

# Efficient Anonymous Communication in Peer-to-Peer Systems

CHING-WEI HUANG and WUU YANG

National Chiao-Tung University

HsinChu, Taiwan, R.O.C.

19th January 2004

## **Abstract**

Anonymity is critical for some network applications. However, current Internet protocols, such as TCP/IP, do not hide the addresses of the communicating parties. Two communicating parties know and must know each other's network addresses. We present an efficient implementation of anonymous communication in the application layer. Our implementation provides fast message routing and sufficient privacy protection. Our implementation of the anonymous P2P system needs no central servers nor a set of trusted hosts. We also observed that there is adjustable balance between the efficiency and the anonymity for the message communication

## **1 Introduction**

Anonymity is critical for some network applications. However, current Internet protocols, such as TCP/IP, do not hide the addresses of the communicating parties. Furthermore, eavesdroppers can easily spy the network activities, even if the contents of communications are encrypted, if the network addresses are not kept secret. The disclosure of network addresses and communicating activities may have adverse implications. Anonymous communication allows people to communicate with each other without fear of surveillance.

We analyzed the system structures and message delivery mechanisms in Gnutella [1] and Freenet [2] and found there were two ways to improve the performance under anonymity message delivery: a better self-organization mechanism and a better message delivery mechanism.

For anonymous systems such as Gnutella and Freenet, the connections between nodes are usually fixed in order to provide maximal anonymity of nodes. Without the connections to new nodes, every node is known only to its neighbors. Every message delivery completely relies on the cooperation of nodes no matter the message is a broadcast request or a reply from one node to another. If we insist on the absolute anonymity for every node, then every message delivery between nodes will be expensive. For this reason, we propose a compromised anonymity mechanism by which every node knows at most  $O(\log(n))$  nodes, where  $n$  is the number of nodes in the system, while the cost of message delivery drops dramatically. Our system, *Efficient Anonymous System (EAS)*, tries to group all nodes into connected clusters. Closer nodes get higher priority to connect into a cluster such that the average cost of message delivery between nodes could be reduced.

Unlike the Gnutella or Freenet which send results back to the request originators on the backward travelling path of the request. Our EAS system makes use of anonymous proxies to relay the reply message back to the originator. One significant benefit of the proxies is that they eliminate the need of return path. Therefore, the traffic on the return path could be reduced completely. Our EAS system is based on our previous work of DSE [3]. The DSE peer-to-peer system provides a self-organization mechanism. It provides efficient message broadcast but no anonymity services. This paper focuses on fulfilling the anonymity function on the DSE system. In our EAS system, every node  $N$  chooses a proxy from its current and past neighbors via which the node may communicate with the rest of the world. Only node  $N$ 's neighbors know the association between  $N$ 's identification and its IP address.

## 2 System Architecture and Implementation

In this section, we first define the anonymity that we target for. Since the design of our anonymity mechanism is based on the DSE system, we briefly review the node clustering control mechanism of DSE, which is related to our anonymity mechanism. The remaining subsections discuss the design of the anonymity mechanism and its security analysis.

### 2.1 Definition and Rules

The word “anonymity” carries many different meanings. Until now there is no standard definition of anonymity, so we give our own definition of anonymity here to show our emphasis. For a P2P system  $S$ , there are thousands of nodes  $N_i$ , which are connected into a network. Each node has a

globally unique ID (called the node identification), which can be generated by a method such as the distributed hash table [4]. Each node also has a unique address, for example, the IP address together with the PORT number. We say that, for two nodes X and Y, X *recognizes* Y if X knows the association between the identification and the address of Y. We say a system is *absolute anonymous* if the following two conditions hold: (1) for each node X in an n-node system, there are at most  $O(1)$  nodes who recognize X, (2) no node can determine the sources of all messages. Message traffic among nodes in a P2P system leaves adversaries chances to corrupt the anonymity of nodes. In DSE, every message contains the identification of its originator. Adversaries may trace the address of the originator, or modify the message so that the address of the originator can be recognized easily by other nodes. In EAS, we say that a message is *anonymous* if no node, except the originator, can determine its source.

Anonymity usually conflicts with self-organization because self-organization needs to know the IP addresses of nodes in order to build a more efficient network. On the other hand, anonymity strives to hide the addresses. Our EAS system attempts to build an efficient communication while providing compromised anonymity. Before developing the mechanism, we explain some basic assumptions. All nodes, except the adversaries, will obey these assumptions all the time.

Assumption 1 : For nodes A and B, if A recognizes B, then there is or was a connection between them.

Assumption 2 : For nodes A and B, if A recognizes B, then B recognizes A.

Assumption 3 : For disconnected nodes A and B, A can not ask any other node to recognize B.

Assumption 4 : For connected nodes A and B, A would not corrupt the anonymity of any message that originates from B.

Assumption 1 ensures that a node can not know the identifications of other nodes except its current and past neighbors. Assumption 2 indicates that the recognition relationship is symmetric. Assumption 3 ensures that no node can ask any other node to recognize a node. Assumption 4 promises that no node would corrupt, intentionally or accidentally, the anonymity of messages that originate from its neighbors.

```

function PassiveNeighborUpdate() {
  // Assume this function is running on node S
  // NBRQ : the set of neighbors of node S,
  // RECID : the set of current and past neighbors, NBRQ ⊆ RECID
  for each node N in RECID {
    find node K such that K ∈ NBRQ and distance(K,N) ≤ distance(P,N)
    for any P ∈ RECID;
    inform nodes N and K to recognize each other;
  };
};

```

Figure 1: The passive neighbor update algorithm

## 2.2 Self-Organization with Anonymity

### 2.2.1 Distance between Nodes

The distance between nodes which was determined by the measurement model provides a reasonable physical meaning. However, the measurement model takes more efforts. Moreover, the measurement can be performed by only one of the paired nodes. Such a measurement model of distance limits the clustering function of NCC in an anonymous system. For this reason, EAS adopts an alternative definition, called the *global distance formula* (GDF), to estimate the distance. For two nodes A (IP address  $a_1.a_2.a_3.a_4$ ) and B (IP address  $b_1.b_2.b_3.b_4$ ), the distance between A and B is estimated as

$$\sum_{i=1}^4 |a_i - b_i| * K^{(4-i)}, \text{ where } K \text{ is a constant}$$

Obviously, this formula does not accurately reflect the physical distance between two nodes that are in different classes of the IP address scheme [5]. However, such a definition provides a meaningful distance for those nodes that are in the same autonomous system (AS)<sup>1</sup> [6][7]. The smaller the distance between two nodes is, the more possible they reside in the same AS. The distance that we defined, to some degree, represents the cost of transporting message between two peers of the connection. The GDF should be updated from time to time based on changing connection qualities among different ASs. The updated GDF will be spread to all nodes via message broadcast.

<sup>1</sup>The Internet is a collection of autonomous systems connected by routers. An autonomous system is a set of routers under a single technical administration that uses an interior gateway protocol and common metrics to route packets within the AS and an exterior gateway protocol to route packets to other AS's. If the messages travel across different autonomous systems frequently, the communication will be prohibitively expensive [13][14].

### 2.2.2 Passive Neighbor Update Method

We use third parties mechanism, called *passive neighbor update*, to help nodes to obtain their new neighbors. Let  $S$  be a neighbor of node  $N$  and  $RECID$  is the set of current and past neighbors of a node.  $S$  continuously locates another node  $M$  among the current and past neighbors of  $S$  that is closest to  $N$ .  $S$  then introduces  $N$  and  $M$  to connect to each other. Figure 1 shows the algorithm. As all nodes run the passive neighbor update algorithm cooperatively, the  $RECID$  in every node grows simultaneously.

Here we define the AVDS of the system is the average distance of a connection and the AVIS of the system is the average number of recognitions (current and past neighbors) per node. Consider the example in Figure 2. The passive neighbor update mechanism transforms graph A to graph B while the AVDS is reduced from 235,477,713 to 104,662,271. Now graph B contains two clusters:  $\{A, B, C, G, H\}$  and  $\{D, E, F\}$ , and each cluster contains closer nodes. We use the notation  $New(X, Y)$  by  $Z$  to denote a new recognition between  $X$  and  $Y$  by their common neighbor  $Z$ , and the notation  $Del(X, Y)$  by  $X$  to denote the deletion of the connection between  $X$  and  $Y$  by  $X$ . The process transforming graph A to graph B is:  $New(D, E)$  by  $F$ ,  $New(H, G)$  by  $F$ ,  $Del(H, F)$  by  $F$ ,  $New(B, H)$  by  $D$ ,  $New(C, H)$  by  $B$ , and  $Del(B, H)$  by  $B$ . Notice that this indirect recognition may not connect enough pairs of closest nodes but it provides better anonymity. The passive neighbor update algorithm reduces the AVDS and increases the AVIS gradually until all nodes connect to their closer neighbors.

Since third parties control the neighbors of a node, it is important that they must be trusted. It is not allowed that a node accepts a command from an arbitrary node to recognize another node. Every node can only accept the recognition command from its neighbors. For the example in Figure 2(a), nodes  $D$  and  $E$  can only accept a command from their neighbor  $F$  to recognize each other. However, node  $E$  can not accept the command from node  $B$  to recognize node  $A$  because node  $E$  does not recognize node  $B$ . The passive neighbor update mechanism works based on assumption 3 for anonymity.

### 2.2.3 Compromised Anonymity

A perfectly anonymous system forbids every node to recognize new nodes except its initial neighbors. However, it is almost impossible for such systems to perform self-organization. On the other hand, a system without anonymity allows every node to recognize all other nodes freely to get more advanced self-organization. Our system adopts a compromised strategy: it allows every

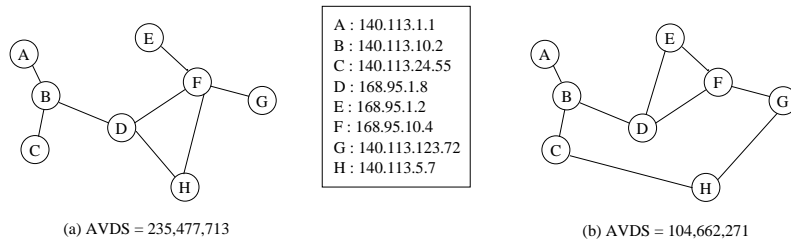


Figure 2: Process of new recognitions and deletions of connections

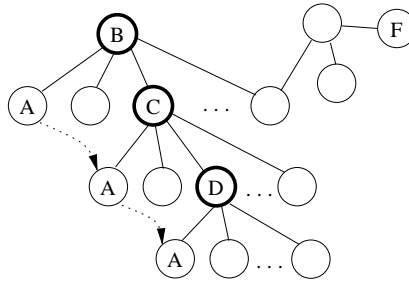


Figure 3: Passive neighbor update : moving path of a joining node A

node to recognize only a limited number of nodes. The value of AVIS determines the degree of anonymity of the system. From the results of our simulation, AVIS of a system maintained by the passive neighbor update algorithm is roughly  $O(\log_k n)$ , where  $n$  is the number of nodes and  $k$  is the maximal number of neighbors per node.

The passive neighbor update algorithm works similarly to the joining of nodes to an ordered, threaded  $k$ -ary tree. Consider the example in Figure 3 where a new node A joins the system. Node A connects to an arbitrary node B initially. Node B performs the passive neighbor update process to choose a node C that is closest to node A and then informs nodes A and C to connect to each other. Similarly, node C performs the passive neighbor update process to choose its neighbor D to connect to A. The update process continues until there is no new node for node A to connect to. Assume that every node contains at most  $k$  neighbors and the passive neighbor update mechanism continuously updates neighbors of every node. The number of increased recognitions for one joining node is roughly  $O(\log_k n)$ .

## 2.3 Communications Between Nodes

### 2.3.1 Message Delivery Proxy

For each node  $X$  in the EAS system, every current and past neighbor of  $X$  recognizes  $X$ . Therefore, node  $X$  could use any one of its current and past neighbors to be its proxy when another node  $Z$  wants to send a message to  $X$ . We call the neighbor through which a message is sent the *message delivery proxy (MDP)* of the node. Besides, every node has a table, named the *forward ID table (FIT)*, which will be explained later.

Consider the example in Figure 4. Assume there is a node  $X$  and its MDP is  $D$ . Before node  $X$  starts broadcasting messages,  $X$  generates a unique forward pointer, which is denoted as  $FID(X)$ . Every broadcast message  $M1$  originating from node  $X$  has a header that contains three pieces of information:  $FID(X)$ , the public key of  $X$ , and the address of  $D$ . Suppose the current  $FID(X)$  is  $FX1$ . After message  $M1$  is broadcast from node  $X$ , every node that receives  $M1$  will insert an entry  $(FX1, S)$  to its FIT, where  $S$  is the node which forwards the message. For example, nodes  $A$ ,  $B$ , and  $Y$  insert  $(FX1, X)$  into to their FIT's because node  $X$  forwards the message  $M1$  to them.

Assume that a node  $Z$  receives a broadcast message  $M1$  which originates from  $X$  and wants to send a reply message  $M2$  back to  $X$ . The body of  $M2$  is first encrypted with the public key of node  $X$ , and then the address of node  $D$  and  $FX1$  are attached in the encrypted message  $M2$ . Node  $Z$  then sends  $M2$  to it's MDP  $W$ . Once node  $W$  receives the encrypted  $M2$ , it reads and deletes the attached address of node  $D$  and forwards the resulting message  $M2a$  to node  $D$ . When node  $D$  receives  $M2a$ , it reads and deletes the attached  $FX1$  and forwards the resulting message  $M2b$  to  $Y$  (because an entry  $(FX1, Y)$  in the FIT of node  $D$  indicates  $M2b$  should be forwarded to  $Y$ ). Node  $Y$  receives message  $M2b$  and forwards it to node  $X$  in the similar way. Node  $X$  eventually receives message  $M2b$  and discovers that it originates from  $X$  (by checking the forward pointer). Hence  $M2b$  can be decrypted by  $X$ 's private key.

Notice that all nodes after MDP on the routing path of the one-to-one message will perform the confusing process to confuse traffic analysis attacks. In Figure 4, after nodes  $D$  and  $Y$  receive message  $M2a$  and  $M2b$ , respectively, they send fake traffic to some of their current neighbors. The fake traffic contains random-generated information which is unrelated to  $M2a$  or  $M2b$ . Moreover, the amount of fake traffic is in proportion to the HOPS of the message. The closer the message is forwarded to the destination, the more fake traffic is generated. With the confusing process, the discovery of the destination of a message by analyzing traffic become more difficult.

For reducing the overhead of checking the forward pointers in FIT, every forward pointer is

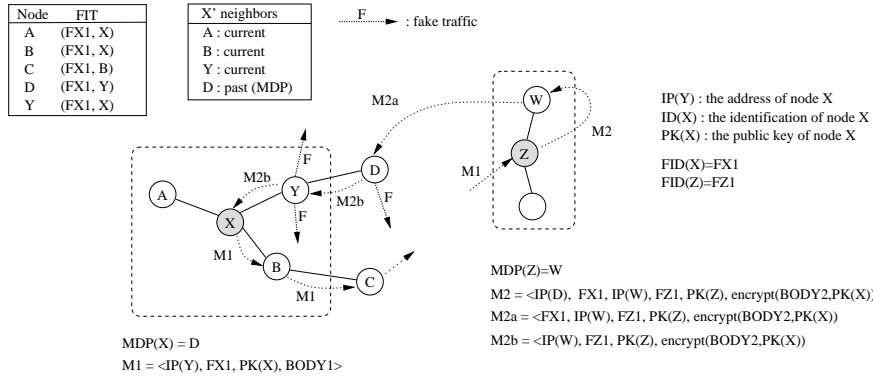


Figure 4: Message Delivery Proxy : Example

	one-to-all (broadcast)		one-to-one (reply)	
	sender	receiver	sender	receiver
local eavesdropper	exposed	beyond suspicion	exposed	beyond suspicion
collaborating nodes	beyond suspicion	beyond suspicion	absolute privacy	absolute privacy
sender	N/A	N/A	N/A	absolute privacy
receiver	absolute privacy	N/A	absolute privacy	N/A
sender.MDP	beyond suspicion	N/A	beyond suspicion	absolute privacy
receiver.MDP	N/A	N/A	absolute privacy	beyond suspicion

Table 1: Anonymity Properties of EAS

timestamped which is initialized by message originators. Expired forward pointers will be deleted from FIT.

On the other hand, every broadcast message has the HOPS and TTL fields, which records the number of nodes the message has travelled through and its upper limit. When the HOPS of a message exceeds the TTL, the message is discarded. Additionally, the initial HOPS of every message request is assigned to a random number so that any node, even the MDP of the originator, can not determine whether the message originates from the predecessor or merely forwarded from it by checking the HOPS.

### 2.3.2 Attacks Analysis

Our EAS provides the protection of addresses anonymity of nodes by using message delivery proxies to bridge the message deliveries. We analysis our mechanism by the research of Reiter and Rubin [8], and the result is listed in Table 1.

### 2.3.3 Efficiency Analysis

Compared to Freenet, the message delivery proxy of our EAS system brings about the following advantages:

- The anonymity and the efficiency of message delivery are adjustable. A users can balance the degree of anonymity and delivery efficiency depending on his own situation.
- A one-to-one message travels through at most two nodes. The load of message delivery is equally shared by the two ends of a message and their MDPs.
- No node will become the bottleneck because the MDP can be selected randomly by the message originator. It is almost impossible that many nodes share one MDP.
- Message delivery in EAS is much faster than the return path mechanism of the Freenet. The only overhead for our method is to check FITs of MDPs and transport some fake traffic between near neighbors. A possible solution to reducing the checking of MDPs in FITs is to timestamp every FIDs so that expired FID will be deleted from FIT.

## 3 Performance Analysis

We simulate the passive neighbor update mechanism and compare its performance with that of the original DSE system. The simulation system adds one node periodically up to 1100 nodes in total. We observe the AVIS, AVDS, HOPS under different situations.

### 3.1 Passive Neighbor Update

We compare AVDS and AVIS of the DSE system without (called A0) and with (called A1) the anonymity mechanism enabled.

The result in Figure 6 shows that AVDS in A0 is less than that in A1. However, the difference gradually diminishes when the number of nodes grows to 1000. On the other hand, AVIS under A0 grows roughly proportional to the total number of nodes. The reason that the AVIS value does not reach the exact number of nodes is that message broadcasts are not frequent enough to let every node recognize all other nodes in our simulation. The most important measurement, AVIS in A1 system, grows very slow with the number of nodes. The AVIS grows to only 36.34 when the total number of nodes reaches 1100. AVIS is approximately  $O(\log(n))$ , where  $n$  is the number of nodes.

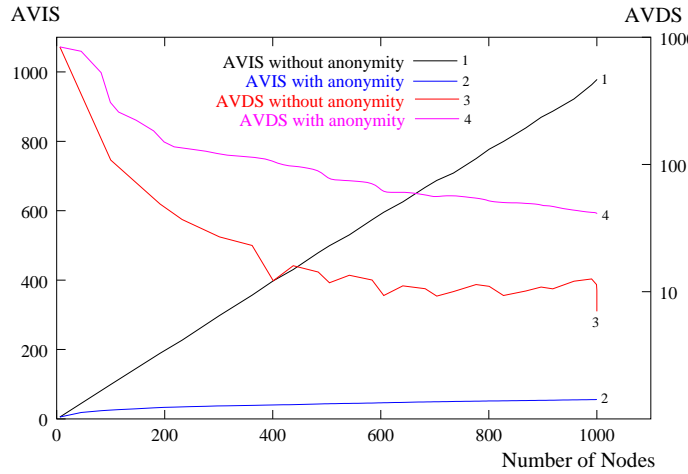


Figure 5: AVIS / AVDS v.s. the number of nodes

### 3.2 Performance under Failures

In contrast to systems with the small-world effect, such as Freenet and Gnutella, the EAS system provides better failure tolerance. As mentioned above, one major characteristic of a small-world graph is the power-law distribution of the number of links per node. If those nodes which provide shortcuts fail, the performance of requests will drop quickly. Therefore, there exists a critical point of the ratio of the failed nodes for small-world networks. If the ratio of failed nodes exceeds the point, the search function crashes.

We simulate random failures of nodes for our EAS system with 1000 nodes initially. We gradually remove more and more nodes selected at random from the system and watch the average path length of requests. The result in Figure 12 shows that our EAS system can tolerate more node failure because there is no obvious critical point of failure in the ratio of failed nodes.

## 4 Conclusion

Our EAS system provides the anonymity service which allows a node to recognize at most  $O(\log(n))$  nodes. The system is able to re-organized itself automatically in order to reduce cost of message delivery. By way of message delivery proxies, one-to-one messages are sent directly from the originators to receivers with few forwardings. The clustering effect of the system guarantees these forwardings with low bandwidth cost. The forwarding mechanism also provides options for users to balance the degree of anonymity and delivery efficiency depending on their own situation.

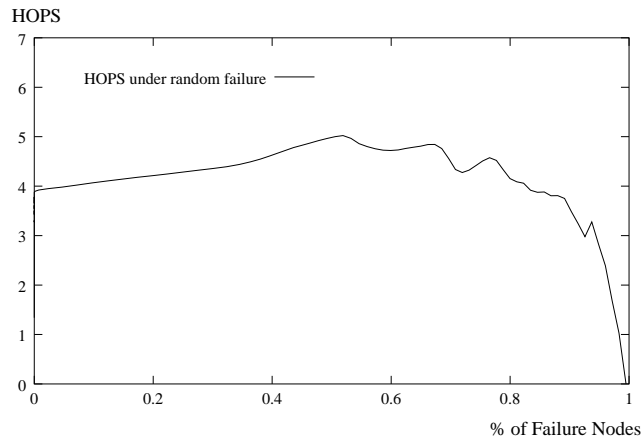


Figure 6: Change in request path length under random failure

## References

- [1] Andy Oram. Peer-to-peer harnessing the power of distributed technologies. O'Reilly 2001, pp. 94-122.
- [2] Ian Clarke, Oskar Sandberg, Brandon Wiley and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. Lecture Notes in Computer Science, vol. 2009, 2001, pp. 46-66.
- [3] Ching-Wei Huang and Wu-Yang. Self-organization for peer-to-peer systems, submitted for publication, 2003.
- [4] Michael J. Freedman and Robert Morris. Tarzan: a peer-to-peer anonymizing network layer. Proceedings of the 9th ACM Conference on Computer and Communications Security, November 2002.
- [5] RFC 1365. An IP Address Extension Proposal. <http://www.faqs.org/rfcs/rfc1365.html>.
- [6] Matei Ripeanu, Ian Foster and Adriana Iamnitchi. Mapping the Gnutella network: properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Computing Journal, vol. 6, 2002.
- [7] RFC 1772. Autonomous System. <http://www.faqs.org/rfcs/rfc1772.html>.
- [8] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for web transactions. ACM Transactions on Information and System Security, vol. 1, no. 1. 1998, pp. 66-92.