update their nCache and choose a new neighbor from its mCache.
  - group leader

    It notifies the backup leader to be the new leader of this group and then floods the DEPARTURE messages in the group it belongs to and sends the DEPARTURE message to the other leaders in the system. On receiving the DEPARTURE message of group leader, a peer will re-connect with the new group leader. A new backup leader is then selected among capable peers in this group. Finally the new leader contacts the RP for registering new primary and backup leader.

System Architecture – Peers Disconnect and Recovery

- Failure
  - When a normal peer fails due to system crash or network disconnection, the failure can be easily recovered by nCache and mCacheIf. The first peer that discovers the failure will issue a failure message which contains the identifier of failed peer, so that all the peers among the group can update its mCache and nCache for the failed peer.
  - However, on noticing the failure of the group leader, any peer, regardless of normal peer or leader, will notify RP to recover the failure. RP will inform the backup leader of the failed group and all other group leaders in the system re-connect to the new leader. Properly, the process of secondary leader selection must be performed again.

System Architecture – Searching and Download Scheduling

- Searching
  - Before sending a query message, a peer must choose target group that is what the peer wants to send query messages to.
  - If the querying peer is the member of the target group, it may just flood the query message in the group. In another case, if the peer is not the member of the target group, the peer can't send the query message to that group directly. So it may send the query message to its group leader, and the group leader will send the query message to the group leader of target group, so that the query message can be flooded in the target group.

System Architecture – Searching and Download Scheduling

- Multiple Interests searching
  - Suppose a peer Pi sends a query message M(q, $a \cup b \cap c$) where q is the keyword of this query and Ga, Gb, Gc are groups interests corresponding to interests a, b, and c.
  - The procedure of search is as follows:
    1). Pi issues a query message M(q, $a \cup b \cap c$)
    2). M is routed to Ga, Gb, Gc and return the corresponding search result Ra, Rb and Rc.
    3). Pi gets the search result $R=Ra \cup Rb \cap Rc$.

System Architecture – Searching and Download Scheduling

- Download scheduling algorithm
  - each file is divided into segments.
  - In this algorithm, it first gets the number of source peers that offer the file we need after sending a query message. Then it has to perform an extended search mechanism, which is extending the source peers by asking each of the source peers what peers are also downloading this file currently.
  - Since a segment with less supplying peer may cause a download fail in the dynamic and heterogeneous network, the download algorithm it performs is downloading the rare segments first. Then, among the source peers, the uploading bandwidth of the source peers is also concerned; the one with rare segments and the highest uploading bandwidth is selected first.

Definition:

source_set: the peer set of search result;
ex_source_set: the result of extended search;
band(j): uploading bandwidth of peer j;
expected_set: set of segments to be fetch;
num_source: number of multiple sources;
seg_size: segment size;
file_size: downloading file size;
popularity[i]: number of supplying peer of segment i

Algorithm:

```
num_source = (file_size) / (8*seg_size);
For( segment i within expected_set )
   For( peer j within source_set∪ex_source_set )
         If( peer j has segment i )
               popularity[i]++;
   End For j
End For i
For( n=1 to num_source  )
   x = select_lowest_popularity_seg( popularity[i]);
   For( peer j within source_set∪ex_source_set)
         If( peer j has segment x )
               If( band(j)> supplier[x] )
                     supplier[x]=j;
   End For j
End For n
```

Output:

Supplier[x]: supplier peer of segment x;

System Architecture – Load Balancing

- Ranking
  - Each peer has to record the requests sent by it and the responses of the other peers. According to these records, we can calculate the hit ratio of the other peers. We define RANKING(X) as the hit ratio of peer X.
  - After a peer joining a group, we perform the ranking mechanism in that peer. After a period of time, that peer will get a list of peers sorted by the hit ratio.

System Architecture – Load Balancing

- Group Divider
  - If a group leader detects that the loading of the group is excessively high (too many members), it will determine that this is an overdeveloped group.
  - The group leader will notice the secondary leader to become the group leader of subgroup and some of the group members will connect to the subgroup.

System Architecture – Load Balancing

- Data Replication
  1). P_i receives a request R_j and floods it out.
  2). If Rj has been requested over 10 times in 60s, Pi will send a request the same with Rj to download these popular files and clear this request from the request_record data structure.
  3). If Rj has not been requested over 10 times in 60s, Pi will just record Rj.

Simulation

$$ISR = \frac{M\_group}{M\_all}$$

- Where

  ISR: Interest Search Ratio

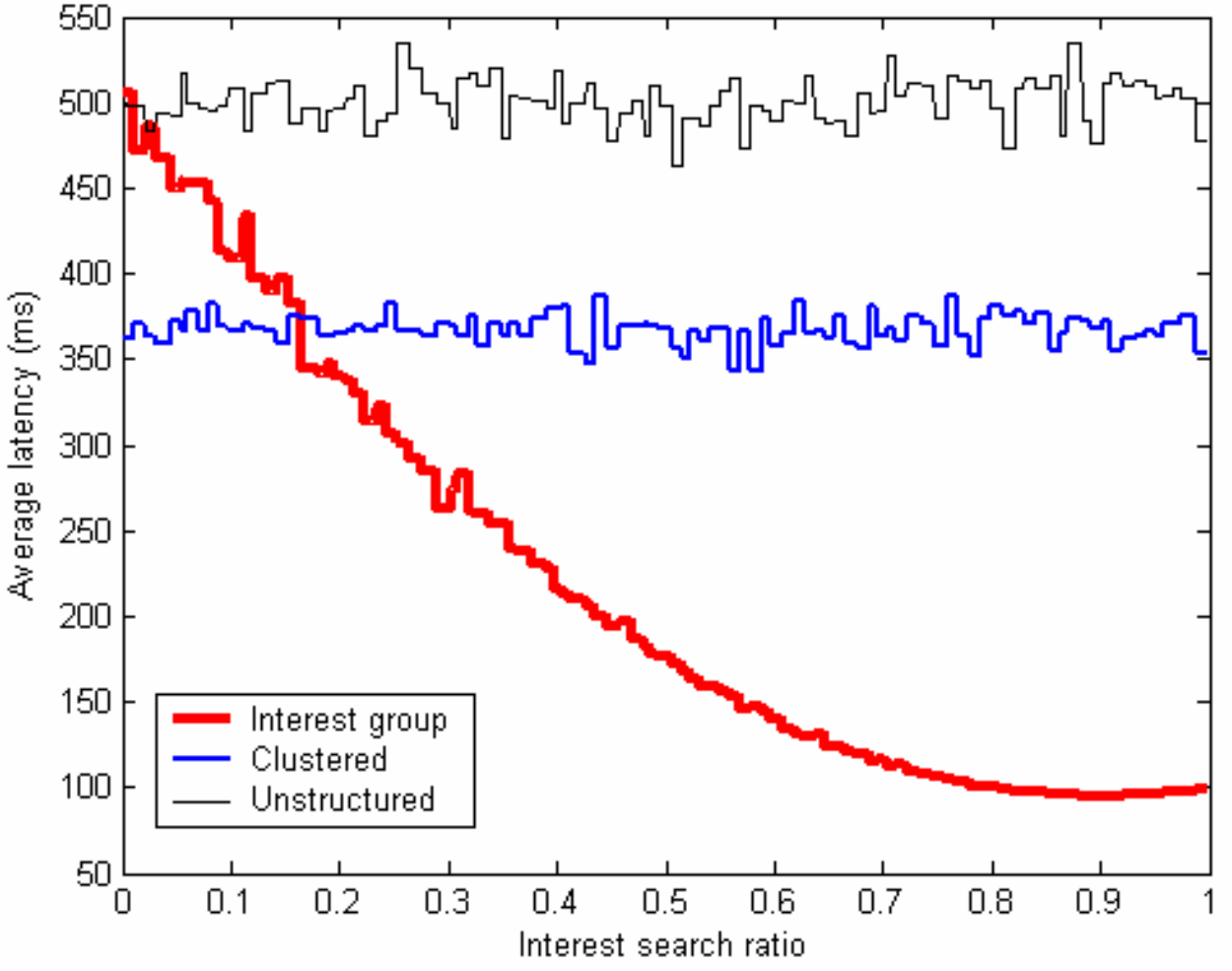
  M _group: Messages sent in groups the peer belongs to

  M_all: All the messages sent

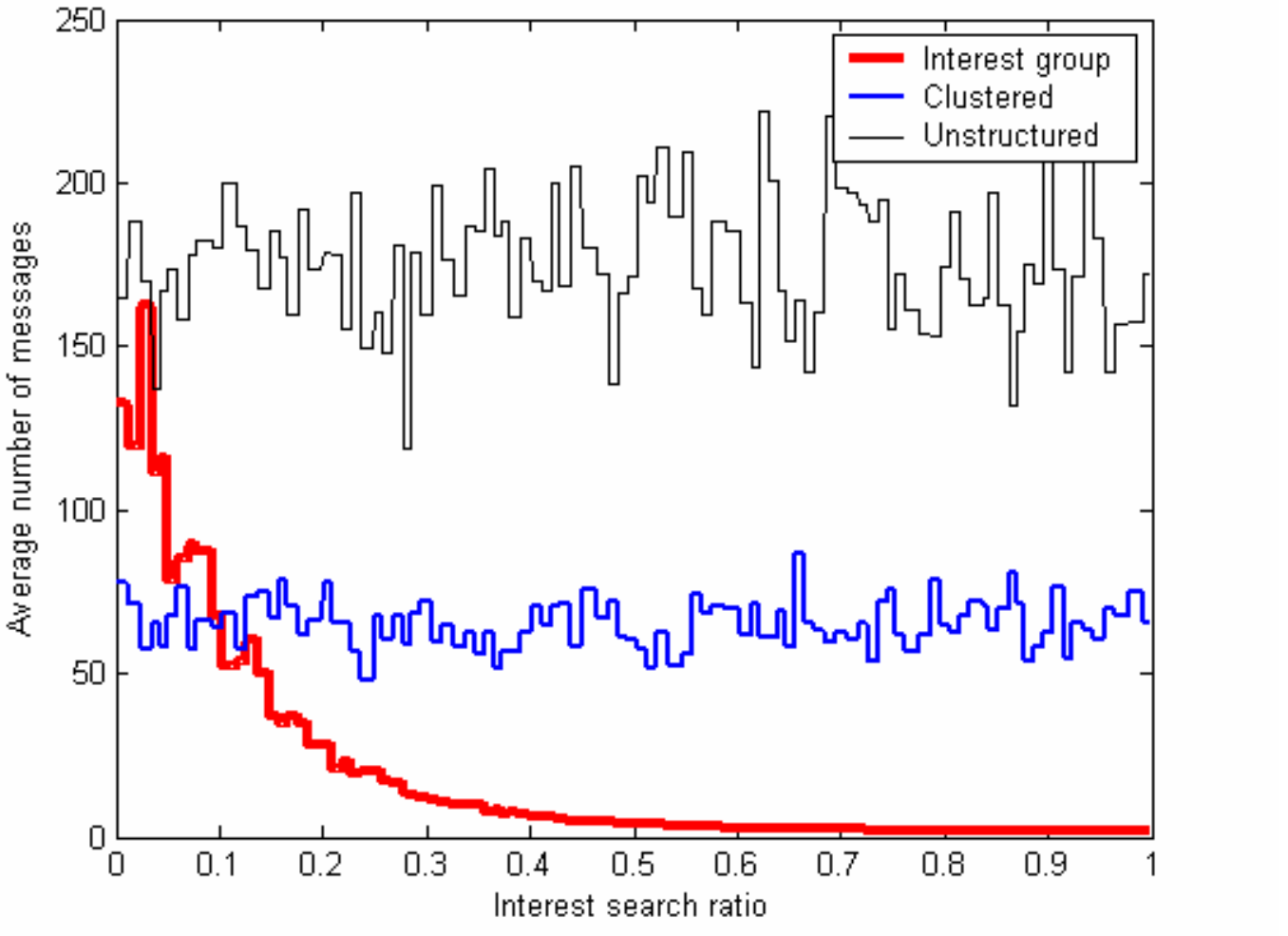
Simulation

parameters	Unstructured	Clustered	Interest group
System size	2000		
Group size	none	50	
Neighbors	3 - 6		
Latency	20 - 50 ms		30 - 60 ms

Simulation


550
500
450
400
350
300
250
200
150
100
50
Average latency (ms)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Interest search ratio
Interest group
Clustered
Unstructured

Simulation


Interest group
Clustered
Unstructured
Average number of messages
Interest search ratio
250
200
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Conclusion

- Considering the real circumstance in the peer-to-peer network system, most users search files what they are interest in at the peer-to-peer network system.
- The average latency and average number of messages are reduced in our proposed architecture while the interest search ratio is higher.

Reference

- [1] Napster Inc. The Napster homepage. In http://www.napster.com/, 2001
- [2] Open Source Community. Gnutella. In http://gnutella.wego.com/, 2001
- [3] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications.” In Proceedings of SIGCOMM’2001, 2001
- [4] Hu. T.H.-t, Sereviratne A. General clusters in peer-to-peer networks. In Networks, 2003. ICON2003. The 11th IEEE International Conference on 28 Sept.-1 Oct. 2003 Page(s):277 - 282.
- [5] Crespo A., Garcia-Molina H., “Routing Indices for peer-to-peer systems.” Distributed Computing Systems 2002
- [6] Portmann M., Sookavatana P., Ardone S., Seneviratne A., “The cost of peer discovery and searching in the Gnutella peer-to-peer file sharing protocol.” Networks, 2001. Proceedings. Ninth IEEE International Conference on , 10-12 Oct. 2001
- [7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In proceedings of ACM SIGCOMM, August 2001.
- [8] A. Rowstron and P.Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-peer Systems. In Proceedings of ACM/IFIP/USENIX Middleware, November 2001.

Reference

- [9] B. Y. Zhao, J. D. Kubiatowicz, and A.D. Joseph. Tapestry: An Infrastructure for Fault-Resilient wide-Area Location and Routing. Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001.
- [10] P. Reynolds and A. Vahdat. Efficient Peer-to-Peer Keyword Searching. In Proceedings of ACM/IFIP/USENIX Middleware, June 2003.
- [11] E. Cohen and S. Shenker. Replication Strategies in Unstructured Peer-to-Peer Networks. In Proceedings of ACM SIGCOMM, August 2002.
- [12] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and Replication in Unstructured Peer-to-Peer Networks. In proceedings of ACM ICS, June 2002.
- [13] E. Cohen, A. Fiat, and H. Kaplan. Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics. In Proceedings of IEEE INFOCOM, April 2003.
- [14] J. Yang, Y. Zhong, S. Zhang. An Efficient Interest-Group Based Search Mechanism in Unstructured Peer-to-Peer Networks. In Proceedings of the 2003 International Conference on Conputer Networks and Mobile Computing.
- [15] M. Hefeeda, A. Habib. B. Botev, D. Xu, B. Bhargava, “PROMISE: Peer-to-Peer Media Streaming Using CollectCast”, MM’03, November 2-8, 2003, Berkeley, California, USA.
- [16] Kobayashi, H.; Takizawa, H.; Inaba, T.; Takizawa, Y.; A self-organizing overlay network to exploit the locality of interests for effective resource discovery in P2P systems. In Applications and the Internet, 2005. Proceedings. The 2005 Symposium on 31 Jan.-4 Feb. 2005 Page(s):246 – 255
- [17] Rongmei Zhang and Y. Charlie Hu. Assisted Peer-to-Peer Search with Partial Indexing. In Proceedings of IEEE INFOCOM 2005, Miami, FL, March 13-17, 2005.