

A Novel Mechanism to Construct a Compatible Overlay on Heterogeneous Mobile Peers

Chuan-Chi Lai
Comp. Sci. and Info. Eng.
National Taipei Univ. of Tech.
Taipei 106, TAIWAN
Email: gbs0856@gmail.com

Chuan-Ming Liu
Comp. Sci. and Info. Eng.
National Taipei Univ. of Tech.
Taipei 106, TAIWAN
Email: cmliu@ntut.edu.tw

Ying-Chi Su
Comp. Sci. and Info. Eng.
National Taipei Univ. of Tech.
Taipei 106, TAIWAN
Email: billandcs@gmail.com

Abstract—As the mobile computing environment emerges, people can use different mobile devices to access information ubiquitously. This has shifted the researchers' sights to Mobile P2P (MP2P) systems, where the membership among peers is more dynamic and ad hoc. In the ubiquitous environments, the mobile devices generally are heterogeneous. In this paper, we consider the MP2P systems where the mobile peers are heterogeneous and propose a mechanism, named *Heterogeneity-aware Overlay Technique (HOT)*, to build up an unstructured overlay. The proposed mechanism considers the overall ability, including power, network, CPU, and memory to set up the overlay. An ability equation is thus defined and the resulting overlay can ensure a good quality of service while the topology mismatching problem is alleviated. Last, we present our simulation for the proposed mechanism and compare the results with other related approaches. The simulation result shows that our proposed mechanism can effectively achieve a better service quality and mitigate the topology mismatching problem.

Keywords—MP2P, heterogeneous, ubiquitous, topology mismatching

I. INTRODUCTION

In recent years, Peer-to-peer (P2P) systems have received a great attention from both research and commerce [2], [3], [9], [14]. Related applications using P2P systems include content sharing, streaming video, VoIP, and even game streaming. Such an architecture has the advantages of decentralization, self-organization, and service availability. Each peer in the system can directly communicate and share resources with other peers. As the modern technologies in communications, networking, information management, and position systems advance, people can use the mobile devices to access information ubiquitously. This has shifted the researchers' sights to Mobile P2P (MP2P) systems, where the membership among peers is more dynamic and ad hoc. Besides, in the ubiquitous environments, the mobile devices generally are heterogeneous. In other words, the mobile devices in a MP2P system may have different abilities in power, memory size, CPU, etc. In this paper, we consider the MP2P systems where the peers are heterogeneous.

Some of the previous works on P2P system [12], [13] for managing peers focused on the *overlay network*, which is

an abstract and logical network built on top of a *physical network*. According to the way to organize peers in overlay, P2P networks can be divided into two categories: *structured P2P* and *unstructured P2P*. All the nodes in structured P2P are organized into a pre-designed structure, such as Chord [12], Pastry [10], CAN [11], Tapestry [15], and Kademlia [8]. In contrast, the nodes in unstructured P2P are organized in a free manner [1].

The topology of overlay network can be much different from the the topology of physical network; thus resulting in the *topology mismatch problem* as shown in figure 1. Such

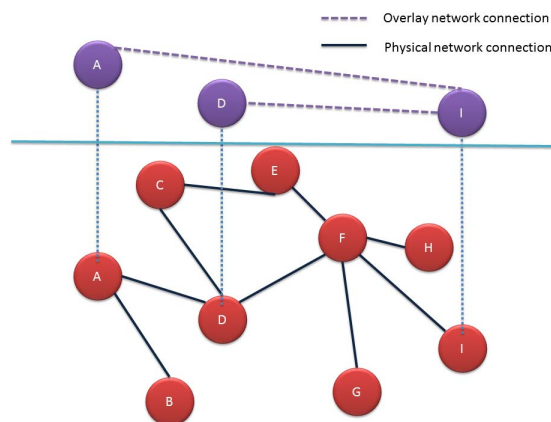


Figure 1. A Scenario for Topology Mismatch where the solid lines represent the physical network and the dashed lines are the overlay network

a phenomenon may deteriorate the performance of a P2P system and can be even worse in an MP2P system, where the mobility of peers is more dynamic. Some techniques [4], [5], [6], [7] have been proposed to alleviate the impacts of the topology mismatch problem. However, most of the existing methods to overcome the topology mismatch problem assume the mobile devices are homogeneous. Besides, we observe that, according to the way to organize peers, it is more promising to reduce the impacts of the topology mismatch problem by building an unstructured P2P system.

We thus propose a mechanism, named *Heterogeneity-aware Overlay Technique* (HOT), to build up an unstructured overlay consisting of heterogeneous mobile peers. According to the simulation results, our proposed mechanism can achieve a good performance on energy consumption and delay time, while alleviate the topology mismatch problem.

The rest of this paper is organized as follows. We introduce some related work in Section II and the terminologies we used in Section III. Section IV presents our proposed protocols and their features. In Section V, we discuss our simulation results. Section VI concludes this paper.

II. RELATED WORK

As mentioned, in our work, the constructed P2P overlay belongs to unstructured P2P. In unstructured P2P systems, the most famous example is Gnutella[1]. Gnutella is a decentralized P2P network. Each peer can be either a server or client. In some circumstances, a single peer can play two roles at the same time. Such a system thus does not have a central server. Like other unstructured P2P systems, Gnutella uses message flooding to locate resources. Since using message flooding may cause so-called broadcast storm problem, the query message for locating resources usually uses *time-to-live* strategy to limit the flooding range. However, Gnutella does not consider the ability of peer and topology mismatch problem when flooding messages; thus, being unscalable and wasting lots of resources in the system and network.

When a P2P system suffers from topology mismatch, redundant messages will occur during peer communication and cause large amount of unnecessary traffic in the network. Let us use Figure 1 as an example. When node *A* sends a message to *D* via *I* on the overlay, the corresponding path in the physical network is (*A-D-F-I-F-D*), which is not as same as the path in overlay. To avoid unnecessary peers in the path, the best path for *A* to *D* on the overlay would be exactly as same as the one on physical network. To achieve this overall, the optimal scenario happens when the overlay topology and physical overlay are exact the same.

The topology mismatch problem can be modeled by using graphs [5]. In [5], the author also shows that given a physical topology, even if the global knowledge is known, it is NP-hard to find an optimal overlay network where the query traffic cost and average response time are minimized. Furthermore, due to the dynamic property of P2P systems, finding the best overlay topology will cost a lot of resources and time. So, simple and efficient heuristics for finding a proper overlay that fits continuously changing peers and alleviates the impact of topology mismatch problem are expected.

In[6], the authors provide a location-aware topology matching technique(LTM) to construct the overlay. The basic idea of the LTM is simple. Each peer detects the connectivity with other peers in a limited range, and then uses the

results to build up the overlay by comparing the possible paths to all other peers within the given range. To detect the connectivity, LTM uses limited flooding strategy. There is a trade-off between the detecting range and topology mismatch. Since LTM still uses flooding for connectivity detection, although the flooding range is limited, an amount of messages can be expected.

Scalable bipartite overlay, SBO, is proposed in [7]. SBO uses a role-play strategy. It separates the peers into two roles in the overlay network. The red and white peers carry different duty. For the white one, it has to fetch the information about delay time within 2 hops range. Then it sends the delay information to the red peers. The red peer is responsible for layout the MST(minimum spanning tree) for the overlay. SBO also has the way to alleviate the topology mismatch problems by further refining the overlay topology.

Both LTM and SBO are proposed to build up the overlay which ease the message duplication situation, thus alleviating the impact of topology mismatch. However, both of them did not take peer's ability into consideration when building up the overlay. For example, suppose a peer can only connect to a limited number of neighbors. In some circumstances, a peer can connect to more neighbors than the limit. In this case, the ability of that peer may not afford; thus, degenerating the quality of service.

III. PRELIMINARIES

Our method is proposed for heterogeneous peers. We use some specific parameters, such as power and number of neighbors, as peer's ability to build the overlay. The generated overlay can have a better performance and be more practical. In the meanwhile, the mechanism also mitigates the topology mismatch problem.

To evaluate the performance of generated P2P system and the impact of the topology mismatch problem, the following parameters are usually considered.

- **Latency:** The latency is the time or hop counts between issuing a query and termination of a query. The parameter can reflect the different structure of overlay network.
- **Average Delay Time:** Suppose peer p_i has k neighbors, $p_{ij}, j = 1, \dots, k$, in the overlay network. Let the delay time between p_i and some neighbor $p_{ij}, 1 \leq j \leq k$ is D_{ij} . Then the average delay time, D_{avg_i} , for peer p_i is $D_{avg_i} = \frac{\sum_{j=1}^k D_{ij}}{k}$. This parameter can be used as a topology mismatch measurement.
- **Path Length Ratio:** For a success query, we take the path length of overlay P_o divided by path length of the physical network P_p . The result is the ratio between overlay and physical layer $PR = \frac{P_o}{P_p}$. The ideal ratio is 1 and the ratio can give us an idea of topology mismatch degree.
- **Message Cost:** During building and maintaining the overlay, peers need to send or receive control messages.

The message cost denotes the total amount of the messages for building and maintaining the overlay. The throughput and response time of the system may suffer from the overhead of messages sending and receiving. It can be one of the parameters for measuring the performance of the network system.

In the proposed mechanism for constructing the overlay among heterogeneous mobile peers, we consider four primary parameters when selecting peers. The objective is to have the resulting system achieve better performance in terms of average delay, low message count, and less overhead. At the same time, the topology mismatching problem is mitigated. The considered parameters include CPU, memory, network speed, and power. Some of the applications will use a lot of CPU power to achieve smooth operating experience. However, this factor effects less for the performance. Sufficient memory indicates a high-end mobile device which can take more responsibility if it has good networking and adequate power remaining. Networking speed and power are the most important issue when peers operate in MP2P. The performances of most applications rely on these two factors. In the following, we define the ability of a peer with the above four parameters.

Suppose C , M , N , and P are the parameters for CPU, memory, network speed, and power, respectively. We define the ability ab_i of peer p_i as

$$ab_i = (N + (C^\alpha + M^\beta)) \times P, \quad (1)$$

where α and β are two variables ranging from 0 to 1. From our observations, CPU and memory are not as important as power and network speed when computing the ability, so their contributions to the ability is not much. On the other hand, power is the most critical parameter for judging the mobile device's stability. So, we set P to time the other parameters in the equation. All the four parameters are scaled from 1 to 10.

IV. PROPOSED SOLUTIONS

In the proposed mechanism, we adapt the method of [6] to take the abilities of heterogeneous peers into consideration. We employ the social behavior and set all peers as two roles, *employer* and *employee*, for constructing overlay. To maintain the overlay, the role remains for each peer. In the constructed overlay, we allow each peer to have at most five two-hop neighbors to avoid unnecessary connections and reduce the amount of network traffic.

To set up the overlay, a employer first evaluates the necessary ability of the employee for running the specified application. Then, the employer seeks for appropriate employees which have adequate ability requested by the employer. There are two main operations for employer. One is to hire the employees by sending a TTL-2 message. The TTL-2 message is a request for two-hop neighbors which

fulfill the ability requirement. The other one is to replace the inadequate employees by better ones.

During the executions of system, the employees continue searching better employers. Once a employee receives a TTL-2 message from employers, it compares all the employers, including the old ones, and select the best employer to join. Now we give more details about the proposed mechanism, including overlay formation and maintenance.

Initially, there is no employment relation among peers like the example in Figure 2(a), where the circled numbers indicate the ability values of the peers. To form the overlay,

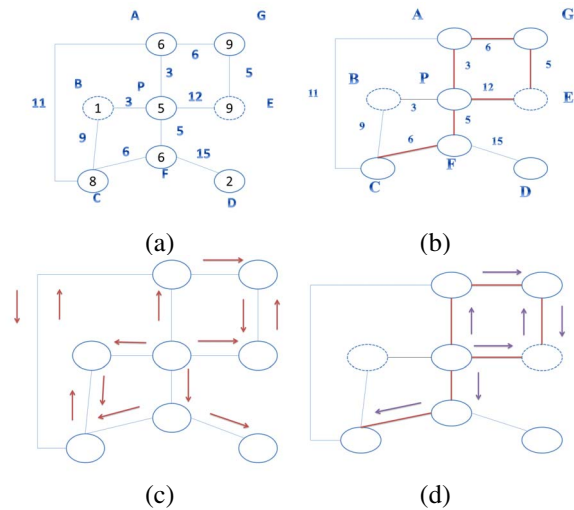


Figure 2. An example for overlay formation, where (a) no employment relation among peers initially, (b) peer P sending a TTL-2 message, (c) peers E and B reply the employer status, (d) connection setting; the dashed circles representing the employers

a peer P first sends a TTL-2 message to decide if it is a employer or employee as shown is Figure 2(b). After collecting the reply messages from its two-hop neighbors, P will set up the overlay with appropriate peers. Figure 2(d) shows an example where the connection (bold line) is built by peer P . All the other peers perform the same process and there are two main steps in this process.

Step 1: (Role Selection)

A peer sends TTL-2 message first to detect if an employer exists among its two-hop neighbors. If no, it will be an employer; otherwise, an employee. As Figure 2(c) shows, peer P is an employee because of employer E and B .

Step 2: (Connection Setup)

In this step, the peer will start to set the connections with others based on their ability values. In Figure 2(d), peer P has set peer A , E , F and C as its neighbors. Peer D is ignored due to inefficient ability. In particular, E was connected even with a larger delay because of its high ability value. The solid lines indicate the connections.

We present the pseudo-code for a peer to join the system in $PEER_JOIN(P_i)$. This is the major function called in the overlay formation phase.

Algorithm: $PEER_JOIN(P_i)$

```
// Join process in overlay formation
Input:  $P_i$  is set as an employee
// TTL-2: Time-to-live in 2 hops
1  $P_i$  sends a TTL-2 message to its neighbors
2 if  $P_i$  finds no other employers in its two-hop neighbors
3    $P_i$  is set to be an employer
4 else
5   for each two-hop neighbor  $v$  of  $P_i$ 
6     if the ability of  $P_i < v$ 's ability
7       make a connection between  $P_i$  and  $v$ 
```

After the overlay formation, the following is the overlay maintenance phase. Only employers will do the maintenance. Once an employer receives TTL-2 messages from the others, it will judge the ability value and decide to connect with it or not. By doing so, unlike the other mechanisms, HOT can fully use TTL-2 messages and decide better connections; thus being able to have a better overlay topology. $MAINTENANCE(P_i, m)$ shows the pseudo-code for maintenance.

Algorithm: $MAINTENANCE(P_i, m)$

```
// System maintenance of an employer
Input: Employer  $P_i$  and a received TTL-2 message  $m$ 
form other peer  $P_j$ 
// Note: the number of employees is limited
1 if  $m$  is not NIL
2   if  $P_j$ 's ability is better than one of  $P_i$ 's employees
3      $P_i$  sends a message asking for connection to  $P_j$ 
4     if  $P_i$  receives the join message from the  $P_j$ 
5       the employee with worst ability
         is replaced by  $P_j$ 
```

V. SIMULATION RESULTS

In this section, we present and discuss our simulation results. For the simulation environment, we consider 100 to 700 mobile nodes in a $1000 \times 1000 m^2$ area. Initially, they are distributed uniformly. Each node has a speed of $1 m/s$ using random waypoint mobility model. The transmission range is $30m$. We define the ability of a peer according to Equation 1. All the values of parameters, $N, C, MandP$, vary from 1 to 10 and we set α and β to be 0.3, respectively.

In the simulation, we have the following measurements: *average delay*, *average number of neighbors*, *message cost*, *overload rate*. Recall that, each peer has an ideal ability for its connected peers, say ab_{ideal} and each of its connected peers has an actually ability. Let ab_{ave} be the average actually ability of the connected peers. The overload rate is the ratio of ab_{ideal} to ab_{ave} and the value of it is the smaller the better. This measure indicates the guarantee on the better

service. We will compare our proposed mechanism, HOT, with other existing methods, including Gnutella, LTM, and SBO.

A. Delay Time

We first discuss our simulation results on delay time. Fig 3 shows the average delay time for all the methods. Gnutella has a greater tendency on delay time as the number of peers grows. In Gnutella, peers join the overlay randomly and no filtering process is involved in topology construction. When the number of peers increases, the density of peers grows and thus the delay becomes more serious. For LTM, SBO, and HOT, the delay time decreases when the number of peers grows because they take delay time as a metric during topology formation. As the density of peers increases, a better overlay can be formed and the actual transmission range reduce; thus, leading to a shorter delay time.

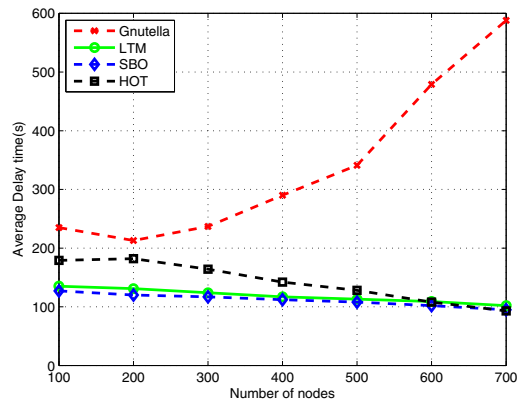


Figure 3. Average delay time between 1 hop neighbors

B. Average number of neighbors

By watching the average number of neighbors, we can figure out how robust the topology formation is. For a peer, more neighbors means the connectivity is stronger. Since there is no filtering process involved in topology construction in Gnutella, it has higher average number of neighbors for each peers. More conditions for topology formation will result in a smaller average number of neighbors. Since HOT considers more parameters for topology formation, the average number of neighbors is the least as shown in Figure 4. However, more neighbors may lead to a larger message cost due to higher connectivity since unstructured P2P usually use flooding to send messages.

C. Message cost

Figure 5 shows the message cost for each method. Among them, Gnutella has the largest message cost than any other methods because the duplication of messages in Guntella happens most. For SBO, even with different techniques for

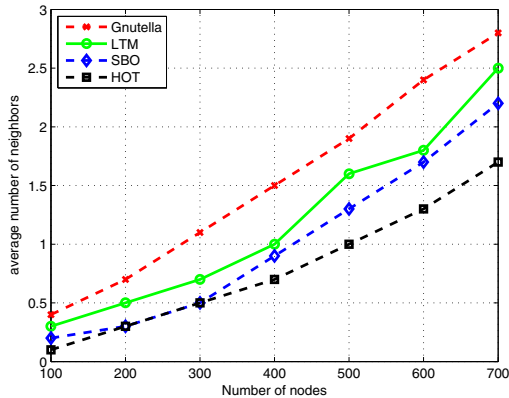


Figure 4. Average number of neighbors for Gnutella, LTM, SBO, and HOT

construction, the process for alleviating topology mismatching problem consumes an amount of messages. Thus, the message cost for SBO won't be lower than other significantly. Our method, HOT, which is adapted from LTM, utilizes TTL-2 messages for maintenance. So, we can still keep low message cost during the topology formation. From Figure 5, we can see that the message costs for SBO, LTM, and HOT are close.

D. Overload rate

Recall that the overload rate indicates the guarantee on the better service. Fig 6 indicates that HOT is the best candidate to provide the best QOS than any other methods.

When there are only few peers, the peers on the network are scatter sparsely. Once the number of peers reaches to a sufficient amount, the overload rate become stabilized. Furthermore, Gnutella, LTM and SBO do not consider the overall ability when constructing the topology. The chance

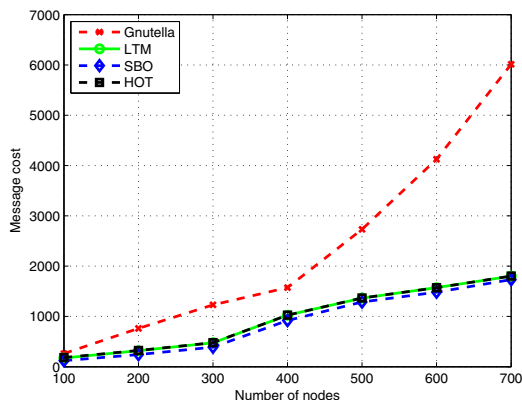


Figure 5. Message cost for Gnutella, LTM, SBO, and HOT

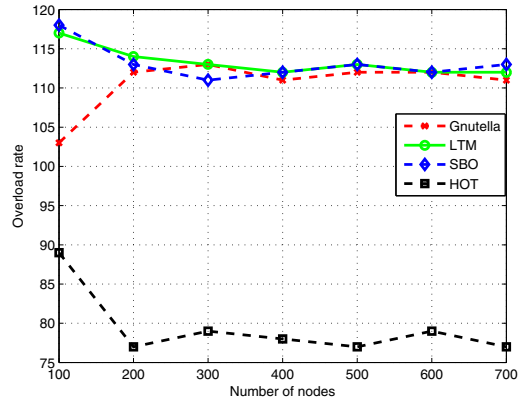


Figure 6. Overload rate for Gnutella, LTM, SBO, and HOT

for providing good service quality become less. With the help of sufficient neighbors having higher ability, the services, such as digital content streaming, can have much better quality than others which do not take the ability into consideration. The overload for SBO, LTM and Gnutella is basically similar because they focus on the connectivity when formatting the overlay .

VI. CONCLUSIONS

In this paper, we provide a mechanism, Heterogeneity-aware Overlay (HOT), to construct an unstructured overlay system consisting of heterogeneous mobile peers. The proposed mechanism considers the overall ability, including power, network, CPU, and memory to set up the overlay. An ability equation is thus defined. By considering all the aspects about peers, the resulting overlay thus can ensure a good quality of service and the topology mismatching problem is alleviated.

The simulation results also show that HOT is effective for constructing the overlay. Although a little more delay time and message count is paid, the quality of service indicated by overload measurement can be enhanced. In the future, more analysis and experiments will be performed for validating the ability equation and we plan to explore how much HOT can improve the topology mismatching problem and evaluate some practical applications using the overlays constructed by HOT.

REFERENCES

- [1] Gnutella website. <http://gnutella.wego.com>(viewed on November 2009).
- [2] H.-K. Chiang, H.-W. Chen, and F.-L. Kuo. Locality support for mobile p2p network. In *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing*, pages 517–522, New York, NY, USA, 2007. ACM.

- [3] P. Flocchini, A. Nayak, and M. Xie. Enhancing peer-to-peer systems through redundancy. *IEEE Journal on Selected Areas in Communications*, 25(1):15–24, Jan. 2007.
- [4] M.-Z. Lin, Y.-C. Su, C.-H. Chen, S.-M. Hsu, and C.-M. Liu. Solving the topology mismatching problem on mobile p2p systems. In *Proceedings of the 15th Mobile Computing Workshop*, 2010.
- [5] Y. Liu. A two-hop solution to solving topology mismatch. *Parallel and Distributed Systems, IEEE Transactions on*, 19(11):1591–1600, nov. 2008.
- [6] Y. Liu, L. Xiao, X. Liu, L. M. Ni, and X. Zhang. Location awareness in unstructured peer-to-peer systems. *IEEE Transactions on Parallel and Distributed Systems*, 16:163–174, 2005.
- [7] Y. Liu, L. Xiao, and . M. Ni. Building a scalable bipartite p2p overlay network. In *Proceedings of the 2004 IEEE International Conference on Parallel and Distributed Systems*, pages 1296–1306, 2004.
- [8] P. Maymounkov and D. Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *Proceedings of the First International Workshop on Peer-to-Peer Systems*, pages 53–65, 2002.
- [9] J. Risson and T. Moors. Survey of research towards robust peer-to-peer networks: search methods. *Computer Networks*, 50(17):3485–3521, 2006.
- [10] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350, 2001.
- [11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. *SIGCOMM Computer Communication Review.*, 31(4):161–172, 2001.
- [12] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, Feb 2003.
- [13] Y.-C. Wu, C.-M. Liu, and J.-H. Wang. Enhancing the performance of locating data in chord-based p2p systems. In *Proceedings of the 2008 14th IEEE International Conference on Parallel and Distributed Systems*, pages 841–846, 2008.
- [14] Z. Xu, X. He, and L. Bhuyan. Efficient file sharing strategy in dht based p2p systems. In *Proceedings of the 24th IEEE International Conference on Performance, Computing, and Communications*, pages 151–158, 2005.
- [15] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, Jan. 2004.