FAN: A Scalable Flabellate P2P Overlay Supporting Multi-Dimensional Attributes

Wei Song, Ruixuan Li, Zhengding Lu, Guangcan Yu
College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, P. R. China
E-mail: weisong@smail.hust.edu.cn, {rxli, zdlu}@hust.edu.cn; gcyu@smail.hust.edu.cn

Abstract

Peer-to-peer (P2P) technology provides an efficient way for resource distribution, and sharing. While most current P2P systems only support queries over a single attribute which limits the popularity of the P2P technology. Full-blown P2P applications require the efficient resource searching supporting multi-dimensional attributes. In this paper, we propose Flabellate overlAy Network (FAN), a scalable P2P overlay supporting multi-dimensional attributes. In FAN, the peers are mapped into a d-dimensional Cartesian space. The resource management and searching are based on the peer's second moment to the origin of coordinates. The theoretical analyses and experimental results demonstrate that FAN has high routing efficiency and low network maintenance cost over the existing structured P2P systems by storing logarithmic routing messages in peers and achieving logarithmic-hop resource searching. And many improved routing algorithms supporting multi-attribute queries can be implemented over FAN and achieve better performance.

1. Introduction

In the past several years, P2P technology has received considerable attention from the research and industry communities. However, most current P2P systems only support content searching over a single attribute (keyword or file name), such as file sharing applications popularized in Gnutella, BitTorrent. Full-blown P2P applications require an efficient resource searching over multi-attributes. For example, we develop an equipment trading system which requires the efficient equipment searching over manufacturer, production date, specifications, and so on. Therefore, an efficient paradigm for P2P resource routing over multi-dimensional resource attributes is needed.

There are many proposals trying to address the above issues, such as [1], [2], [3]. In the existing researches, CAN is a typical structured P2P overlay supporting multi-dimensional attributes, but searching in CAN is not very efficient and its maintenance cost is considerably high [3]. Many routing algorithms [2], [4] based on CAN have been proposed. However, these routing algorithms are proposed to supply complex query (e.g., range query, KNN query), and they can not improve the inherent weakness of CAN. Consequently, there is a need of methodologies and techniques for a scalable P2P overlay supporting efficient searching over multi-dimensional resource attributes.

The ultimate goal of our research is to develop a scalable P2P overlay supporting multi-dimensional resource attributes with high routing efficiency and low maintenance cost. Towards this end, we propose the Flabellate overlAy Network (FAN), a new structured P2P network supporting multi-dimensional attributes. In FAN, the peers are mapped into a d-dimensional Cartesian space. The resource searching and network management in FAN are based on the peer's distance. FAN peers can find resources in $O(\log(N/k))$ hops when there are N peers in FAN and a subspace contains up to k peers. We have presented the FAN construction, the network maintenance, and the corresponding peer management mechanism in this paper.

We carried out the simulation experiments to evaluate the FAN performance. The experimental results demonstrate that FAN can get a logarithmic routing efficiency with a low maintenance cost. Since FAN is a scalable routing overlay, many improved routing algorithms [1] supporting complex query based on CAN and the P2P routing algorithms [5] independent of any underlying framework can also be implemented over FAN.

The rest of the paper is organized as follows. In the next section, we compare FAN to some related work. Section 3 details the FAN protocols for resource

1550-445X/08 $25.00 © 2008 IEEE
DOI 10.1109/AINA.2008.66

searching and network management. In Section 4, we evaluate the FAN performance with the experiments. Finally, the paper is concluded in Section 5.

2. Related work

Researchers have done much work in P2P resource searching, which can be classified by many points of view such as network structure, query dimension, etc. In this section, we discussed unstructured and structured P2P networks, and introduced their respective routing algorithms.

The first category is the routing algorithms in the unstructured P2P network, such as Gnutella [6]. A peer in Gnutella uses flooding to query its neighbors within a radius. Such searching approach does not depend on any peer, but it leads to unacceptable network load.

The second category is the routing algorithms in the structured P2P network. We divided the routing algorithms into two subcategories: 1) algorithm supporting single dimensional attribute; 2) algorithm supporting multi-dimensional attributes.

Chord [7] and Tapestry [8] are two typical structured P2P network supporting resource search over single dimensional attribute. Chord assigns keys to nodes with consistent hashing and adopts a ring topology to manage resources and its routing efficiency is $O(\log N)$. Tapestry is similar to the longest prefix matching technology in the classless inter-domain routing (CIDR). And its routing efficiency is also $O(\log N)$.

CAN uses a d-dimensional Cartesian space to manage the resources and support resource search over multi-dimensional attributes. Many improved routing algorithms [1], [2], [4] over CAN have been proposed. Each CAN peer has a set of d-dimensional coordinates. Every node manages a virtual zone containing itself and stores information of its immediate neighbors. CAN routing efficiency is $O(dN^{1/d})$. However, both the normal leaving procedure and the immediate takeover will make a node hold several zones. When peers frequently join and leave CAN or simultaneously appear failure with multiple adjacent peers, the maintenance cost of CAN is considerably high.

In the past several years, many related work has been done on P2P resource searching supporting multi-dimensional attributes. For example, Bin Liu et al. proposed a routing algorithm [1] supporting multi-dimensional queries, A. R. Bharambe et al. proposed Mercury [9], C. Tang et al. proposed pSearch [2], and H. V. Jagadish proposed a multi-dimensional indexing P2P schema VBI-Tree [10]. Moreover, some other P2P routing algorithms such as adaptive connection establishment [11], P-Ring [12], Gossip query [13], SCOPE [14], clustering search [15] and assisted P2P search [16] have been proposed. While the motivations of these works were to enable the complex query over multi-dimensional attributes, they were not to improve the resource searching over multi-attributes. Hence, the P2P applications need an efficient P2P underlying routing framework supporting multi-dimensional attributes. And it is also the motivation of our research. Since the proposed overlay is fan-shaped, we call the network architecture in this paper as the Flabellate overlAy Network (FAN), and the corresponding routing method is named FAN routing algorithm. Many improved routing algorithms [1], [4] over CAN and other algorithms [5] independent of underlying DHT topology can also be implemented in FAN and achieve better performance.

3. FAN protocols

3.1. FAN construction

The resources are described by d-dimensional attributes. And FAN uses consistent hashing [18] to compute the attributes and map the resources into a node of a d-dimensional Cartesian space which is called FAN mapping space. In a d-dimensional FAN, given a node P at $(x_1,x_2,...,x_d)$, we define its second-moment to the origin of coordinates which is shown in Equation 1 as its distance in FAN.

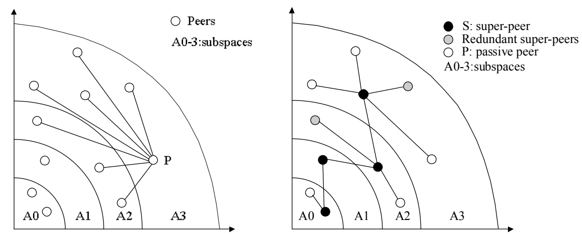
$$D_p = x_1^2 + x_2^2 + ... + x_d^2 = \sum_{i=1}^{d} x_i^2 . \tag{1}$$

FAN mapping space is divided into non-overlapping and continuous subspaces, and each node falls into a unique subspace. The FAN subspace is defined as follows. We use $A(a,b)$ to denote a subspace which covers peer's distance range $(a,b]$. For each node P in the subspace $A(a,b)$, its distance D_p satisfies $a<D_p \bullet b$. Furthermore, all nodes P whose distance satisfies $a<D_p \bullet b$ belong to the subspace $A(a,b)$. And we also define the distance between the subspace $A(a,b)$ and a peer P in Equation 2.

$$Distance_{AtoP} = \begin{cases} a - D_P & a > D_P \\ 0 & a \le D_P \le b \\ D_P - b & D_P > b \end{cases} . \tag{2}$$

We draw a 2-dimensional FAN structure in Figure 1(a). Resource searching in FAN is equal to finding its subspace. Therefore, a peer in FAN must store peer information in the same and adjacent subspaces, like the peer P in Figure 1(a). This structure makes peers store much routing information. To reduce the routing messages at every peer, we use super-peers to manage the subspaces. Figure 1(b) illustrates a 2-dimensional FAN structure with super peers. Other dimensions are similar.

In FAN, super-peers manage the subspaces, process searching requests. The super-peer plays more important roles than the passive peers. To achieve good performance and high stability, a super-peer should have more computational power than the passive peers. With more peers joining the subspaces, it is possible that some passive peers have more computational power than the super-peers. In FAN, we assume that every peer is altruistic, the super-peer periodically checks all peers in the subspace to look for any passive peer with more computational power and better network bandwidth. If found, the powerful passive peer will replace the super-peer to manage the subspace. According to the related analysis [17], the redundant super-peer is necessary for a realistic P2P system. So in FAN, a redundant super-peer watches the supper-peer and synchronizes the super-peer information. When the super-peer leaves, the redundant one will replace it.



(a) FAN without super-peer (b) FAN with super-peers

Figure 1. 2-dimensional FAN network structure

3.2. Routing in FAN

Every peer in FAN belongs to a subspace and maintains a routing table which stores peer coordinates, IP address, and the subspace information. The super-peer stores the information of all peers in its subspace, i.e. item 1 to 3 in Table 1(a) and the super-peers in its adjacent subspace, i.e. item 4 and 5. The passive peer only stores the super-peer and the back-up super peer in the subspace.

Table 1. FAN routing table in a super-peer

Peer coordinates	Peer address	Subspace range
(1,0.5,1,1.5, 1, 1)	211.69.192.70:9705	(5,10)
(0,0.5,2,1.5, 1)	211.69.192.80:9705	(5,10)
(1,1,1,1.4,0, 1.5)	211.69.202.18:9706	(5,10)
(1,0,0.5,0.5, 1)	211.80.102.79:9705	(3,5)
(2,2.5,1.5,0, 2)	211.82.101.78:9705	(10,18)

The peers in FAN store all resource information in the same subspace. Therefore, searching resources in FAN is equivalent to finding a peer who is in the same subspace with the resources. When a peer *P* searches the resources *R*, *P* firstly computes *R*'s coordinates and searches *R* locally. If not found, *P* delivers the query to its super-peer *S* (if *P* is a super-peer, this step can be omitted). When *S* receives the query request, *S* also searches *R* locally. If not found, *S* delivers the query message to the super-peer of an adjacent subspace closer to *R*. Thus, FAN routing strategy always delivers the query message to the super-peer closer to the target until it arrives.

Algorithm 1. FAN resource searching algorithm

Input: *P*, query point; *R*, query target resources.

```
Routing (P, R)
if (R is in P's resource list) then
    return R;
else if (P is a passive peer) then
    deliver the searching request to its super-peer;
else
    deliver the request to the nearest super-peer;
end if end if
```

For the sake of generality, we assume that the FAN mapping space is *d*-dimensional, network size is *N* and a subspace can contain *k* peers at most.

FAN subspace amount is $O(N/k)$, and using the routing process in Algorithm 1 the routing messages can only be relayed from one subspace to its immediate neighbor. Therefore, the routing efficiency is also $O(N/k)$ which is not satisfied. The key to optimization of FAN is the selection of long-distance links that are maintained in addition to the immediate adjacent subspaces. We introduce the extended adjacent subspace to improve the FAN routing efficiency.

In an improved FAN overlay, a super-peer stores not only the immediate adjacent super-peers but also the super-peers in the subspaces at intervals of 2^j layers which are defined as the extended adjacent subspace. Figure 2 illustrates the improved FAN overlay structure.

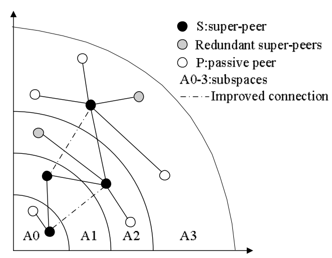


Figure 2. The improved FAN structure

With the improvement, super-peers can deliver the query messages to the super-peer whose subspace is nearest to the target resources. Suppose that a routing source peer and the target resources are at the intervals of *M* layer subspaces and every subspace covers an equal distance range. And then the distance between the source peer and the target resources can be halved during each query message transferring. Moreover, *M* follows the uniform distribution from 1 to $O(N/k)$, so a query message can reach the target in $O(\log(N/k))$ hops.

Using the extended adjacent subspaces, FAN routing efficiency can be improved from $O(N/k)$ to $O(\log(N/k))$. However, the additional maintenance cost can not be ignored. The extended adjacent subspace in FAN has an important property, the transitivity property.

Property 1. Transitivity. If subspace B_2 is an extended adjacent subspace of subspace B_1 at interval of $2^j(j>1)$ layers, then there must exist a subspace B between B_1 and B_2, and B is both B_1's and B_2's extended adjacent subspace. We call this property as the transitivity of the extended adjacent subspace.

For the transitivity property of the extended adjacent subspace, a super-peer only needs know the immediate adjacent subspace information and periodically explores the known extended adjacent subspaces. By this way, it can update all the extended subspace information. Since the super-peer stores more routing information ($O(\log(N/k))$) which enables it quickly deliver query requests to improve FAN routing efficiency. Moreover it can reduce the load of the passive peers.

3.3. Peer joining

When a peer P attempts to join FAN, P firstly connects to a peer as bootstrap, and finds the subspace A which covers P's distance and join. If the peers in A does not reach k, P registers at A's super-peer to join the subspace. Otherwise, because FAN routing efficiency is relative to the subspace amount, we adjust A with its immediate adjacent subspaces rather than split A immediately. In such situation, P firstly registers at A's super-peer S, which checks whether peers in two immediate adjacent subspaces A_1 and A_2 have reached k. If peers in both A_1 and A_2 already have reached k, and then A splits into two new subspaces with equal peer amount. Otherwise, A adjusts with the immediate adjacent subspace which has the fewer peers.

Algorithm 2. Peer joining FAN

Input: P, the peer attempting to join the FAN.

PeerJoin(P)

1. connect to a peer and find subspace A covers P;
2. if (peer amount in $A < k$) then
3. P registers at super-peer to complete joining;
4. else if (peer in the immediate adjacent subspaces = k)
5. A split into two subspaces, P joins one;
6. else
7. A adjusts with an immediate adjacent subspace;
8. end if end if

In FAN we use the subspace adjustment to decrease the subspace amount. However, the super-peer may be transferred during the adjustment operation. We call the transferred super-peer as the outdated super-peer. The message generated during the joining process is computed in three cases:

a) When subspace peer is less than k, the new joining peer registers at the super-peer to complete joining. The joining process generates $O(1)$ messages.

b) We assume that the peers in a subspace follow the uniform distribution from 1 to k. When the joining process requires subspace adjustment but no super-peer transferring, some passive peers will change subspaces. The mathematical expectation of changing subspace peer is shown in Equation 3. Meanwhile, two adjusted subspaces should transmit new subspace information to their $O(\log(N/k))$ extended adjacent subspaces. It generates $O(k/4+2\log(N/k))$ messages in total.

$$E=\sum_{i=1}^{k-1}\left(k+1-\frac{k+1+i}{2}\right)/(k-1)=\frac{k}{4}-\frac{1}{2}\ . \qquad (3)$$

c) If super-peer is transferring during subspace adjusting, $O(k/4)$ peers will change subspaces. And two super-peers need issue the new subspace information to $O(\log(N/k))$ extended adjacent subspaces. It generates $O(k/4+2\log(N/k))$ messages in total.

When the super-peer transferred in peer joining process, before the new subspace information reaches all extended adjacent subspaces, the routing requests will still be delivered to it. So the outdated super-peer does not discard the routing information immediately. We start a timer initialized in proportion to the volume of the outdated super-peer routing table. When the timer expires, the outdated super-peer considers that all the extended adjacent subspaces have gotten the new subspace information and discards the outdated routing information. Analyzing case a, b, and c, we can draw a conclusion that the cost for a peer joining FAN is $O(\frac{\log(N/k)}{k})$. The expected message amount in a joining process is expressed in Equation 4.

$$E(\cos t)=\frac{(k-1)\times 1+k/4+2\log(N/k)}{k}\approx\frac{2}{k}\log(\frac{N}{k})\ . \quad (4)$$

3.4. Peer departing

When a peer attempts to leave FAN, the associated routing information must update on time to ensure FAN integrity. When a passive peer attempts leaving FAN, it notifies its super-peer to complete leaving. In this case, it generates $O(1)$ message. When a super-peer attempts to leave, the backup super-peer replaces it. The message generated in this process is also $O(1)$. However, if the leaving peer is the last one, the immediate adjacent subspace B with fewer peers takes over this empty subspace and $O(\log(N/k))$ messages would be generated. Therefore, it is found that the peer's department brings $O(\frac{\log(N/k)}{k})$ massages which are expressed in Equation 5.

$$E(\cos t) = \frac{(k-1)\times 1 + 2\log(N/k)}{k} \approx 1 + \frac{2}{k}\log(\frac{N}{k}) . \quad (5)$$

Algorithm 3. Peer departing from FAN
Input: *P*, peer attempting to leave FAN
PeerLeave (*P*)
1. if (*P* is a passive peer) then
2. super-peer deletes leaving peer information;
3. else if (*P* is the last one in the subspace) then
4. an immediate adjacent subspace takes over;
5. else
6. backup super-peer replaces the leaving one;
7. end if end if

The simultaneous departure of the super-peer and backup one in a subspace will distort the extended subspace links. To ensure the stability of FAN, each passive peer maintains information of immediate adjacent subspaces and the passive peers in the same subspace. When a passive peer detects that the super-peer and backup one are both offline in periodical exploration, the passive peer notifies other peers in the same subspace to choose one as the new super-peer. The new super-peer connects the immediate adjacent subspaces to reconstruct the extended subspace links.

4. Performance evaluation

4.1. Simulation setup

This section presents the detailed evaluation of FAN protocol using simulations. We use PeerSim [19], a P2P simulation framework for testing P2P protocols, to implement FAN. In the FAN experiments, a peer is described by *d*-dimensional attributes. Every dimensional attribute is an integer followed the uniform distribution from 0 to *M*.

The proportion of peer joining and leaving operations kept roughly equal. Each peer averagely issues 100 queries during online. Additional parameters in the simulations are shown in Table 2. In this section, we implemented Chord and FAN protocols over PeerSim, while the data of CAN come from [3].

Table 2. Additional experimental parameters

	Parameter Descriptions	Values
N	The total number of peers	256-64K
d	mapping space dimensions	2, 3, 5, 7
k	The capability of a subspace	4, 8, 12, 16, 24
M	The range of each dimension	500, 1000, 2000

4.2. Maximal number of peers with the same distance

The peer distance is the foundation of subspace partitioning and resource searching. However, in a *d*-dimensional Cartesian space, some different peers had different coordinates may have the same peer distance. Furthermore, the peers with the same distance cannot be placed into different subspaces. So, the max number of peers with the same distance can not exceed the subspace capability (*k*). We carried out the experiments to evaluate it with various *N*, *d*, and *M* values.

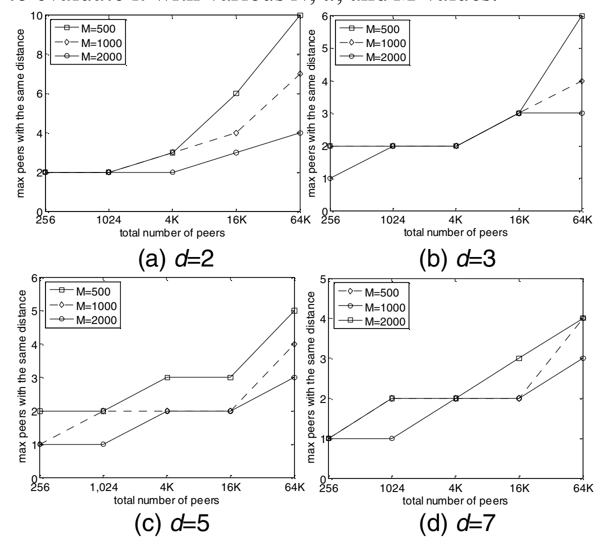


Figure 3. Peers with the same distance in FAN

The numerical results in Figure 3 show that, in most situations, the maximal number of peers with the same distance is smaller than 8, i.e. it can be well supported by a subspace. The only exception is that when *d*=2 and the total number of peers is very large (more than 64K), we should choose an appropriate *M* value (bigger than 1000). And we can conclude that the FAN subspace division strategy will not be invalidated due to the maximal number of peers with the same distance exceeding the subspace capability.

4.3. Average amount of the subspaces in FAN

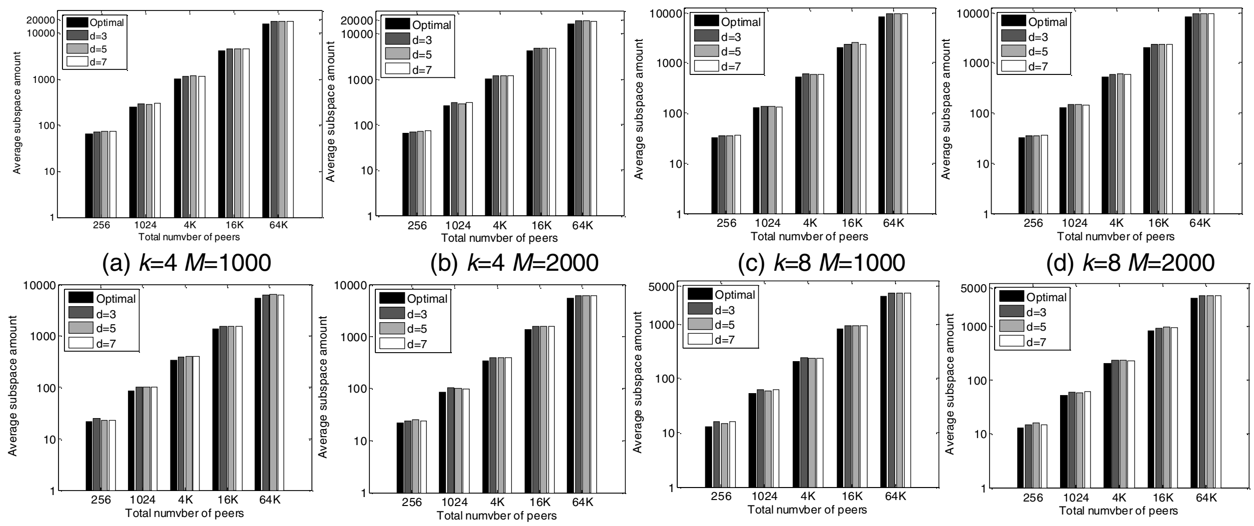
As discussed in Section 3, FAN routing efficiency and the message amount of peer joining and leaving FAN, both are related to the subspace amount. And all our analyses are based on the assumption that the subspace amount is $O(N/k)$. Therefore, whether the FAN subspace management strategy can efficiently slow the subspace increasing is crucial for the scalability and availability of FAN.

We develop the simulations to measure the statistics of average subspace amount with various *N*, *d*, *k*, and *M* values. The numerical results in Figure 4 show that the average subspace amount is approximately equal to *N*/*k*. Through this simulation experiments, we have found that FAN subspace management is efficient and can keep that the subspace amount is $O(N/k)$. Moreover, the analysis of FAN routing efficiency and maintenance cost based on the assumption that the subspace amount is $O(N/k)$ is reasonable.

4.4. Routing efficiency of single-dimensional FAN

FAN uses the extended adjacent subspace links similar to Chord to improve its routing efficiency. FAN is a routing algorithm proposed to support multi-dimensional attributes, however it can also support single-dimensional attribute. Moreover, the searching in the single-dimensional FAN can also represent the efficiency of the routing message spreading in the extended adjacent subspaces. So we carried out a simulation experiment to compare routing efficiency of single-dimensional FAN and Chord.

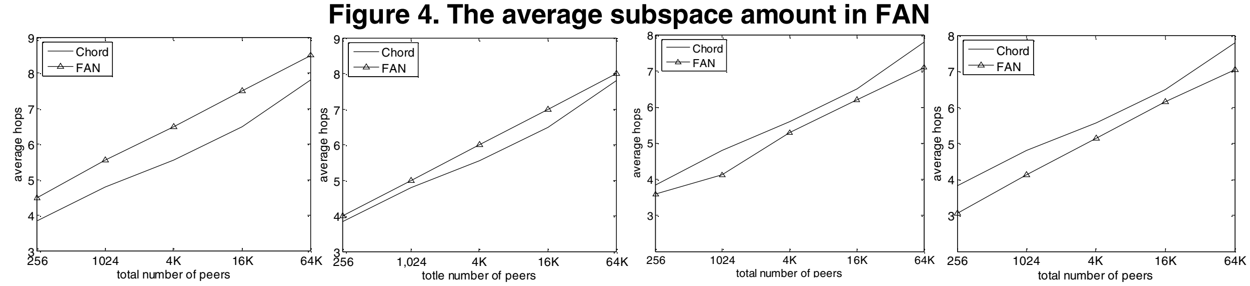
As the numerical results shown in Figure 5, the single-dimensional FAN also can achieve logarithmic routing efficiency like Chord does. Thus, we can say that routing messages can efficiently spread using the extended adjacent subspace links.



(a) k=4 M=1000 (b) k=4 M=2000 (c) k=8 M=1000 (d) k=8 M=2000

(e) k=12 M=1000 (f) k=12 M=2000 (g) k=20 M=1000 (h) k=20 M=2000

Figure 4. The average subspace amount in FAN



(a) k=1 (b) k=2 (c) k=4 (d) k=8

Figure 5. Routing efficiency of single-dimensional FAN and Chord

4.5. Routing efficiency of multi-dimensional FAN

As the main idea described above, FAN is an efficient P2P overlay supporting multi-dimensional attributes. In Section 3, we analyses the FAN routing efficiency in theory. In this section, we design simulation experiments to evaluate the average routing hops in multi-dimensional FAN with various N, k, and d values. Since CAN uses greedy forwarding to deliver the routing messages, multiple peers in a zone will be overloaded with the continuous increase of message amount. So we compare FAN and CAN routing efficiency with k=1, 2, 3, 4 as [3] suggests. Moreover, we carried out more experiments to investigate how N, k, and d values influence the FAN routing efficiency.

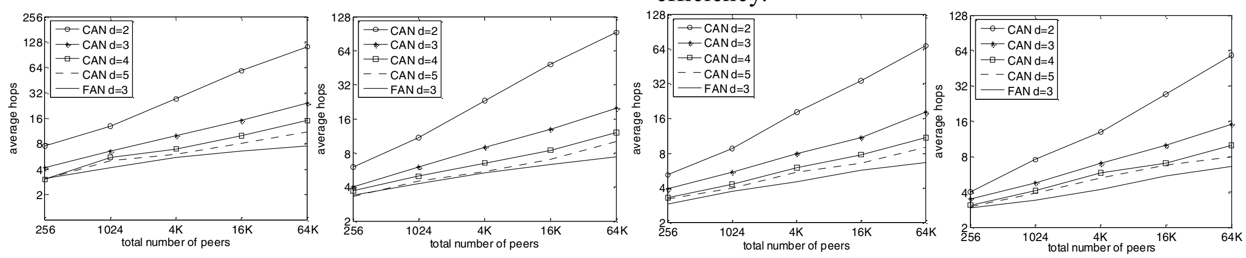
Firstly we compare routing efficiency of FAN and CAN with the same capability (k) in a subspace. In Section 3, we analyses the FAN routing efficiency is $O(\log(N/k))$ which has little relation with the dimension size. However, CAN routing efficiency has great relation with the dimension of resource attributes. So we compare 3-dimensional FAN with CAN having various d values. The graphs in Figure 6 show that

FAN can get better routing efficiency over CAN in the same circumstances.

We also carry out simulation experiments to find the relationship between FAN routing efficiency and k values. As the graphs in Figure 7 shown, the FAN routing efficiency gets better with bigger k value, and FAN can work well in the large network size (N=64K) as well. Furthermore, we can see from the numerical results that the average routing hops has an logarithmic relationship with N/k, which illustrates that our analyses of FAN routing efficiency is reasonable.

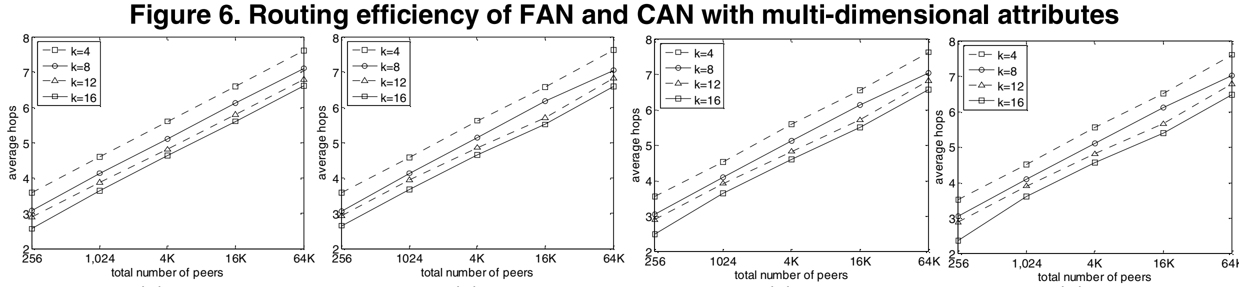
FAN is proposed to support multi-dimensional attributes, so we hope FAN can well support high attribute dimension. Then we carry out simulation experiments to evaluate FAN routing efficiency with various d values. As the graphs in Figure 8 shown, the FAN routing efficiency has little change with various d values, so we can draw a conclusion that FAN can work well in a large resource attribute dimensions.

Through the experiments in this subsection, we can find that FAN is an efficient overlay supporting multi-dimensional attributes with O(log(N/k)) routing efficiency.



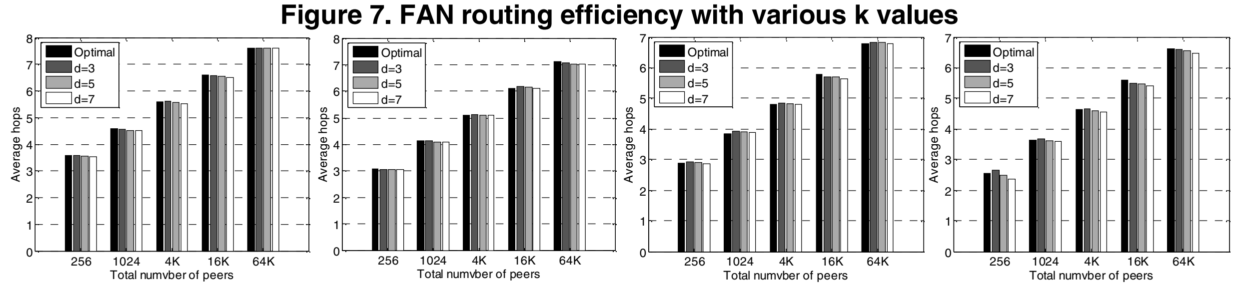
(a) k=1 (b) k=2 (c) k=3 (d) k=4

Figure 6. Routing efficiency of FAN and CAN with multi-dimensional attributes



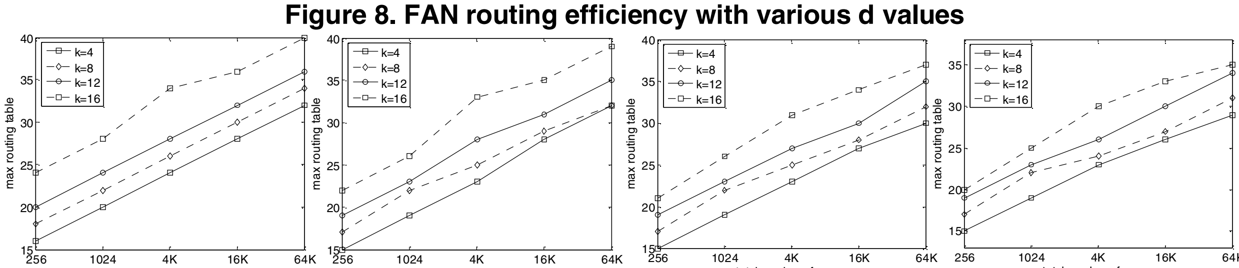
(a) d=2 (b) d=3 (c) d=5 (d) d=7

Figure 7. FAN routing efficiency with various k values



(a) k=4 (b) k=8 (c) k=12 (d) k=16

Figure 8. FAN routing efficiency with various d values



(a) d=2 (b) d=3 (c) d=5 (d) d=7

Figure 9. Maximal routing table messages of super-peers in FAN

4.6. Maximal routing table messages in FAN

The links of the extended adjacent subspaces achieve fast resource searching in FAN and also make super-peers store more routing information. So we have to take into account whether the routing messages will overburden the super-peers and influence FAN stability or not. We carried out an experiment to evaluate routing table data items of FAN super-peers with various N, k, and d values.

The graphs in Figure 9 show that the maximal amount of routing table items has logarithmic relationship with N. And the routing table size is acceptable (less than 50), even though the total number of peers in FAN is more than 64K. Hence, we can say that the routing table size will not overburden the super-peers in FAN.

5. Conclusion and future work

FAN is a P2P overlay supporting multi-dimensional attributes. Comparing with CAN, FAN has the advantages in routing efficiency and maintains cost. Current P2P applications require efficient resource searching over multi-dimensional attributes. Many algorithms over CAN are proposed to solve this problem. The improved algorithms such as [1], [4] and [5] can also be implemented in FAN and achieve better performance due to the advantages of FAN architecture. For future work, we plan to facilitate FAN optimization in the relation between cost and various network parameters in FAN. Moreover, the complex queries based on FAN may still need to be explored.

Acknowledgment

This research is partially supported by National Natural Science Foundation of China under Grant 60403027, and 60773191, National High Technology Research and Development Program of China under Grant 2007AA01Z403, Natural Science Foundation of Hubei Province under Grant 2005ABA258.

References

[1] B. Liu, W.-C. Lee, and D. L. Lee, "Supporting Complex Multi-Dimensional Queries in P2P Systems", In: *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005, pp. 155-164.

[2] C. Tang, Z. Xu, and M. Mahalingam, "pSearch: Information Retrieval in Structured Overlays", *ACM SIGCOMM Computer Communications Review*, 2003, 33(1), pp. 89-94.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A Scalable Content-Addressable Network", In: *Proc. of the ACM SIGCOMM 2001*. pp. 161-172.

[4] M. Demirbas, H. Ferhatosmanoglu, "Peer-to-Peer Spatial Queries in Sensor Networks", In: *Proc. of the 3rd IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2003, pp. 32-39.

[5] I. Bhattacharya, S. R. Kashyap, and S. Parthasarathy, "Similary Searching in Peer-to-Peer Databases", In: *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005, pp. 329-338.

[6] Gnutella. *http://www.gnutella.com*.

[7] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", In: *Proc. of the SIGCOMM 2001*. pp. 149-160.

[8] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", *IEEE Journal on Selected Areas in Communications*, 2004, 22(1): 41-52.

[9] A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting Scalable Multi-Attribute Range Queries", In: *Proc. of the ACM SIGCOMM Conference*, 2004, pp. 353-366.

[10] H. V. Jagadish, B. C. Ooi, Q. H. Vu, Z. Rong, and O. Zhou. "VBI-Tree: A Peer-to-Peer Framework for Supporting Multi-Dimensional Indexing Schemas", In: *Proc. of the 22nd International Conference on Data Engineering (ICDE)*, 2006.

[11] Y. Liu, Z. Zhuang, L. Xiao, and L. Li, "A Distributed Approach to Solving Overlay Mismatching Problem", In: *Proc. of the 24th International Conference on Distributed Computing Systems (ICDCS 2004),* 2004, pp.132-139.

[12] A. Crainiceanu, P. Linga, and A. Machanavajjhala, "P-Ring: An Efficient and Robust P2P Range Index Structure", In: *Proc. of the SIGMOD* 2007. pp. 223-234.

[13] E. Simonton, B. Choi, and S. Seidel, "Using Gossip for Dynamic Resource Discovery", In: *Proc. of the 35th International Conference on Parallel Processing (ICPP),* 2006. pp. 319-326.

[14] X. Chen, S. Ren, H. Wang, and X. Zhang, "SCOPE: Scalable Consistency Maintenance in Structured P2P System", In: *Proc. of the 24th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2005. pp. 1502-1513.

[15] L. Ramaswamy, B. Gedik, and L. Liu, "A Distributed Approach to Node Clustering in Decentralized Peer-to-Peer Networks", *IEEE Transactions on Parallel and Distributed Systems*, 2005, 16(9), pp. 814-829.

[16] R. Zhang, and Y. C. Hu, "Assisted Peer-to-Peer Search with Partial Indexing", In: *Proc. of the 24th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2005. pp. 1514-1525.

[17] D. R. Karger, E. Lehman, F. Leighton, M. Lewin, and R. Panigrahy, "Consistent hashing and random trees: Distributed cashing protocols for relieving hot spots on the World Wide Web", In: *Proc. of the 29th Annual ACM Symposium Theory of Computing*, 1997. pp. 654-663.

[18] B. Yang, and H. Garcia-Molina, "Design a Super-Peer Network", In: *Proc. of the 19th International Conference on Data Engineering (ICDE)*, 2003. pp. 49-60.

[19] PeerSim. http://sourceforge.net/projects/PeerSim.