A Topology Adjustment Algorithm for P2P Systems

Zhen Yu, Jie Zhu, Guicheng Shen
School of Information, Beijing Wuzi University, Beijing, China
Email: yuzhenhappy@163.com

Haiyan Liu
State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing, China
Email: 48801343@qq.com

***Abstract*—In order to improve the Peer-to-Peer (P2P) network resource locating efficiency and isolate malicious peers, a topology adjustment algorithm for unstructured P2P networks is proposed. TTL (time-to-live) value for each neighbor has been set up and adjusted; specific information on the query path is recorded; a topology adjustment algorithm is given. The topology adjustment algorithm is evaluated by peer shortest path, simulation and analysis show that the topology adjustment algorithm can isolate malicious peers to the network edge, and improve service quality of P2P networks.**

***Index Terms*—Peer-to-Peer; unstructured; topology; shortest path**

I. INTRODUCTION

The P2P system is essentially a distributed system without the central server. In the P2P system, each peer interconnects with other and forms a self-organizing overlay network [1]. The P2P system is a virtual overlay over the underlying physical network topology, which can provide supports for all types of P2P applications. The topology of P2P networks is closely related to the operation of the P2P system, and affects the quality of service. Some studies suggest that optimizing the structure of the P2P network topology can improve resource location efficiency, and reduce overhead, improve service the quality of P2P networks.

According to different manners of topology constructing, P2P systems can be divided into unstructured P2P systems, structured P2P systems, and other systems with new structures [2]. Currently, the most popular P2P applications are based on unstructured P2P systems [3]. Unstructured P2P systems based on a random topology structure and its flooding way have a large number of network redundancies of query messages affecting the efficiency of resource orientation. In the unstructured P2P network, a "good" topology can target resources more effectively, including the ability to find the target service peer, and the number of duplicate query messages, and the isolation of malicious peers, and so on. In paper [4-6], topology optimization techniques have been proposed in some aspects, but not consider the problem of malicious peers in the network. In paper [7, 8], the topology is adjusted by the credibility of peers, but the calculation of abilities of neighbors is one-sided, and fails to adjust to the fairness of topology. To improve the efficiency of resource orientation, and isolate malicious peers, this paper presents a topology adjustment algorithm for unstructured P2P networks, and the effectiveness of topology has been verified based on the shortest path. Simulation and analysis show the effectiveness of the algorithm.

The rest of this paper is organized as follows: section Ⅱ discusses related work. Section Ⅲ introduces our approach to record forwarding information of the algorithm. Section Ⅳ gives detailed explanations and analyses of the topology adjustment algorithm. Section Ⅴ presents results of experiments testing the algorithm. In the last section, we present conclusions.

II. RELATED WORK

The FastTrack [5] is based on two-tier architecture, P2P protocol, and KaZaA [9] and Grokster [10] are applications based on the protocol. The protocol chooses peers with higher performance as a super-peer, and other peers as normal peer. Paper [11] introduced some peers with the relative strong ability as super-peers hosts more calculations in order to speed up the routing and query speed. Paper [12-14] also present topology optimization methods based on the processing capacity of a peer, respectively, according to peer degree, routing, or a variety of metrics to select the neighbor comprehensive. ACE (Adaptive Connection Establishment) [15] in each peer to its neighbors to establish a minimum spanning tree to solve the topology mismatch problem. Paper [16] presents a dynamic adaptive algorithms and search methods, for how to use the user's interest to form a different cluster. Paper [17] proposed an approach in the P2P network to forming some "cluster" among peers with similar interests; which can get a response within a short distance. However, the above topological optimization methods did not take malicious peers into account. A class of topological optimization methods based on peers' credibility adjusted the topology by the credibility of servicers; peers tended to select some peers with higher

doi:10.4304/jsw.9.8.2217-2222

trust value as their neighbors and malicious peers providing unreliable services had smaller chances of being chose as a neighbor. Paper [18] proposed a reliability-sensitive topology evolutionary algorithm. Paper [19] proposed a topology model based on profit sharing. Paper [20] proposed a novel distributed network and implemented resource location algorithms with the hierarchical combination of grid and P2P, which using SW-R2P as the underlying P2P component and the Globus Toolkit as the grid component.

APT [7] proposed a peer-level protocol for forming adaptive topologies for data-sharing P2P networks. However, the query in APT protocol only recorded the first neighbor who forwarded the query and disposed the first neighbor finally by the result returned from the interaction, which was unfair to the first neighbor and lead to false operation of normal peers. In order to improve the deficiency in APT protocol, a topology concerning direct forwarding peers is proposed in this paper, each neighbor of the peer possess a changeable and reliable TTL which is changed after each interaction by its transmitting condition, and every transmitting condition of direct forwarding peers in query paths is also recorded; some storage structures are used to store peers' transmitting information in the topology, and relative algorithms are also proposed to adjust the topology of P2P systems.

III. Locating Method of Forwarding Information

P2P networks can be represented as an undirected graph $G=(P,E)$, where P is peer set and E is edge set, (i,j) is the connect between peer *i* and peer *j*, $i,j\in P$ [7]. Neighbor set of peer *i* can be expressed as $N(i)=\{j\mid(i,j)\in E\}$.

Peers can use a mechanism similar to Gnutella network: the peer connects to the network by some other peers already been in the network and forward query messages by way of flooding with the TTL to control the scale of queries. Taking P2P file sharing networks as the example, the inter-peer resource locating method is described as follows:

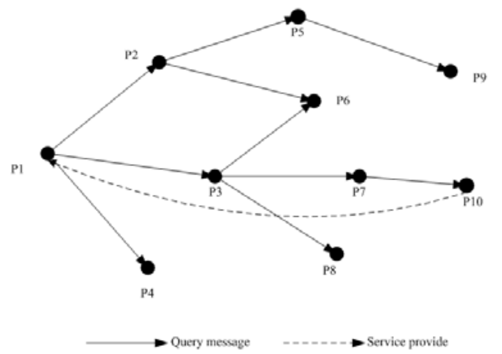

Figure 1. Resource location example

When the resource location of peer P1 is concerned, the query path is P1-P3-P7-P10, of which P3 and P7 are forwarding peers, and P3 is a neighbor of P1 who send out the initial request, in addition, P7 is called the direct forwarding peer corresponding to the service peer P10. The query path generally have multiple forwarding peers, and the direct forwarding peer in the forwarding path plays a crucial role compared to other forwarding peers. Consequently the topology proposed in the paper, in addition to recording results of neighbors in query paths, also records information of direct forwarding peers, and adjusts the topology based on these information.

In the adjustment algorithm, each peer in the topology has the corresponding neighbor list and connection factor list. The neighbor list of Peer *i* is showed as Table 1, which remember each neighbor ID and their credible TTL value from Peer *i*' s point of view.

TABLE I.
INFORMATION OF NEIGHBORS

Neighbor ID	Credible TTL
n_1	$TTL^i_{n_1}$
n_2	$TTL^i_{n_2}$
…	…
n_k	$TTL^i_{n_k}$

The connection factor list of peer *i* is showed as Table 2, which remember the ability of a direct forwarding peer t_k connecting to a server from Peer *i*' s point of view. $f^i_{t_k}$ represents a connection factor of a direct forwarding peer t_k, $Succ^i_{t_k}$ indicates the number of successful response received by peer *i* from the transfer of direct forwarding peer t_k; $Fail^i_{t_k}$ indicates the number of failed response received by peer *i* from the transfer of direct forwarding peer; $TranSat^i_{t_k}$ represents the contribution from t_k as a direct forwarding peer to the current interaction, set as 0.5 when successful and -0.5 when fail. The connection factor is defined as follows:

$$f^i_{t_k}=\frac{f^i_{t_k}\times\left(Succ^i_{t_k}+Fail^i_{t_k}\right)+TranSat^i_{t_k}}{Succ^i_{t_k}+Fail^i_{t_k}+1}$$

(1)

TABLE II.
INFORMATION OF CONNECTION FACTOR

Transmit Peer ID	Connect Factor	Success Number	Fail Number
t_1	$f^i_{t_1}$	$Succ^i_{t_1}$	$Fail^i_{t_1}$
t_2	$f^i_{t_2}$	$Succ^i_{t_2}$	$Fail^i_{t_2}$
…	…	…	…
t_k	$f^i_{t_k}$	$Succ^i_{t_k}$	$Fail^i_{t_k}$

IV. Topology Adjustment Algorithm

First, some definition of several parameters to be used in topology adjustment algorithm is given below:

$Nb(i)$: The neighbor set of peer *i*;

Nb_{max} : The Max number of neighbors for a peer connecting;

$Min(S)$: The peer with lowest cumulative trust in set *S*;

$Max(S)$: The peer with highest cumulative trust in set *S*;

$FMax(S)$: The peer with highest cumulative connection factor in set *S*;

$Num(S)$: The number of peers in set *S*;

$TTL(i,n)$: The neighbor set whose credible TTL value is n of peer *i*;

$Local(i)$: Peer set who has provided services for peer *i*;

$Tran(i)$: Peer set who has provided query forwarding for peer *i* and goes directly to respond peers;

j_d : The direct forwarding peer of service peer *j*;

$SumTrust(i, j)$: From the view of peer *i*, peer *j*'s good cumulative trust value;

$SumTran(i, j)$: From the view of peer *i*, peer *j*'s good accumulated connection factor, $f_j^i * (Succ_j^i - Fail_j^i)$.

Related primitives are as follows:

Disconnect (i, j): Peer *i* disconnects the connect with peer *j*;

AddNeighbor (i, j): Peer *i* add peer *j* as a neighbor;

RequestConnect (i, j): Peer *i* request to establish a connect with peer *j*, if request is received, return *true* , else return *false* ;

PickNewNeighbor (i): Peer *i* pick a new neighbor.

Peers need to update the local service information and recommendation information, connection information after each interaction. When getting to adjusting time of the topology, the system performs different adjustment procedures based on the satisfaction degree to the service peer.

A. Topology Adjustment with Service not Satisfying

If the peer *i* is not satisfied with the service provided by peer *j*, it need to adjust its neighbors and TTL values.

TTL indicates the number of query messages transmitted in the network forward. If a credible TTL value of a neighbor is 0, it will decrement to 0 and does not forward the query message again, and at this time the neighbor no longer bear the forwarding function. The topology in the paper specified the number of neighbors whose TTL value is 1 cannot exceed half of the total number of neighbors, which is to ensure the proliferation of the number of queries and responses. Because neighbors whose TTL is 1 no longer forward the query message and the number of which is supposed to be controlled consequently, and forwarding functions should be bore by other neighbors whose TTL value are greater than 1. TTL values are set artificially at the beginning to control the scale of the query message delivered in P2P systems, depending on the actual situation of the network, for example, TTL is usually set to 7 in Gnutella. The TTL value is set to 4 in the simulation of this topology which can ensure the flooding of query messages. The storage overhead includes information corresponding to neighbors' credible TTL value and direct forwarding peers,.

Adjustment algorithm is as follows:

```
Procedure ChangeTTL( ID_i , ID_j )

Update f^i_{j_d} in connect_table;
Update TTL^i_{n_k} in ttl_table;
if TTL^i_{n_k} ==0 then
  Disconnect(i, n_k );
  PickNewNeighbor(i);
else if TTL^i_{n_k} ==1 then
  if D_{in_k} ≥ threshold_d && SumTrust(i, n_k) ≥ threshold_s then
    if Num(TTL(i,1)) ≥ Num(Nb(i))/2 then
      if SumTrust(i, n_k) > Min(TTL(i,1)) then
        Disconnect(i, Min(TTL(i,1)) );
        PickNewNeighbor(i);
      else
        Disconnect(i, SumTrust(i, n_k)) );
        PickNewNeighbor(i);
      end
    end
  else
    Disconnect(i, n_k );
    PickNewNeighbor(i);
  end
end
End
```

If the peer *i* is not satisfied with the service provided by peer *j*, TTL values of its neighbors need to be adjusted. First, a requesting peer *i* need to update the connection factor of the direct forwarding peer in the query path, and then update credible TTL value of neighbors n_k in the query path according to the corresponding number of hops $Hops_j^i$ to the service provider *j* $TTL_{n_k}^i = Hops_j^i - 1$ ($Hops_j^i$ indicates the number of messages has been forwarded when reaching the service provider). If $TTL_{n_k}^i$ =0, or service provider *j* is the neighbor of the requesting peer *i*, peer *i* disconnect with peer j and pick a new neighbor. If $TTL_{n_k}^i$ =1, the query message was forwarded to the neighbor n_k and the corresponding TTL value decreases to 0, and the neighbor n_k cannot forward the query message to its neighbors anymore and only can be a responding peer, so the requesting peer *i* need to determine the neighbor n_k is worth it. The algorithm needs to query the neighbor list, and the algorithm complexity is $O(n)$, *n* is the scale of the system.

B. *Topology Adjustment with Service Satisfying*

If the peer *i* is satisfied with the service provided by peer *j*, and peer *j* is not its neighbor, the adjustment algorithm is as follows.

First, the requesting peer *i* updates the connection factor of direct forwarding peer in the query path, and then view the table of connecting factor information if there is a negative record of connection factor of the service provider. If not with negative forwarding record, peer *i* will try to add the service provider as its neighbor. Considering the different number of neighbors compared with $Nb_{\max}$, there are different measures to add a new neighbor. Service provider or direct forwarding peer j_d could be selected as a new neighbor. The Cumulative trust value of a service provider can be computed by the number of successes and failures. The satisfaction threshold in the algorithm is typically set to 0.5, which can be adjusted by actual situations.

The requesting peer updates the connection factor of the direct forwarding peer in the query path, and then adopts different strategies according to whether the number of neighbors has got the maximum. The algorithm needs to query the neighbor list, and the algorithm complexity is $O(n)$, *n* is the scale of the system.

Procedure UpdateNb(ID_i , ID_j)
Update $f_{j_d}^{i}$ in connect_table;
if $f_j^i \geq threshold_f$ then
  if $j \notin Nb(i)$ && $Num(Nb(i)) < Nb_{\max}$ then
    if (RequestConnect(i, j) $== true$)
      AddNeighbor(i, j);
      return form function;
    else if $f_{j_d}^{i} \geq threshold_f$ && $D_{ij_d} \geq threshold_d$ then
      if (RequestConnect(i, j_d) $== true$)
        AddNeighbor(i, j_d);
        return form function;
      end
    end
  end
  if $j \notin Nb(i)$ && $Num(Nb(i)) \geq Nb_{\max}$ then
    if $D_{iMin(Nb(i))} < threshold_d \parallel SumTrust(i, j) > Min(Nb(i))$ then
      if (RequestConnect(i, j) $== true$)
        Disconnect(i, $Min(Nb(i))$);
        AddNeighbor(i, j);
        return form function;
else if $f_{j_d}^{i} \geq threshold_f$ && $D_{ij_d} \geq threshold_d$ && RequestConnect(i, j_d) $== true$ then
        Disconnect(i, $Min(Nb(i))$);
        AddNeighbor(i, j_d);
        return form function;
      end
    end
  end
end
End

V. Simulation and Results

A. *Simulation Environment*

In this paper, the query cycle model [21, 22] is used as the simulator, which constructs a P2P file-sharing network. Each simulation consists of some simulation cycles. In each simulation cycle, the peer in the network can initiate queries and response to queries; queries are broadcast like Gnutella, via TTL control the size of the query. Peers initiating the query will wait to receive the response and select the highest trust value of nodes from the response list to download the file. In the simulation, peers in the network are divided into two categories: good peers and malicious peers, good peers provide reliable service, and malicious peers provide unreliable service. The simulation environment setting is showed in Table 3.

TABLE III.
Simulation Settings

Network	Topology construct		Power-law
	peer scale		100
	malicious peer in all		20%
	Min neighbor		3
	Max neighbor		10
	TTL for query messages		4
Service	Evaluation criteria		Download speed File Quality
Peer	Good Peer	Active	100%
		Query	100%
		To respond	march
		Forward request	100%
		Request file	Random
	Malicious Peer	Active	100%
		Query	100%
		To respond	50%
		Forward request	100%
		Request file	Random
Content	Content category		20
	Shared file content		Zipf
	Distribution of shared file in content		Random
Simulation	Query cycles		200

B. *Topology Performance Evaluation*

In this paper, we use the characteristic path length to evaluate the peer network topology. The average characteristic path length of peer *i*: $cpl_i = \frac{1}{\left|P \setminus i\right|} \sum_{j \in P \setminus i} shortestPath(i, j)$, which is the average of shortest paths between peer *i* and all other peers in the network, P / i means that the peer group in the network without containing peer *i*. When there is no path between a peer and some other peers, then set the shortest path as 15.

Figure 2 shows the average CPL between each peer and all other good peers in the first cycle and the 200th query. Peer 1 to peer 80 is good peers. After the 200th cycle, the average CPL from good peer to all other good peers did not change significantly, but the average CPL from most of the malicious peers to good peers is greatly increased.

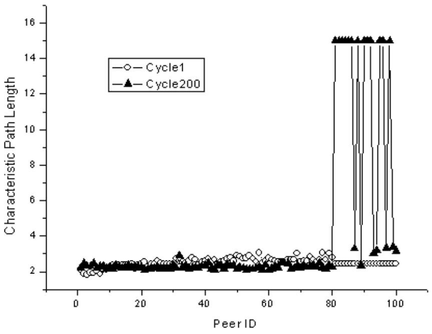

Figure 2. CPL of peers

Figure 3 shows the average characteristic path length from each good peer to all other malicious peers in the first cycle and the 200th query. The average characteristic path length from good peers to other malicious peers increases as the cycle increases, reducing malicious peers to provide unreliable services.

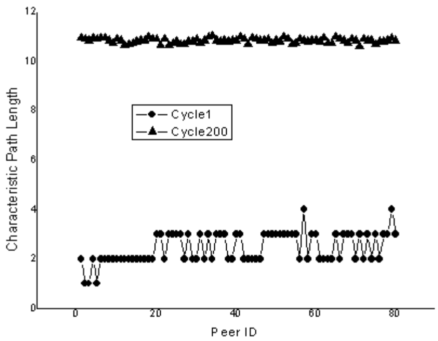

Figure 3. CPL of good peers

Figure 4 shows the change of the average characteristic path length from good peers and malicious peers to all good peers when the cycle increasing. In Figure 4, the average characteristic path length from good peers to other good peers is very stable at a low value; while the average characteristic path length from malicious peers to other good peers is increased with the increasing cycle. Malicious peers are excludes to the network edge gradually.

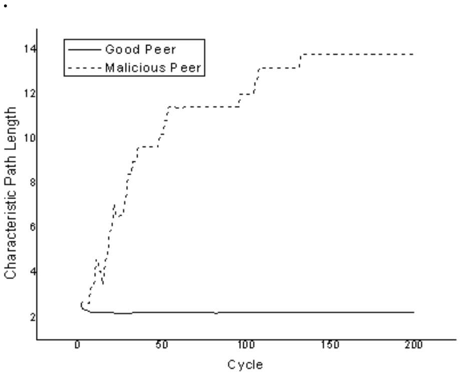

Figure 4. CPL in cycles

CONCLUSION

To improve the efficiency of resource locating and isolate malicious peers, this paper presents a topology adjustment algorithm for an unstructured P2P network. This paper sets the variable TTL value for each neighbor, not only ensure the spread of queries, but also reduce query messages in the network; also records the condition of forwarding peers in query paths. Simulation and analysis show the effectiveness of the topology adjustment algorithm, which can isolate malicious peers to the edge of the network and improve service quality.

ACKNOWLEDGMENT

The research is sponsored by Beijing key Laboratory of Intelligent Logistics System, Beijing Wuzi University. This paper is also supported by three projects including: Youth Research Funding Project of Beijing Wuzi University (2011XJQN003), Project for Beijing University Logistics Technology Engineering Research Center(BJLE2010), Special Funding Project for Education Quality Improvement of Beijing Wuzi University.

REFERENCES

[1] Lua E, Crowcroft J, Pias M, et al. A survey and comparison of Peer-to-Peer overlay network schemes. IEEE Communications Survey and Tutorial, 2005, 7(2): 72-93.

[2] Chen ZG, Liu JQ, Li D, Liu H. SOBIE: a novel supernode P2P overlay based on information exchange [J]. Journal of computers, 2009, 4 (9): 853-861.

[3] Tsoumakos D, Roussopoulos N. A comparison of peer-to-peer search methods. In: Proceedings of the International Workshop on the Web and Databases (WebDB), SanDiego, CA, US, 2003: 61-66.

[4] Sripanidkulchai K, Maggs B, Zhang H. Efficient content location using Interest-Based locality in Peer-to-Peer systems. In: Proceedings of the Conference on Computer Communications (IEEE Infocom'03). IEEE Press, 2003, 22: 2166-2176.

[5] Fasttrack peer-to-peer technology company. http://www.fasttrack.nu/.

[6] Liu Y H, Liu X, Xiao L, et al. Location-Aware topology matching in P2P systems. In: Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), Hong Kong, 2004.

[7] Condie T, Kamvar S D, Garcia-Molina H. Adaptive peer-to-peer topologies. In: Lambrix P, Duma C, eds. In: Proceedings of the 4th International Conference on Peer-to-Peer Computing. New York: IEEE Press, 2004: 53-62.

[8] Tian H R, Zou S H, Wang W D, et al. Incentive compatible adaptive topologies for P2P networks. Journal of software, 2006, 17(4): 845-853.

[9] KaZaa. http://www.kazaa.com.

[10] Grokster. http://www.grokster.com.

[11] Li J, Vuong S. An efficient clustered architecture for P2P networks. In: Proceedings of the 18th International Conference on Advanced Informatin networking and Application (AINA'04) AINA, 2004, 1:278-283.

[12] Chawathe Y, Ratnasamy S, Breslau L, et al. Making Gnutella-like P2P systems scalable. In: Feldmann A Zitterbart M,ed. In: Proc.of the ACM SIGCOMM 2003. ACM Press, 2003: 407-418.

[13] Crespo A, Garcia-Molina H. Routing indices for peer-to-peer systems. In: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02). IEEE Computer Society, 2002: 23-34.

[14] Zhuang Z, Liu Y, Xiao L, et al. Hybrid periodical flooding in unstructured Peer-to-Peer networks. In: Proceedings of

International Conference on Parallel Processing (ICPP'03). IEEE Computer Society, 2003: 171-178.

[15] Xiao L, Liu Y H, Ni L M. Improving unstructured Peer-to-Peer systems by adaptive connection establishment. IEEE Transactions on Computers (TC), 2005, 54(9): 1091-1103.

[16] Yang J, Zhong Y, Zhang S. An efficient Interest-Group based search mechanism in unstructured Peer-to-Peer networks. In: Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing (ICCNMC), 2003: 247-252.

[17] Ramanathan M K, Kalogeraki V, Pruyne J. Finding good peers in Peer-to-Peer networks. In: Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS'02). IEEE Computer Society, 2002: 232-239.

[18] Dou W. The research on trust-aware P2P topologies and constructing technologies [Ph.D. Thesis]. Changsha: National University of Defense Technology, 2003.

[19] Guo L T. Research on the trust aware resource transaction model in Peer-to-Peer Networks [Ph.D. Thesis]. Hefei: University of Science and Technology of China, 2007.

[20] Xia Y J, Zhu M Z, Li Y. Towards Topology-and-Trust-Aware P2P Grid [J]. Journal of computers, 2010, 5 (9): 1315-1321.

[21] QueryCycleSimulator [OL]. http://p2p.stanford.edu/www/demos.htm, 2009-03-10.

[22] Schlosser M, Condie T, Kamvar S. Simulating a file-sharing P2P network. In: Proceedings of the 1st Workshop on Semantics in P2P and Grid Computing (SemPGRid2003), Budapest, 2003: 69-80.

Zhen Yu was born in Shandong Province, P.R. China on autumn, 1983. Now she is a teacher in school of Information, Beijing Wuzi University. She received the bachelor's degree in 2000 from Shandong University, and received her Ph.D. Degree from the University of Science and Technology Beijing in January 2010. Her major is computer application. Her research interests include Peer-to-Peer network, etc.

Jie Zhu is a professor in school of Information, Beijing Wuzi University. He received the bachelor's degree in 1982 from Peking University. He was awarded the honored title of “outstanding young teacher in Beijing” in 1994. His major is computer science and technology. His research interests include system development, database application, etc.

Guicheng Shen was born in Jiangsu Province, P.R. China in 1966. Now he is a professor in school of Information, Beijing Wuzi University. He received the bachelor's degree and master’s degree from Beijing Normal University in 1990 and 1993, and received her Ph.D. Degree in the University of Science and Technology Beijing in 2009. His major is computer application. His research interests include information security, business intelligence, etc.

Haiyan Liu was born in Neimeng Province, P.R. China in 1978. Now she is doing her postdoctoral research in State Key Laboratory of Hydro science and Engineering, Tsinghua University, Beijing, China. She received her Ph.D. Degree from the University of Science and Technology Beijing in January of 2009. Her major is computer