Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Hybrid ants-like search algorithms for P2P media streaming distribution in ad hoc networks*

ZUO Dong-hong†, DU Xu†‡, YANG Zong-kai

(*Department of Electronics and Information Engineering, Huazhong Univerisity of Science and Technology, Wuhan 430074, China*)

†E-mail: sixizuo@mail.hust.edu.cn; duxu@mail.hust.edu.cn

Received Oct. 16, 2006; revision accepted Feb. 5, 2007

Abstract: Media streaming delivery in wireless ad hoc networks is challenging due to the stringent resource restrictions, potential high loss rate and the decentralized architecture. To support long and high-quality streams, one viable approach is that a media stream is partitioned into segments, and then the segments are replicated in a network and served in a peer-to-peer (P2P) fashion. However, the searching strategy for segments is one key problem with the approach. This paper proposes a hybrid ants-like search algorithm (HASA) for P2P media streaming distribution in ad hoc networks. It takes the advantages of random walks and ants-like algorithms for searching in unstructured P2P networks, such as low transmitting latency, less jitter times, and low unnecessary traffic. We quantify the performance of our scheme in terms of response time, jitter times, and network messages for media streaming distribution. Simulation results showed that it can effectively improve the search efficiency for P2P media streaming distribution in ad hoc networks.

Key words: Ad hoc networks, Media streaming distribution, Search algorithms, Peer to peer (P2P)
doi:10.1631/jzus.2007.A1191 **Document code:** A **CLC number:** TP393.09

INTRODUCTION

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without a centralized administration. The mobile hosts or pervasive end-user devices are small and light enough to be carried around. Due to the limited range of their transmissions, a multi-hop network must be constructed and the hosts need cooperating to perform communication tasks. Media streaming in such networks is attractive and is also challenging due to stringent resource restrictions at the mobile hosts, dynamic network connectivity, and potentially high loss rate (Jin, 2004). To support media streaming applications with limited resources, one approach is divide-and-conquer (Jin, 2004; Ghandeharizadeh *et al*., 2004; Xue *et al*., 2004): partitioning a media stream into segments and managing them in a peer-to-peer (P2P) fashion. If any end-user wants to display it, the device first sends queries to its neighbors to ask for content. Then from multiple sources, the device receives the segments and continuously plays out the entire clip. The primary advantage of this approach is that it eliminates the requirement of large memory space in the devices. This is important if we want to support long and high-quality media streams. Additional advantages include more balanced load among mobile hosts. Unfortunately, this approach brings us challenges on searching for multiple segments and reassembling them in such unstructured P2P networks. Flooding-based search algorithms are adopted by most of them for reliability and low latency. However, they incur large volume of unnecessary traffic in networks, and in wireless ad hoc networks they waste a large amount of energy for wireless devices to propagate the excessive traffic overheads. This paper will focus

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 60302004) and the Natural Science Foundation of Hubei Province, China (No. 2005ABA264)

on the search algorithms for unstructured P2P media streaming distribution in ad hoc networks.

There are some particular characters for media streaming distribution in unstructured P2P networks. One is that of the inherent dynamics of P2P networks: changes in the content repositories occur frequently, and peers can join or leave at any time (churn). Another is that in media distribution P2P networks all peers query, cache and play the media segments in sequence. If peer A has a former segment that peer B is interested in, then it is likely that A will have the latter segment that B is also interested in, which means the searched contents in such unstructured P2P networks is correlated. This paper aims to deliver an algorithm that includes strategies to cope with these dynamics and content correlation.

We propose a hybrid ants-like search algorithm (HASA) for P2P media streaming distribution in ad hoc networks. It gathers the advantages of random walks and ants algorithms for searching in unstructured media streaming distribution P2P networks, such as: (1) low latency, (2) less jitter times, and (3) low unnecessary traffic by using the ant colony theorem (Bonabeau *et al*., 2000). As a real ant moves, it deposits a substance called pheromone on the ground, which is evaporated over time, to collect and to maintain the statistics information.

This paper is organized as follows. In Section 2, we review some related works on search algorithms for unstructured P2P networks and debate their fitness for unstructured media streaming distribution P2P networks. Section 3 describes the HASA algorithm in detail and Section 4 evaluates the performance of HASA in experiment. The further research direction is discussed in Section 5.

RELATED WORKS

In an unstructured P2P system, such as Gnutella and KaZaA, file is placed randomly, which has no correlation with the network topology. The most popular query operation in use, such as Gnutella and KaZaA (among super-nodes), is to blindly "flood" a query to the network. A query broadcasts and rebroadcasts until a certain criterion is satisfied. Blind flooding has the following merits: (1) modest latency (or response time), (2) large coverage, and (3) high reliability (Jiang *et al*., 2003). However, in wireless ad hoc networks, the wireless devices' power is limited while the excessive traffic overheads caused by a large number of redundant message forwarding consume significant part of the network's energy and bandwidth. The inefficient blind flooding search technique causes the wireless unstructured P2P systems being far from practicable for long media streaming distribution.

To avoid the large volume of unnecessary traffic incurred by flooding-based search, many efforts have been made to improve search algorithms for unstructured P2P systems (Adamic *et al*., 2001; Lü *et al*., 2002; Yang and Garcia-Molina, 2002; Jiang and Jin, 2005; Michlmayr, 2006). One approach is K random walks (Lü *et al*., 2002), in which instead of flooding to all immediate overlay neighbors, a peer randomly selects a subset of its neighbors to query. Instead of randomly selecting relay neighbors, some mechanisms have been proposed to select relay neighbors based on some statistics information of some metrics, such as the number of neighbors, the number of results received from neighbors from previous queries or the latency of the connection with that neighbor. A peer selects a subset of the neighbors to send its query based on some heuristics, such as selecting the neighbors that have the largest number of neighbors (Adamic *et al*., 2001) or selecting the neighbors that have returned the largest number of results from previous queries or selecting the neighbors that have smaller latency (Yang and Garcia-Molina, 2002). When handling a query message (either relayed from its neighbor or originated from itself) in a statistics-based search algorithm, the peer determines the subset of its logical neighbors to relay the query message. These search mechanisms may significantly reduce the traffic volume but may also reduce the query coverage range so that a query may traverse a longer path to be satisfied or not be satisfied.

In order to effectively reduce the traffic incurred by flooding-based search and alleviate the partial coverage problem, the hybrid search algorithms (Zhuang *et al*., 2003; Gkantsidis *et al*., 2005) are proposed for unstructured P2P systems, in which the number of relay neighbors can be changed based on a function, with the relay neighbors being selected based on multiple metrics and the search depths determined based on some algorithms. But all these

hybrid search algorithms do not study on how to collect the statistics information. They have no strategies to deal with unstructured P2P network churn, and are designed for the file sharing system. They do not consider the correlation of continuous queries for media streaming segments, so they are not adapted to media streaming distribution in P2P network well for reasons of strict media streaming start-up and consistent time constraints.

DESCRIPTION OF HASA

The application scenario for the algorithm is that of a distributed media segments search engine for one media streaming distribution where each peer (1) replicates some media segments in its repository which are played back recently and offers its content to other peers, and (2) performs searches based on media segment ID. The maximum number of pheromone types equals the media segments number, which represents the sorts of query message. At each peer P_i, pheromone trails which record the amount of pheromone laid by ants are maintained in a table of size $m \times n$, where m is the number of the media segments and n is the number of peer P_i's outgoing links to neighbor peers P_u (u=1, 2, ..., n). Each $\tau_{s,i,u}$ stores the amount of pheromone type S dropped at the link from peer P_i to peer P_u, for each segment s and each neighbor peer P_u. At startup, all table entries are initialized with the same small value τ_{init}. Table 1 shows the pheromone table at each peer node.

Table 1 The pheromone table at peer *i*

Pheromone type	Peer			
	P_1	P_2	…	P_n
S_1	$\tau_{S1,i,P1}$	$\tau_{S1,i,P2}$	…	$\tau_{S1,i,Pn}$
S_2	$\tau_{S2,i,P1}$	$\tau_{S2,i,P2}$	…	$\tau_{S2,i,Pn}$
…	…	…	…	…
S_m	$\tau_{Sm,i,P1}$	$\tau_{Sm,i,P2}$	…	$\tau_{Sm,i,Pn}$

HASA algorithm

Every peer in the network can function as a media segments query originator peer, media segments providing peer, and/or intermediate peer. A high-level flow chart for these functions is described in Fig.1. When a peer wants to find some media segment, it sends forward ants searching for this segment (Fig.1a). Forward ants move in the P2P network searching for the segment according to the intermediate peers' forwarding rule which uses pheromone tables (Fig.1b). Forward ants collect paths' information and intermediate peers' local information as they travel. The intermediate peers who are visited by forward ants update their pheromone table based on information carried by the forward ants. When a forward ant finds the segment at the providing peer, the providing peer's pheromone table will be updated. Then the forward ant will be killed and a backward ant will be generated (Fig.1c). The backward ant carries its corresponding forward ant's intermediate nodes identifications. The backward ant is sent back following the reverse path of its corresponding forward ant. As backward ants move in the reverse path, they collect the reverse paths' information and intermediate peers' local information as they travel. The intermediate peers modify their pheromone table based on the information carried by the backward ant (Fig.1d). Finally, the originator peer eventually receives the backward ants, updates its pheromone table and kills the backward ant (Fig.1e).

(1) Originator Peer (Fig.1a)

When a peer wants to search a segment, it generates forward ants. A forward ant carries the path source address, the searched segment ID and the cached segment ID. All the forward ants generated at the same peer that search for the same segment have a same unique description ID, we name them the same target forward ants. The next peers are selected by the forwarding rule.

When a backward ant is received at the source peer, the pheromone table will be updated using the pheromone updating rule and the backward ant will be killed.

(2) Intermediate Peer (Fig.1b)

Every peer has a pheromone table. Each required segment has an entry in these tables. In the pheromone table, each entry has the pheromone trails' values corresponding to all the intermediate peer's available neighbors.

The intermediate peer performs two main functions. The first function works when receiving a forward ant, and the second one works when receiving a backward ant.

When they receive the forward ants, the intermediate peers judge whether the forward ant

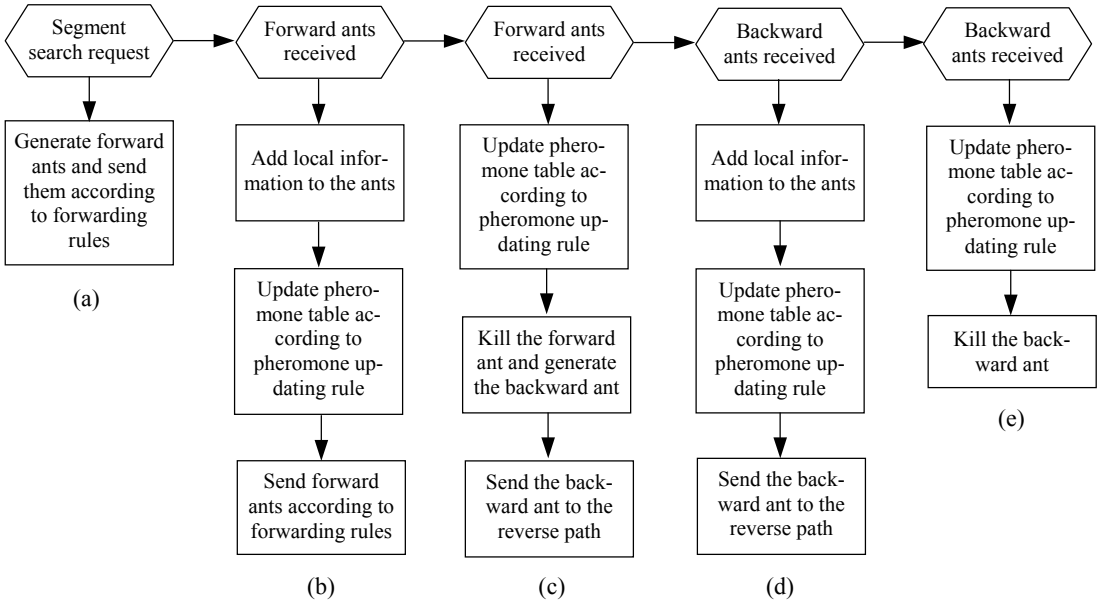

Fig.1 The high level flow chart of HASA. (a) Originator Peer; (b) Intermediate Peer; (c) Providing Peer; (d) Intermediate Peer; (e) Originator Peer

experiences any unwanted path constraint [looping, time to live (TTL) value, etc.], or they have received the same target forward ants before. If either of them is true, Intermediate Peer kills the forward ant. Otherwise the peer collects the path information from these forward ants and the pheromone table will be updated using the pheromone updating rule. The intermediate peer *i* inserts its ID "*i*" into the forward ants and adds and/or modifies the forward ant path information (such as number of hops, traffic load, etc.), then forwards them according to the forwarding rule.

When a backward ant is received at intermediate peer *i*, the peer collects the path information from this backward ant and the pheromone table will be updated using the pheromone updating rule. The intermediate peer *i* adds and/or modifies the backward ant path information, and then the backward ant will be sent to the next peer in the reverse path of the forward ant.

(3) Providing Peer (Fig.1c)

When a forward ant is received at the providing peer, the providing peer judges whether it has received the same target forward ants before. If do so, the providing peer kills the forward ant; otherwise the pheromone table will be updated using the pheromone updating rule and the forward ant will be killed. A backward ant will be generated and will carry all the IDs of the intermediate peers visited by the corresponding forward ant. The backward ant will be sent to the next peer in the reverse path of the forward ant.

Forwarding rule

Both intermediate and originator peers forward the forward ants in the same way. When a forward ant is received or generated, this ant will be forwarded to *K* neighbor peers except the nodes in the ant's path. The selection of these *K* neighbor nodes is done using the probability forwarding table. The values of the probability forwarding table are calculated using the pheromone tables. *K* nodes of the neighbor nodes will be selected and the ant will be forwarded to them. The selection of the next nodes is done as follows.

At node *i*, for segment *s*, the probability of selecting a neighbor *j* is:

$$prob(s,i,j)=\begin{cases}\dfrac{Fun(\tau_{s,i,j},\tau_{s-1,i,j},\ldots,\tau_{s-b,i,j})}{\sum_{l\in M_i}Fun(\tau_{s,i,l},\tau_{s-1,i,l},\ldots,\tau_{s-b,i,l})}\cdot K, & j\in M_i,\\ 0, & j\notin M_i,\end{cases} \tag{1}$$

where $Fun(\tau_{s,i,j}, \tau_{s-1,i,j}, \ldots, \tau_{s-b,i,j})$ is a function of the pheromone values for segment *s* and the former *b* segments before *s* on link (i,j). Example of such function is:

$$Fun(\tau_{s,i,j},\tau_{s-1,i,j},\ldots,\tau_{s-b,i,j})=\sum_{l=0}^{b}\tau_{s-l,i,j}w_l,$$

where $\sum_{l=0}^{b}w_l=1$.

M_i is the set of all feasible neighbor nodes that satisfy the constraints such as the guarantee of loop free. K is the number of the neighbor nodes which are selected.

Because

$$\sum_{j\in M_i}prob(s,i,j)=\sum_{j\in M_i}\frac{Fun(\tau_{s,i,j},\tau_{s-1,i,j},\ldots,\tau_{s-b,i,j})}{\sum_{l\in M_i}Fun(\tau_{s,i,l},\tau_{s-1,i,l},\ldots,\tau_{s-b,i,l})}\cdot k=k,$$

there must be k nodes of M_i to which ants are forwarded. The pseudo code of the forwarding rule is shown in Fig.2.

```
Set N=0;
While N<K
  Set ω=0;
  Generate random probability prob∈(0,1);
  For x=p1: pn
    If x∈M_i
      Calculate
        ω ← Fun(τ_{s,i,x},τ_{s-1,i,x},...,τ_{s-b,i,x}) / Σ_{l∈M_i} Fun(τ_{s,i,l},τ_{s-1,i,l},...,τ_{s-b,i,l}) + ω;
      If prob<ω
        N←N+1;
        Ant is forwarded to node x, and x is deleted
          from M_i;
        Break;
      End
    End
  End
End
```

Fig.2 The pseudo code of the forwarding rule for segment *s* at node *i*

Pheromone updating rule

Whatever originator peers or providing peers may cache some media segments in media streaming distribution unstructured P2P networks. When originator peers generate queries, they publish what they cache in the forward ants. So in HASA algorithm the pheromone table is updated when peers receive forward ants or backward ants. When a peer joins a media distribution, we calculate the elapsed time into equal intervals. When receiving ants, peers update pheromone table using Eq.(2). At the beginning of the nth interval, peers update pheromone table using Eq.(3).

$$\tau_{s,i,j}(n)=\begin{cases}\tau_{s,i,j}(n)+\nabla\tau_{s,i,j}(n,a_m), & \text{for the incoming link,}\\ \tau_{s,i,j}(n), & \text{for other links,}\end{cases} \tag{2}$$

$$\tau_{s,i,j}(n)=\tau_{s,i,j}(n-1)\rho_s, \tag{3}$$

where $\tau_{s,i,j}(n)$ is the pheromone value for segment s corresponding to neighbor j at node i after passing n intervals with n being the number of intervals. The time interval is set according to the cache replacement frequency of peers. ρ_s is the evaporation parameter of segment s. The evaporation parameter is used to help the system forget the old information faster. $\nabla\tau_{s,i,j}(n,a_m)$ is the pheromone left by the mth ant for segment s corresponding to neighbor j at node i in the nth interval. $\nabla\tau_{s,i,j}(n,a_m)$ is used to help the system increase the amount of pheromone on the edges. $\nabla\tau_{s,i,j}(n,a_m)$ are different at peers left by the same ant. The closer peer i to the originator or the providing peer, the more the pheromone left by the ant. Example of such function is:

$$\nabla\tau_{s,i,j}(n,a_m)=w_pI_{path,i}+w_h(C-h_{s,i}),$$

where C is a constant, it should be larger than the longest hop for searching any segment s in the network. $h_{s,i}$ is the hop that the ant has travelled from the originator or providing peer to the intermediate peer i for segment s. $I_{path,i}$ is the path information from the originator or providing peer to the intermediate peer i. w_p is the path information weight, and w_h is the hop weight, $w_p+w_h=1$.

Path information collection

The path information may include the peer's local information and global path information. This information should be the parameters we would like to optimize, such as "quality of service". We should be very careful in selecting the path information to be collected. The more information collected about the peers and path, the more close to the optimum. However, the sizes of the ants will increase, which

increase the overlay network overheads. These overheads in MANETs consume significant part of the network's energy and bandwidth.

In this paper we want to optimize the stability of the path and the balance of the peers to obtain low latency and reliable media streaming distribution. In HASA, we adapt the strategy used in ARAMA (Hussein and Saadawi, 2003) to collect path information.

PERFORMANCE EVALUATION

HASA is evaluated by simulating a P2P system with 1600 peers and by comparing its performance with that of the well-known K random walks search (KRS) approach (Lü *et al*., 2002). In this experiment, we take the following methodology. First, a small-world network is generated using the model described and analyzed by Kleinberg (2000). Second, we generate workloads to drive the simulation. A long media stream is divided into 20 segments. Initially, all segments are stored in one peer in the network. Then streaming accesses are triggered at random points in the network. A streaming access may request the complete media stream, or a partial stream. In both cases, the access starts with requesting the first segment, and may stop in the middle. Each node has enough memory space for caching three segments. To model the popularity of the segments, a Zipf-like distribution is used. Because the display time of a segment fixed assuming a CBR continuous media, to simplify discussion, we assume the time to retrieve a segment one hop away is fixed, which equals the display time of a segment for CBR media streaming. For both algorithms, assume that the search process is terminated when the query is hit.

The optimal values for the configurable parameters of HASA are mostly dependent on the network topology changing speed and on the content distribution within the network. Especially the optimal values for ρ_s and the elapsed time interval T of HASA should mate the topology changing speed. Table 2 shows the performance comparison of start-up delay and average transmitting delay at different settings of ρ_s and interval T for HASA in the same simulation scene. The other parameters were chosen as follows: b=1, w_0=0.8, w_1=0.2, w_p=w_h=0.5, K=2, TTL=15. It shows that the performance is better when ρ_s=0.1 and T is set to 15 s. Although the performances are different, they are all better than K random walks algorithm because of adopting the probability forwarding rules. It is planned to investigate this issue in more detail with the goal of finding a method for determining the optimal parameter settings at runtime.

Table 2 The performance comparison between different ρ_s and T for HASA and K random walks

Algorithm	T (s)	ρ_s	Start-up delay (hops)	Average transmitting delay (hops)
HASA	15	0.2	2.74684	3.01555
		0.3	2.82278	3.04737
		0.4	2.92405	3.07476
	10	0.1	2.71519	2.97249
	20	0.1	2.79620	3.14115
	25	0.1	2.82911	3.09629
			2.81646	3.10048
K random walks			3.14557	3.50538

In the comparison experiments between HASA and K random walks algorithms described here, the parameters of HASA were chosen as follows: T=15 s, b=1, w_0=0.8, w_1=0.2, ρ_s=0.1, w_p=w_h=0.5, K=2, TTL=15. To provide a fair comparison, the TTL and K of K random walks algorithm are set to 15 and 2 respectively, while the simulation time of the two algorithms is set to 300 s.

Three important matrixes are set to evaluate the performance of HASA. The main performance matrix is the average number of hops in order to discover and retrieve a replica of a segment. It means the delay in finding and retrieving a segment as measured in the number of hops. The second performance matrix is the jitter times for multimedia segments. Due to the stringent resource restrictions at the mobile hosts, the jitter times for displaying media streaming segment is important for media streaming distribution in wireless ad hoc networks. It refers to the user's experience for media streaming distribution on networks. The jitter times T_j we considered here is the interval between the segments arrival time T_{arrival} and the nonstop playing back time T_{deadline}, T_j=T_{arrival}−T_{deadline}, if T_{arrival}−T_{deadline}<0, then set T_j=0. And the third matrix, the average number of messages, has to propagate in the P2P network for each query. It is the overhead of an

algorithm as measured in the average number of search messages per query. The motivation for this matrix is that in P2P systems, especially in power limited wireless P2P networks, the most notable overhead tends to be the unnecessary transmitting load that the network imposes on each participant. Fig.3 shows the performance comparison for the three matrixes of *K* random and HASA search algorithms.

The simulation results show that the HASA algorithm is better than the *K* random walks algorithm at segment transmitting delay and playing back jitter times, especially for the latter media segments distribution. HASA algorithm forwards queries following the trails for latter generated searches, but *K* random walks still forwards queries randomly. So HASA algorithm can find the queried segment much quickly (Fig.3a). Because HASA algorithm concerns with the segments correlations of media streaming, the latter segments' jittery delay is less than the former segments and even to zero (Fig.3b). And Fig.3c shows that the average number of propagated messages for each query imposed on the peers of HASA algorithm is less than that of K random walks algorithm. It means that HASA algorithm generates less unnecessary query messages, because peers forward the latter queries following the former trails in HASA, which can effectively avoid forwarding the queries to the useless peers. Then it can save the energy for wireless nodes to transmit unnecessary query messages. So the HASA algorithm is much more suitable for the media streaming delivery in a P2P fashion in wireless ad hoc networks.

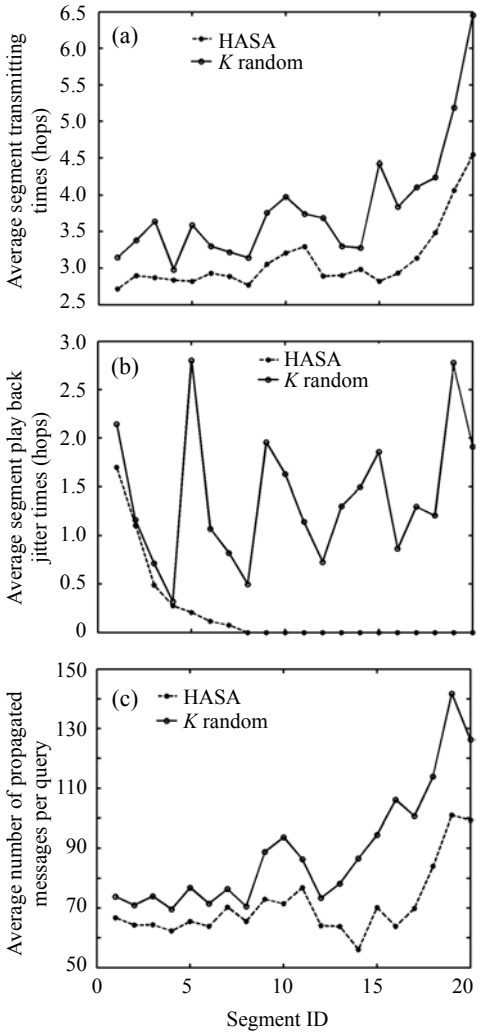

Fig.3 The average parameters. (a) Segment transmitting delay; (b) Segment jitter times; (c) Number of propagated messages per query

CONCLUSION AND FUTURE WORK

The HASA algorithm uses the *K* random walks and ants-like based search strategies for searching a segment. When receiving queries and responses, peers update their pheromone tables immediately, then use the pheromone trails to forward *K* ants for searching segments at each intermediate peer or originator peer. It is different from *K* random walks algorithm and traditional ants-like algorithm. First, peers forward query messages according to the pheromone trails instead of random selection. Secondly, peers forward query messages to *K* neighbor nodes instead of one. Thirdly, peers update the pheromone trails when receiving backward ants or forward ants. And fourthly, by exploring the correlation of media streaming segments, peers forward query messages according to not only the same segment's pheromone trails but also the former segment's pheromone trails. HASA takes the advantages of *K* random walks and ants-like search algorithms, so it can avoid most unnecessary traffics, while keeps the low transmitting latency and less jitter time for media streaming distribution in wireless ad hoc networks.

Performance of the algorithm is different in dynamic settings in which peers are changing their moving speeds and the nodes joining and departing rates are changed. In these cases, it is necessary to

adapt the pheromone trails in order to reflect these changes. To correct the amount of pheromone, a self-adaptive strategy will be researched, such as how to change the parameters according to these changes at each peer node.

ACKNOWLEDGEMENT

We thank Dr. Tai Wang for his participating in the early stage of the work.

References

Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A., 2001. Search in power-law networks. *Phys. Rev. E*, **64**(4): 046135. [doi:10.1103/PhysRevE.64.046135]

Bonabeau, E., Dorigo, M., Theraulaz, G., 2000. Inspiration for optimization from social insect behavior. *Nature*, **406**:39-42. [doi:10.1038/35017500]

Ghandeharizadeh, S., Krishnamachari, B., Song, S.S., 2004. Placement of continuous media in wireless peer-to-peer networks. *IEEE Trans. on Multimedia*, **6**(2):335-342. [doi:10.1109/TMM.2003.822787]

Gkantsidis, C., Mihail, M., Saberi, A., 2005. Hybrid Search Schemes for Unstructured Peer-to-Peer Networks. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, **3**:1526-1537. [doi:10.1109/INFCOM.2005.1498436]

Hussein, O., Saadawi, T., 2003. Ant Routing Algorithm for Mobile Ad-Hoc Networks (ARAMA). Proc. 2003 IEEE Int. Conf. on Performance, Computing, and Communications, p.281-290.

Jiang, S., Guo, L., Zhang, X., 2003. LightFlood: An Efficient Flooding Scheme for File Search in Unstructured Peer-to-Peer Systems. Proc. 2003 Int. Conf. on Parallel Processing, p.627-635. [doi:10.1109/ICPP.2003.1240631]

Jiang, H., Jin, S., 2005. Exploiting Dynamic Querying like Flooding Techniques in Unstructured Peer-to-Peer Networks. Proc. 13th IEEE Int. Conf. on Network Protocols, p.122-131. [doi:10.1109/ICNP.2005.17]

Jin, S., 2004. Replication of Partitioned Media Streams in Wireless Ad Hoc Networks. Proc. 12th Annual ACM Int. Conf. on Multimedia. New York, USA, p.396-399. [doi:10.1145/1027527.1027621]

Kleinberg, J., 2000. The Small-World Phenomenon: An Algorithm Perspective. Proc. ACM Symposium on Theory of Computing, p.163-170.

Lü, Q., Cao, P., Cohen, E., Li, K., Shenker, S., 2002. Search and Replication in Unstructured Peer-to-Peer Networks. Proc. 16th ACM Intl. Conf. Supercomputing, p.84-95. [doi:10.1145/511334.511369]

Michlmayr, E., 2006. Ant Algorithms for Search in Unstructured Peer-to-Peer Networks. Proc. 22nd Int. Conf. on Data Engineering Workshops, p.x142-x142. [doi:10.1109/ICDEW.2006.29]

Xue, G., Jia, Z., You, J., Li, M., 2004. Group Mobility Model in Mobile Peer-to-Peer Media Streaming System. Proc. IEEE Int. Conf. on Services Computing, p.527-530.

Yang, B., Garcia-Molina, H., 2002. Improving Search in Peer-to-Peer Networks. Proc. 22nd Int. Conf. on Distributed Computing Systems, p.5-14.

Zhuang, Z., Liu, Y., Xiao, L., Ni, L.M., 2003. Hybrid Periodical Flooding in Unstructured Peer to Peer Networks. Proc. Int. Conf. on Parallel Processing, p.171-178. [doi:10.1109/ICPP.2003.1240578]