

## EOS: Evolutionary Overlay Service in Peer-to-Peer Systems

L.I. Zhijun and Liao Minghong  
School of Computer Science and Technology  
Harbin Institute of Technology, No. 318 HIT, Harbin, 150001, China

---

**Abstract:** The peer-to-peer (P2P) systems nowadays can be mainly classified into two categories: structured and structure-less systems based on their overlay organization. The structured systems can achieve determinate efficiency due to their rigorous structure with the cost of losing robustness and vice versa. We provide a semi-structured overlay based on the separation of routing structures and overlay organizations in this paper and the new overlay can achieve determinate efficiency with high robustness. Moreover, the performance of the existing overlay is determined by the initial design and the overlay can not evolve with the information collected. But the new overlay devised in this paper is evolutionary inherent and accompanied by evolving service (EOS), EOS can improve the performance with the running of the P2P systems. Finally, our evolutionary overlay structure is constructed on the basis of linear algebra. So, the EOS can be theoretically analyzed and the results indicate that EOS can work with preferable integrated performance. The experimental results gained on the simulative platform verify the performance of EOS further.

**Key words:** Evolutionary over lay service, peer-to-peer systems, structured, structure-less systems

---

### INTRODUCTION

In nowadays times, the P2P systems can mainly divide into two categories: structured P2P systems and unstructured P2P systems based the construction of its overlay.

For structured P2P systems, such as Chord<sup>[1]</sup> and CAN<sup>[2]</sup>, etc., the documents are stored determinately at peers based on the hash of the document and the ID of peers and the relations between of peers are determinate also based on their hash values, therefore, the overlay in structured P2P systems is also are called as distributed hash table (or DHT)<sup>[1,2]</sup>. If the overlay in practically running is same with its initial design from theoretical analysis, the system performance is optimal. So the focus in structured overlay is the maintenance of the overlay<sup>[3,4]</sup> when the system is running and the performance of the system is always under the theoretical level. On the other hand, the performance of the system may be very unstable and its robustness is low because of the dynamic characteristic of P2P systems<sup>[5]</sup>.

For unstructured P2P systems, such as Gnutella<sup>[6]</sup> and Morpheus<sup>[7]</sup>, etc. The unstructured overlay can be constructed with flexibility and robustness<sup>[6,7]</sup>, so they endow the excellent adaptability to the dynamic changes of P2P systems. But at the same time, their routing efficiency can't be guaranteed due to their uncertain inherent.

Consequently, a semi-structured overlay is devised in this paper. It is more robust and stable than the rigid DHTs and more efficient and controllable than

unstructured P2P systems as well. Moreover, the property to be extraordinarily noticed of our overlay is its evolution and in fact, it is just because of this property, our overlay can achieve better integrated performance than nowadays typical P2P systems. There are some researches about the evolutionary overlay in P2P systems. They are all different with our work more or less.

Considering the evolution of P2P overlays, there is another branch of researches, such as the work of David Liben-Nowell *et al.*<sup>[8]</sup> mainly focuses on the maintenance of Chord routing tables with nodes' join and leave and the work of Pandurangan *et al.*<sup>[9]</sup> studies the problem of maintaining an N-node P2P network as nodes join and depart according to a Poisson process. They are all focused on the design and analysis of the maintenance protocols under determinate DHTs to guarantee the system performance under the dynamic changes in structured P2P systems. Different from them, our research focuses on the evolutionary adjustment of overlay itself to improve the performance.

The work endowed with similar function as our paper is Tyson Condie *et al.*<sup>[10]</sup>. They provide a method to evolve the P2P overlay based on the trust management and their adaptive P2P topologies services can move the malicious peers and free-riders to the fringe. But there isn't an evolutionary structure formally defined in<sup>[10]</sup> and the adaptive property is only described qualitatively. Moreover, our EOS can evolve the overlay by considering not only the trustworthiness of peers, but also other properties such as their durability and their capability to filter inaccurate information.

---

**Corresponding Author:** L.I. Zhijun and Liao Minghong, School of Computer Science and Technology, Harbin Institute of Technology, #318 HIT, Harbin, 150001, China

Ying Zhu *et al.*<sup>[11]</sup> devise a distributed algorithm (oEvolve) which can evolve the overlay for the mapping from overlay to a physical network. The oEvolve implements the overlay's evolution based on the tree structure. Different from them, our work uses the mathematical evolutionary model to achieve similar results with more assurance.

Moreover, from the work of Paul Silvey *et al.*<sup>[12]</sup> and G. Pandurangan *et al.*<sup>[13]</sup>, etc., we can conclude that there are respectable researches on improving the system performance by adapting the P2P topologies. Comparing with them, our work has two remarkable differences: (1) EOS adapts the topology on an evolutionary overlay model not just through cutting the edges. (2) The final overlay after evolution is more deeply depicted by EOS than others.

**The description of EOS model**

**The architecture of EOS:** Figure 1 shows the architecture of EOS. EOS composes of three layers: P2P overlay; evolutionary mechanisms and applications. The kernel of EOS is the model of routing tables and related adjusting service.

**Linear model for the routing Tables in EOS:** The overlay in P2P systems is materialized as the routing table of the peers. In EOS, the structure of a routing table (for peer *p*) is represented as Fig. 2.

In Fig. 2, *k* is the number of *p*'s neighbors, *d* is the probe depth and *N<sub>ij</sub>* is a real number in [0,1] used for routing. From the form of Fig. 2, the model of EOS in this paper is also called linear model, the research of nonlinear model for EOS will be launched in futures.

Obviously, the key of this model is the definition of the *N<sub>ij</sub>* which defines the digest derived from information collected from the neighbors of a peer. Considering a P2P overlay shown in Fig. 3, for the peer *P*, its neighbors are *Q<sub>1</sub>*, *Q<sub>2</sub>* and *Q<sub>3</sub>*. Through *Q<sub>1</sub>*, *P* can reach the nodes *R<sub>1</sub>*, *R<sub>2</sub>* and *R<sub>3</sub>*, so the *N<sub>12</sub>* at *P* is the digest of *R<sub>1</sub>*, *R<sub>2</sub>* and *R<sub>3</sub>*, i.e. *N<sub>12</sub>*=Dig (*R<sub>1</sub>*, *R<sub>2</sub>*, *R<sub>3</sub>*). For *N<sub>11</sub>*, it should contain the information in the direction of *Q<sub>1</sub>*, which are *Q<sub>1</sub>*, *R<sub>1</sub>*, *R<sub>2</sub>* and *R<sub>3</sub>*. Now, *N<sub>11</sub>* = Dig (*Q<sub>1</sub>*, Dec (*N<sub>12</sub>*)), where Dec is a decline function to reflect the distance from *P*. Thus, in the routing table shown at Fig. 2, each row as a vector can represent digest the information over documents in some direction accompanied with the information about distance.

In our EOS, the digest of documents is defined as a vector *DV*={*d<sub>1</sub>*,*d<sub>2</sub>*,...,*d<sub>i</sub>*} where *d<sub>i</sub>* is a digest for document type *i*. It is needed to note that how to classify the documents is outside of the scope of this paper and is supposed existent directly here. For the overlay example of Fig. 3, we assume there are 3 types of documents denoted as {1,2,3}. They are distributed in the P2P overlay represented as the shadow boxes in Fig. 3.

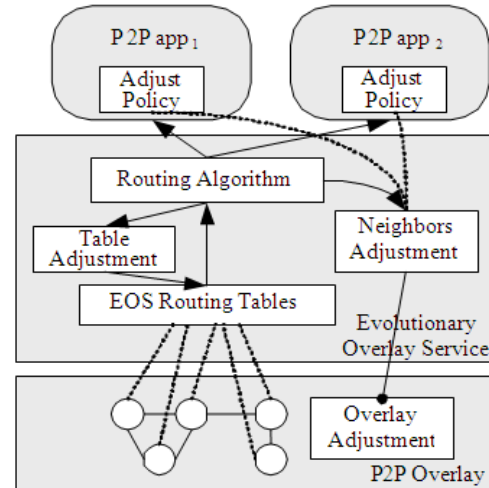


Fig. 1: The architecture of EOS

$$\begin{matrix}
 N_{11} & N_{12} & \dots & N_{1d} \\
 N_{21} & N_{22} & \dots & N_{2d} \\
 \dots & \dots & \dots & \dots \\
 N_{k1} & N_{k2} & \dots & N_{kd}
 \end{matrix}$$

Fig. 2: The structure of the *p*'s routing table in EOS

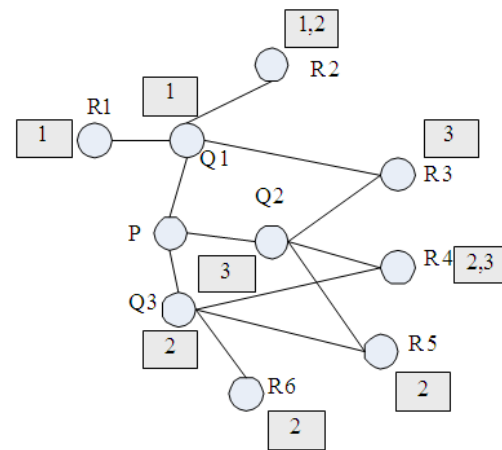


Fig. 3: A P2P overlay example

Different definition of the Dig can induce different effect and here two definitions (Dig\_e and Dig\_a) are devised for the object of efficiency and availability:

$$\text{Dig}_e(D_1, D_2, \dots, D_m) = \{\max\{d_{11}, \dots, d_{1t}\}, \dots, \max\{d_{m1}, \dots, d_{mt}\}\} \quad (1)$$

$$\text{Dig}_a(D_1, D_2, \dots, D_m) = \{\Sigma(d_{11}, \dots, d_{1t}), \dots, \Sigma(d_{m1}, \dots, d_{mt})\} \quad (2)$$

Similarly, different definition of the Dec can induce different effect and here we also define two declining modes: exponential decline (Dec\_e) and linear decline (Dec\_l).

Here, we adopt Dig\_e and Dec\_e to complete the illustration of the above example. Now,  $N_{12} = \text{Dig}(R_1, R_2, R_3) = \text{Dig}(\{1,0,0\}, \{1,1,0\}, \{0,0,1\}) = \{1,1,1\}$  and  $N_{11} = \text{Dig}(Q_1, \text{Dec}(N_{12})) = \{1, 1/2, 1/2\}$ .

**The routing algorithm in EOS:** In EOS, the query is also a vector generated by the peer willing to query the documents with some characteristics which are represented by the query vector (or  $QV$ ). The structure of  $QV$  is same with the structure of  $DV$ , i.e.  $QV = \{q_1, q_2, \dots, q_t\}$ , where  $Q_i$  is the weight occupied by the type  $I$  in  $QV$ .

The routine makes progress on the basis of the linear arithmetic of  $QV$  and  $N_{ij}$ . For each peer  $p$  (containing the peer launch the query) passed by the query  $QV$ ,  $p$  will find the documents completely satisfying the  $QV$  in its local storage, if the results can be found, the response is sent. Otherwise, the query will be ongoing by transmitting the  $QV$  to the next peer  $q$  that is selected according to the following formula:

$$q = \max\{i \in \text{Ng}(p) | (N_{i1} \cdot QV^T) / |N_{i1}| \cdot |QV|\} \quad (3)$$

where, the  $QV^T$  is the transpose of  $QV$  and obvious, the  $q$  is the direction which makes the cross-angle formed between it with the  $QV$  minimum.

**The evolution of the overlay in EOS:** The evolution of the overlay completely rests with the evolution of the routing tables. The  $N_{ij}$  is amended according to some rules to ensure the improvement of the system performance with the information collected.

**The maintenance of the routing tables:** Considering the stability and correctness of the  $N_{ij}$ s, we stipulate that the peer launching the query makes the modification. When a  $p$  launched a query  $QV$  receives a response  $Res$  for  $Qv$  and if the validity of  $Res$  has been verified,  $p$  then modifies  $N_{ij}$  by resolving the  $Res$  ( $Res$  records the path passed by  $QV$ ). Assume the peer initiating the  $Res$  is at the  $h^{\text{th}}$  hop of  $p$ , the  $N_{i1}, N_{i2}, \dots, N_{ih}$  will be modified where  $i$  is the direction selected by  $QV$ . When amending, the  $N_{ih} = \text{Dig}(Res, NIH)$  is computed firstly and then the similar adjusting process is carried through at  $N_{i1}, N_{i2}, \dots, N_{ih-2}$ .

**The self-adjustment of P2P's dynamic changes:** We discuss the adjustments of EOS considering the dynamic changes of P2P systems as four situations:

**New documents are shared:** In EOS, the documents newly shared are aware by other peers only by the responses of some queries.

---

**OverlayEvolution**

for a peer  $p$   
 $p$  has a array  $RN$  /\* $RN[i]$  record the number of routings for direction  $i$ \*/  
 when  $p$  select the next hop  $r$  by the routing formula  
 $RN[r] = RN[r] + 1$

When a Period Event comes  
 for each direction  $d$ , if  $(RN[d] < T_{r,d})$   
 drop the direction  $d$   
 clear the array  $RN$   
 if the number of neighbors is less than some bound  
 add some random peers as neighbors

---

Fig. 4: Overlay evolution algorithm in EOS

**Shared documents are deleted:** When the responses aren't received, we cannot assert that documents are deleted definitely. So, we deal with this situation by the scheme of  $N_{ij}$ 's periodical decline.

**New peers join:** The new joined peer  $p$  will be introduced into the system by a bootstrap peer  $q$ . Now, the  $p$  will become a new neighbor of  $q$  and a new direction ( $N_{n1} = \text{Dig}\{p\}$ ;  $N_{n2} = N_{n3} = \dots = N_{ni} = 0$ ) will be added in the routing table of  $q$ . At the same time,  $p$  builds its single direction of ( $N_{11} = \text{Dig}(q, \text{Dec}(N_{i1}$  at  $q))$ ;  $N_{12} = \text{Dig}(N_{i1}$  at  $q)$ ;  $\dots$ ;  $N_{1i} = \text{Dig}(N_{i1-1}$  at  $q))$ .

**Old peers leave:** The peer's leave is also disposed by the periodically descending of the  $N_{ij}$ s.

**The evolution of the overlay:** The evolution of the overlay is inherent in EOS and which is an outstanding characteristic comparing with other overlays. From the routing formula (3), we can see that the direction chosen by the routing algorithm is the direction with maximum information. Moreover, with the adjustment of  $N_{ij}$ s into consideration, the direction selected for routing is also the direction in which the digest information is more accurate with high probability. Based on the above analyses, we can easily devise an overlay evolution algorithm shown in Fig. 4 which will wash out the neighbors providing inaccurate information, or malicious peers and free riders.

**Theorem 1:** No matter what an initial overlay, there will be a stem come into being in the ultimate overlay after the evolution in EOS if the P2P is heterogeneous.

**Proof:** Firstly, we consider a random regular graph as the initial state of the overlay. Initially, all peers can be selected as the next hop with equal probability. With the ongoing of query and response, the  $N_{ij}$ s in the directions from which queries can't receive the response will be decreased due to the periodical decline. And some peers will be dropped from other peers' neighbors and the slope appears in the number of peers' neighbors.

When the slope appears, the peers with more neighbors (also with more digest information) will be selected with higher probability due to the formula (3) and its neighbors will be increased due to the washing out mechanism shown in Fig. 3. Now, the positive feedback is formed and the peers with more accurate information and more trustable will constitute into the stem of the overlay. The evolution for other overlays is obvious.

It is very interesting to notice that the overlay after the evolution via EOS is like with super-peer systems<sup>[14]</sup>.

**Formal analysis of EOS**

**The routing efficiency of EOS**

**Theorem 2:** If all information is completely correct in EOS, the number of average routing hops(AH) is less than  $O(\log_{E(k)}|P|)$ .

**Proof:** Considering any a peer  $p$  in EOS with arbitrary overlay structure, it can sniff the sharing information within the  $l$  bound by  $N_{ij}$ s. If all information is completely correct,  $p$  and successive peers all can choose the correct direction. Now, if one object for query is shared by the peers within the  $l$  bound, the routing hops must be less than  $l$ . Assume all neighbors aren't overlapping, the number of peers sniffed by  $p$  is  $k_1+k_2^2+...+k_l^l \approx (E(k)-E(k)^{l+1})/(E(k)-1)$ .

Obviously,  $(E(k)-E(k)^{l+1})/(E(k)-1) \geq |P| \rightarrow AH = \lceil l \rceil$  only if there is at least one object for query shared in the P2P systems. Thus,  $AH = \lceil \log_{E(k)}|P| \rceil$ .

From Theorem 2, we can see that EOS can achieve the equal performance with other classical DHTs<sup>[1-3]</sup> from the theoretical aspect.

**Theorem 3:** Let  $\gamma$  is the replica ratio of the document  $d$ , i.e.  $\gamma = |d|/|P|$  where  $|d|$  is the number of peers sharing  $d$ , now,  $AH = O(\log_{E(k)}1/\gamma)$ .

**Proof:** Its proof is same with the proof of Theorem 2.

The Theorem 3 is absolutely accurate if the replicas of  $d$  are dispersed at most possible in P2P systems. Comparing with other DHTs<sup>[1-3]</sup> which normally don't take replication mechanism in consideration, EOS can exploit sufficiently the replication mechanism because of  $\log_{E(k)}1/\gamma = \log_{E(k)}|P| - \log_{E(k)}|d|$ . In fact, in many realistic P2P systems, the number of replicas is proportional to the  $|P|$  and now  $AH$  is a constant.

Moreover, the heterogeneity of P2P overlay can make the realistic EOS improve the performance again because the query in EOS normally go along the stem of networks and intuitively, the  $AH$  along the stem will be less than the  $AH$  along the all networks.

**Theorem 4:** The  $AH = O(\log_{E(k)}|S|)$  when the stem is formed in EOS, where  $S$  is the set of peers in the stem.

**Proof:** When the stem is formed, the query will be routed mainly over the  $S$  by formula (3). The remaining calculation is same as the proof of Theorem 2.

If we assume that each peer in the stem can collect the information of  $c$  other peers, i.e.  $|S| = |P|/c$ , now,  $AH = \log_{E(k)}|P| - \log_{E(k)}|c|$ .  $AH$  can be reduced further.

**The robustness of EOS:** The analysis made in the above section is based on the assumption that all information of  $N_{ij}$ s is correct and updated. However, this assumption isn't reasonable in highly dynamic P2P systems<sup>[6]</sup>.

**Theorem 5:** If a document is pretended to share with a peer  $p$  in the EOS, there are

$$\sum_{d=1}^{d_T} [1 - (k^{d+1} - 1) \cdot \gamma / (k - 1)] \cdot k^d \quad (4)$$

peers will be misled, where  $d_T = \log_k((k - 1) / \gamma) - 1$ .

**Proof:** From the routing rules of EOS, it can be easily deduced that if a peer  $q$  with a distance  $d$  away from  $p$  is misled if and only if there aren't peers truly sharing the document within the bound  $d$  of  $q$ . Within the bound  $d$  of  $q$ , there are  $1 + k + k^2 + \dots + k^d = (k^{d+1} - 1) / (k - 1)$  peers and where  $(k^{d+1} - 1) / (k - 1) \cdot \gamma$  peers truly share the document. Obviously, if  $(k^{d+1} - 1) / (k - 1) \cdot \lambda \geq 1$ , then  $q$  can be misled by  $p$  with negligible probability. When  $d$  is small and satisfies  $(k^{d+1} - 1) / (k - 1) \cdot \gamma \leq 1$ , i.e.  $d \leq d_T$ ,  $q$  can be misled by  $p$  with probability  $1 - (k^{d+1} - 1) / (k - 1) \cdot \gamma$  and there are  $k^d$  peers like  $q$  at the distance  $d$  away from  $p$ .

To describe the stability of EOS clearly, the illustration of (4) is shown in Fig. 5.

From Fig. 5, we can see that when the  $k$  in EOS are chosen appropriately (normally large with a certain extent), with the larger replica ratio, the influence of the inaccurate information will become very little, on the other words, the stability and robustness of EOS can be improved with a large scale by the replication mechanism which is void in DHTs.

Moreover, the stem in EOS after the evolution can improve the stability of P2P systems more because the peers constituted into stem are the peers with more stability and with more accurate information. If the queries are routed mostly in the stem peers, the robustness of the system will be much larger.

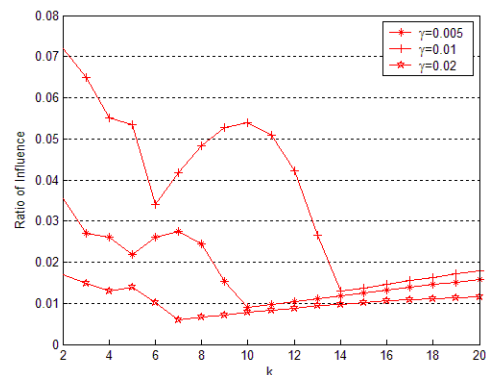


Fig. 5: The influence of inaccurate information in EOS

Table 1: The comparison of EOS, chord and gnutella

	Gnutella	Chord	EOS
Strength of the coupling in the overlay	None	High	Moderate
Efficiency ( $AH$ )	$O(1/\gamma)$	$O(\log P )$	$O(\log 1/\gamma)$
Robustness	Full	Low	$\approx$ Gnutella
Make use of replication mechanism	Full	Little	Full
The definitude of theoretical analysis	None	High	$\approx$ Chord

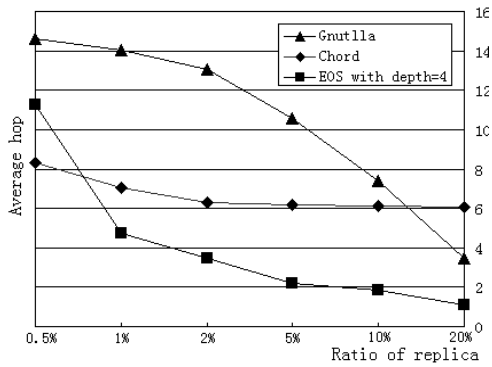


Fig. 6: The routing efficiency of EOS comparing with chord

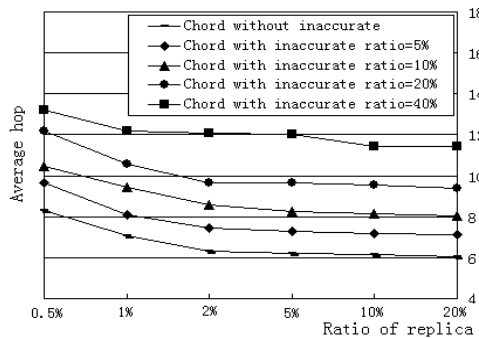


Fig. 7: The robustness of chord

**Model evaluation:** We will evaluate the EOS comprehensively over a simulative P2P file sharing platform.

**The routing efficiency of EOS:** Figure 6 illustrates the routing efficiency of EOS in a static P2P system with all peers are available. From the figure, it is obviously that EOS works with extraordinary performance comparing with Gnutella, even with Chord. The average hop of the query in Chord equals 6 approximately due to the formula  $\log_2(|P|)$  where  $|P|=1000$  even with high replica ratios (20%). Thus, the function of replication mechanism is unapparent in structured P2P systems. On the other hand, the Gnutella can adopt the replication sufficiently and with the increasing if replica ratio, the average hop decreases. However, comparing with EOS, EOS can not only achieve similar efficiency with Chord when the replica ratio is low, but also adopts the replication mechanism more deeply and the average hop in EOS may be very small with a moderate replica ratio ( $AH \approx 1$  with  $\gamma=20\%$  in above simulation).

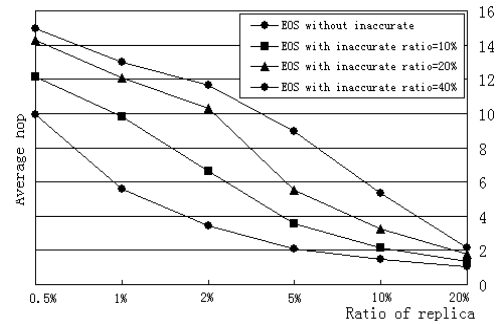


Fig. 8: The robustness of EOS

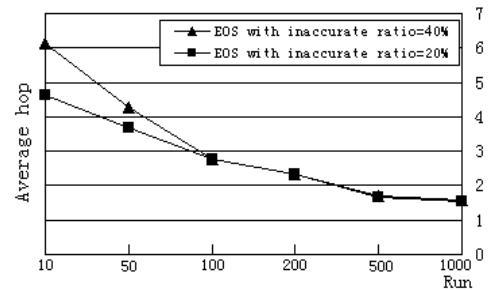


Fig. 9: The routing efficiency of EOS with evolution

**The robustness of EOS:** For comparison, Fig. 7 shows the robustness of Chord. We can see that the performance of Chord decreases equally with more inaccurate information. However, the average query hop of EOS will fall into a very small value with the increasing of replicas even when there is a great deal of inaccurate information in the system.

**The evolution of EOS:** From Fig. 9 we can see, the system performance is improved by the evolution of the overlay.

## CONCLUSION

Comparing with popular P2P systems nowadays, the contribution of EOS mainly contains two aspects:

1. EOS can assimilate the advantages of structured and unstructured systems summarized in Table 1.
2. EOS can evolve the overlays based on many factors, such as peers' reliability; peers' capacity; peers' ability to judge the correctness of the information, etc. The evolved overlay can improve the system performance, which is proved by theoretical analysis and verified by experimental results.

## ACKNOWLEDGEMENT

This work is supported by the HIT school fund (No: HIT 2002.74).

## REFERENCES

1. Stoica, I., R. Morris, D. Karger, M.F. Kaashoek and H. Balakrishnan, 2001. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. ACM SIGCOMM'01, SanDiego, CA.
2. Ratnasamy, S., P. Francis, M. Handley, R. Karp and S. Shenker, 2001. A scalable content-addressable network. In: Proc. ACM SIGCOMM'01.
3. Jagannathan, S. and G. Pandurangan, 2003. Stochastic analysis of a fault-tolerant and bandwidth-efficient P2P network. Technical Report CSD TR-03-029, Purdue University.
4. Mahajan, R. and M. Castro, 2003. Controlling the cost of reliability in peer-to-peer overlays. In IPTPS.
5. Krishna P. Gummadi and Richard J. Dunn, 2003. Measurement, modeling and analysis of a peer-to-peer file-sharing workload. Proc. Oper. Syst. Principles.
6. Gnutella. <http://gnutella.wego.com>.
7. Morpheus. <http://www.morpheus-os.com>.
8. David Liben-Nowell and Hari Balakrishnan, 2002. Analysis of the evolution of peer to peer systems. In: 21st ACM Symp. Principles of Distributed Computing (PODC), Monterey, CA.
9. Pandurangan, G. and P.E. Raghavan, 2001. Building low-diameter P2P networks. In Proc. FOCS.
10. Tyson Condie and Sepandar D. Kamvar, 2004. Adaptive Peer-To-Peer Topologies. In P2P 2004.
11. Ying Zhu, Jiang Guo and Baochun Li, 2004. Evolve: Towards evolutionary overlay topologies for high bandwidth data dissemination. IEEE J. Selected Areas in Communications, Special Issue on Quality of Service Delivery in Variable Topology Networks, 22: 7.
12. Paul Silvey and Laurie Hurwitz, 2004. Adapting peer-to-peer topologies to improve system performance. In Proc. 37th Hawaii Intl. Conf. System Sci., 2004.
13. Pandurangan, G., P. Raghavan and E. Upfal, 2003. Building low-diameter peer-to-peer networks. IEEE J. Selected Areas in Communications, 21: 6.
14. Yang, B. and G. Hector, 2003. Designing a super-peer network. Proc. 19th Intl. Conf. Data Engineering. IEEE.