The Design of Topology-Aware Overlay Networks for Ubiquitous Applications

Chuan-Feng Chiu, Steen J. Hsu, Sen-Ren Jan, Shao-Min Lyu
Department of Information Management
Minghsin University of Science and Technology
E-mail : cfchiu@must.edu.tw

Abstract

In recent year, ubiquitous computing has gained more attention in academic area. With ubiquitous computing, users can process or get information anytime and anywhere. Fortunately, with the advent of peer-to-peer overlay network, the ubiquitous environment can become scalable with using the peer-to-peer overlay network as the communication middleware. Peer-to-Peer overlay network is an application model without considering underlying network topology. But there exists mis-match problem between peer-to-peer overlay network and physical network topology. This cause inefficient transmission or routing between peers in the peer-to-peer overlay network. On the other hand, the situation will have serious delay in real-time service, for example streaming service. Therefore, in this paper we propose an improvement mechanism based on physical network hop information to reduce the transmission cycles to adjust the arrangement of peer-to-peer overlay network dynamically.

1. Introduction

In recent year, ubiquitous computing has gained more attention in academic area. With ubiquitous computing, users can process or get information anytime and anywhere. The famous applications include ubiquitous home or ubiquitous care etc.. All devices ubiquitous environment are connecting and do not need to know about each device. In such architecture, ubiquitous computing must be scalable. Fortunately, with the advent of peer-to-peer overlay network, the ubiquitous environment can become scalable with using the peer-to-peer overlay network as the communication middleware.

Peer-to-peer overlay network is a popular application model to provide several applications including file-sharing, load balance etc.. Peer-to-peer overlay network is a distributed and virtual architecture without considering underlying network topology. Gnutella[6], Freenet[7], Naspter[5], CAN[1], Chord[2], Pastry[3], Tapestry[4] are the famous peer-to-peer overlay network architecture to realize these usage. Because the hosts in the peer-to-peer overlay network are connected in a virtual way, the data transferring do not consider the efficient usage of underlying network topology. When a host transmits a data to another host, the transmission path will be according to overlay network topology. Therefore, the nearby host in peer-to-peer overlay network might have long distance in the underlying network so that the efficiency of transferring might be in worse situation. This problem is called mismatch problem in peer-to-peer overlay network. Now we can consider the overlay network and underlying network topology relationship in Figure 1. When host A transfer a data piece to host B according to the overlay network topology in Figure 1. The actual data transferring path is A→C→D→B according to the underlying network in Figure 1. The actual transmission path is not reflected with the better topology in overlay network. We can consider another scenario according to the Figure 1. When host A send a data piece to host D, the data will be transferred from host A to host D via host B. But the actual transmission path will be A→C→D→B→D, the path between host B and host D are visited twice and cause the redundant messages to degrade the network performance and overall throughput.

978-1-4244-5228-6/09/$26.00 c2009 IEEE

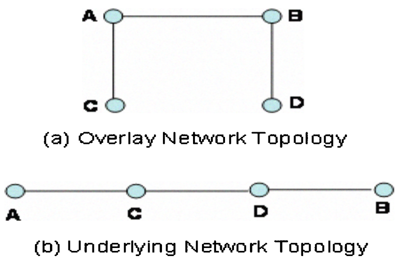

Figure 1 : Overlay Network and Underlying Network Topology

The rest of this paper is organized as followings. Section 2 will reveal peer-to-peer overlay network concept, and related research on mismatch problem. Section 3 describes the design of improvement mechanism to make topology-aware peer-to-peer overlay network. Section 4 describes the system architecture of the proposed implementation. Finally, we give a brief conclusion of our work in section 5.

2. Related work

2.1. Peer-to-Peer Overlay Network

In this section, we will investigation the literatures related with the peer-to-peer overlay network and mismatch problem in overlay network.

In general, peer-to-peer overlay network can be divided into two basic categories including unstructured peer-to-peer overlay network and structured peer-to-peer overlay network. The famous unstructured peer-to-peer overlay network includes Gnutella[6], FreeNet[7], Fasttrack[8]/KaZaA[9], BitTorrent[10] etc.. The hosts in these overlay network are connected in a random and distributed way. In order to query the data pieces in overlay network, these overlay network use flooding, random walks or expanding-rich TTL approaches to query data pieces. The query message is sent to other peers to search the needed data pieces and the query message is also forwarded by these intermediate hosts to enhance the query scope. But the problem that message will be sent and reply in a long period will be produced. Therefore, another architecture Fasttrack[8]/KaZaA[9] is proposed to enhance the query performance. In this architecture, a super node is proposed to index the data pieces and the nodes hold the data pieces belong to the super node. The hosts in overlay network are organized as the hierarchy architecture.

Another kind of peer-to-peer overlay network is structured peer-to-peer overlay network. The famous are CAN[1], Chord[2], Pastry[3] and Tapestry[4]. These architectures are based on Distributed Hash Table(DHT) mechanism to allocate the hosts in the overlay network. DHT assign *key* to the data and compute a *value* for the *key*. The *(key, value)* pair is used for retrieving and locating the data item on a peer. CAN[1] is the first architecture of peer-to-peer overlay network. The hosts are located in a geographic way. The overall hosts' spaces are divided into d-dimension Cartesian coordinated spaces. Each host in overlay network belongs to on a distinct zone in the overlay network.

Chord[2] is another famous structured peer-to-peer overlay network. Chord[2] organizes the hosts in a ring structure. Each host maintains a finger table as the routing table and the routing path will be according to the finger table. The query message will be transmitted in a clockwise way until the data piece is found or not.

2.2. Topology-Aware P2P Network

The above peer-to-peer overlay networks all have mismatch problem because they do not consider the underlying network topology in constructing the overlay network. Liu *et al.* propose several solutions to resolve the mismatch problem for unstructured pee-to-peer overlay network. In 2005, Liu *et al.*[12] propose a location-ware topology matching(LTM) technique to solve mismatch problem in unstructured peer-to-peer overlay network. In LTM host does not require global topology of peer-to-peer overlay network to optimize the overlay network structure. LTM issue a detector to detect the delay information in a constrained range(hops) and hosts collect these information to estimate the optimized overlay network structure. At the same time, Liu *et al.*[13] also propose another mechanism Adaptive Connection Establishment(ACE) algorithm to resolve the mismatch problem. Hosts collect the delay information by send the probing message and then calculate the minimum spanning tree as the optimized overlay structure according to the collecting delay information. Based on the optimized overlay structure, the host will probe other hosts to find the other closed host and try to establish overlay connection with the better host. Another mechanism scalable bipartite overlay scheme(SBO) is proposed by Liu *et al.*[22] to resolve the mismatch problem and that is similar with the ACE algorithm. But SBO divide the hosts into two types, one is responsible for collecting delay information and the other is responsible for calculate the optimized overlay structure. Therefore, overall performance of algorithm of finding optimized overlay network is improved.

Xin Yan Zhang *et al.*[17] propose mOverlay to resolve mismatch problem also. They take the locality of the hosts, i.e. distance, into account to construct the overlay network by using dynamic landmark. They introduce the group concept that hosts in group have same distance with group's neighbors. The group's neighbors are the dynamic landmark node to find the minimum distance.

Tongqing Qiu *et al.*[18] propose a generic approach to construct the topology-aware overlay network and they also use landmark as the basic scheme. By using the information from the landmark, the two hosts will be swapped to have better performance in overlay network. The decision for swapping is based on the calculation of delay information before swapping and after swapping. If the performance before swapping is better than after swapping, the two hosts will not swap to exchange hosts' information. Otherwise, they swapped to have better overlay network topology.

Guangtao Xue *et al.*[19] propose a two hierarchy architecture to construct overlay network topology to resolve mismatch problem. The hosts in lower hierarchy are closest hosts. If the closed host can not be found in the lower hierarchy, they will search for high layer hierarchy to find the closed hosts. Based on the two hierarchy architecture, the locality of the hosts in overlay network can be realized to solve the mismatch problem.

Guoqiang Zhang *et al.*[20] propose a simple approach to solve mismatch problem by collecting global information from BGP table in Internet. The global information reveals the global information with respect to the hosts in the overlay network. Therefore, the topology-aware topology can be constructed in a simple way.

Sylvia Rantnasamy *et al.*[21] use landmark information to calculate the network latency and they present binning scheme to divide the hosts into different clusters based the landmark information. Therefore, the hosts in the same cluster will have short distance and the better overlay structure is established.

The basic solution of the above literatures is to gather the network delays of the hosts in overlay network and then produces the optimized topology. The topology status is defined as the delay information between hosts. But in our opinion, the delay information just reveals the performance of the association path between hosts in overlay network and can not represent the actual topology of the overlay network. Therefore, our proposed method does not consider the delay information only and we also take the underlying network hop information into account.

3. Topology-aware p2p improvement algorithm

3.1. Topology Inconsistent Overlay Network

In previous work, most researchers use network latency or network delay as the measurement for host distance. In our work, we introduce hop-based solution as another viewpoint that is different from previous works. Network latency is calculated alone with a path from one host to another host, therefore, there exist some edges will be calculated more than ones so that this situation could be improved. In addition, the network latency is calculated more than ones, therefore, there exist redundant message transmission over the transmission path. We illustrate an example in Figure 2. There exist a overlay network include host A, host B and host C. Host B is the neighbor of host A and host C is the neighbor of host B. Host A resides on network node 1, host B resides on network node 4 and host C resides on network node 5. network node 2 and 3 might be router or gateway. We suppose host A send a message to host C via host B according to the overlay network in Figure 2. The message will be sent from network node 1→ network node 2→ network node 3→ network node 4→ network node 3 → network node 5. Therefore, we can find the revisit edges between network node 3 and network node 4. This might cause erroneous judgement of optimized topology.

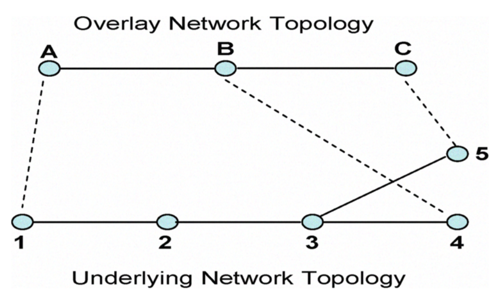

Figure 2: An Example of Revisit Path

3.2. Overlay Topology Improvement Scheme

3.2.1 Network Hop Tracer Mechanism

Based on our observation, we introduce network hop-based counting and detect the revisit edges along the query path to make the overlay network topology-aware. In practical, we can use network diagnose tool, *traceroute* or *tracert*, to explore the network path. These tools can trace the network address from source host to destination host so that the network hop can be recognized.

Because query message is delivered from source host to destination host via some intermediate hosts, therefore, in order to obtain network hop information we design a *tracerHop-K* algorithm that is extended from the mechanism of *traceroute* or *tracert* tool to explore network hops between two hosts. Message delivery path is based on peer-to-peer overlay network topology, so the virtual delivery path is according to peer's connectivity in peer-to-peer overlay network. Each intermediate host on the path will execute *tracerHop-K* to get the network hop information to next host and then reply the network information to source host alone the message delivery path. For example, host A will have network hop information 1→2→3→4, and host b will have 4→3→5 and reply to host A. Figure 3 is an example of network hop information gathering process and Figure 4 is the illustration of the *tracerHop-K* algorithm.

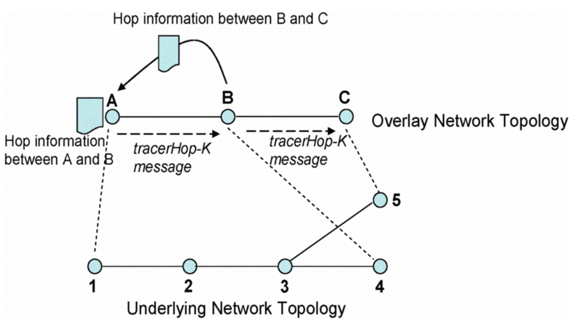

Figure 3 : Network Hop Tracer Example

INPUT : $ADDR_{dest}$, K
OUTPUT : $INFO_{hop}$
IF $ADDR != ADDR_{dest}$ *AND* $K > 0$
        $HOST_{next} \leftarrow FIND(TABLE_{host})$
        $INFO_{hop} \leftarrow traceroute(HOST_{next})$
    $K \leftarrow K - 1$
IF $K > 0$
        $SEND(HOST_{next})$

Figure 4 : *tracerHop-K* Algorithm

3.2.2. HopK-based Optimization Scheme

In order to have topology-aware peer-o-peer overlay network, the host connection relationship that forming the peer-to-peer overlay network need to take underlying network information into account. Because peer-to-peer overlay network is a distributed architecture, each host is self-organized and self-managed so that having global underlying network information in peer-to-peer overlay network is a difficult process. Therefore, in our research we also use the similar concept that using local information to estimate global optimized topology. Each host will send a *tracerHop-K* message to have local network hop information within *K* host away from source host in peer-to-peer overlay network. After gathering the network hop information within the range of overlay network hop *K*, the host will construct an argument network topology which are connecting with peer-to-peer overlay network host and underlying network routers. Figure 5 is the example of argument network topology with 2 hops. The host with wathet blue color is the host that constructing the argument network topology for topology improvement. The host with red color is 1 hop from the wathet blue host and the host navy blue color is 2 hops from the wathet blue host. The black nodes are the router between adjacent hosts in peer-to-peer overlay network.

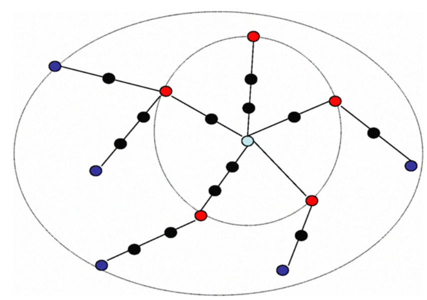

Figure 5 : Example of Argument Network Topology

When the host constructs the argument network topology, the host will detect the repeat edges or cycle edges in the argument network topology and add the final host in the path to the host's neighbor. Figure 6 is the algorithm of the *HopK*-Based optimization scheme. The algorithm will be executed in each host of peer-to-peer overlay network periodically.

In the following we conclude the improvement of overall performance with respect to the network transmission time. In general, the overall network transmission time is a summation of transmission delay, queuing delay, propagation delay and processing delay. Equation (1) is the formula that describes the overall network delay of peer-to-peer overlay network.

$$D_{network} = \sum_{1}^{n}\sum_{1}^{m} D_{host} = \sum_{1}^{n}\sum_{1}^{m} D_r + D_q + D_p + D_{prc} \quad (1)$$

$D_{network}$ is the overall network delay of peer-to-peer overlay network. D_{host} is the network delay with respect to host of peer-to-peer overlay network. D_t is the transmission delay, D_q is the queuing delay, D_p is the propagation delay and D_{pro} is the processing delay. Because our method is to reduce the revisit edges when transmitting messages over peer-to-peer overlay network, the number of pass through router and the redundant messages are decreased. So the transmission delay, queuing delay, propagation delay and processing delay are decreased. Therefore, we can derive the result after using *HopK*-Based Optimization Scheme is better than before using *HopK*-Based Optimization Scheme simply.

```
function detect(PATH_a->b)
begin
      for each HOST_i in PATH_a->b
            if HOST_i<>HOST_a and HOST_i<>HOST_b
                  if predecessor(HOST_i) == successor(HOST_i)
                        return true
                  else
                        return false
end
function Topology_Optimization
begin
      for each INFO_hop in Argument_Topology T
            for each HOST_i in INFO_hop
                  if detect(PATH_HOST->HOST_i)
                        add HOST_i as HOST's neighbor
                        break
end
```

Figure 6 : *HopK*-Based Optimization Scheme

4. System implementation

In order to make the topology-aware overlay network useful, we also propose the design of ubiquitous device architecture based on peer-to-peer computing mechanism and our proposed topology-aware improvement scheme. Figure 7 shows the software architecture of ubiquitous device. P2P lookup mechanism and routing mechanism is the basic component in P2P-bsed software, in order to make topology-aware peer-to-peer overlay network, we add the topology-aware estimation component with the topology-aware improvement mechanism describing in previous sections.

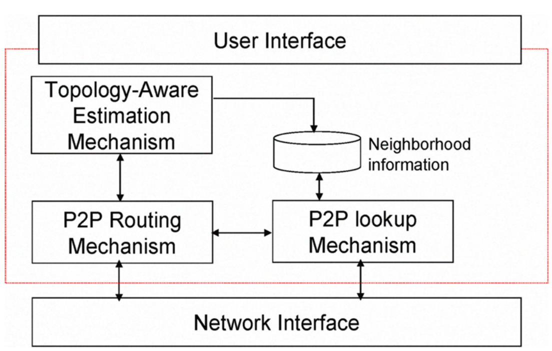

Figure 7 : Ubiquitous Device Software Architecture

5. Conclusion

In this paper, we introduce using network hop information as the estimation factor to resolve mismatch problem between peer-to-peer overlay network and underlying network. From the transmission path analysis, we can discover the proposed approach that could improve the peer-to-peer overlay network topology according to the network hop information between each virtual connection hosts. Therefore, ubiquitous applications will become more efficient and improve overall performance with the proposed method.

6. Reference

[1]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in Proceedings of the ACM SIGCOMM, 2001, pp. 161-172

[2]. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Transitions on Networking, vol. 11, no. 1, pp. 17-32, 2003

[3]. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," on Proceedings of the Middleware, 2001

[4]. Y. Zhoa, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowica, "Tapestry: A resilient global-scale overlay for service deployment," IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, pp41-53, January 2004

[5]. Napster.[Online]. available: http://www.napster.com/

[6]. (2001)Gnutella development forum, the gnutella v0.6, [Online]. Avaliable: http://groups.yahoo.com/group/the_pdf/files/

[7]. Clarke, O. Sandberg, B. Wiley, and T. W. Hong.(1999) Freenet: A distributed anonymous information storage and retrieval system. Freenet White Paper. [Online]. Available: http://freenetproject.org/freenet.pdf

[8]. (2001)Fasttrack peer-to-peer technology company. [Online]. Available : http://www.fasttrack.nu/

[9]. (2001)Kazaa mdeia desktop. [Online]. Available: http://www.kazaa.com/

[10]. (2003)Bittorrent. [Online]. Available: http://bitconjurer.org/BitTorrent/

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

[12]. Yunhao Liu, Li Xiao, Xiaomei Liu, Lionel M. Ni, and Xiaodong Zhang, "Location Awareness in

Unstructureed Peer-to-Peer Systems," IEEE Transaction on Parallel and Distributed Systems, vol. 16, no. 2, pp. 163-174, February 2005

[13]. Li Xiao, Yunhao Liu, and Lionel M. Ni, "Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment,", vol. 54, no. 9, pp. 1091-1103, September 2005

[14]. M. Waldvogel and R. Rinaldi, "Efficient topology-aware overlay network," in Proceedings of HotNets-I Princeton, NJ, USA, October 2002

[15]. Z. Xu, C. Tang, and Z. Zhang, " Building topology-aware overlays using global soft-state," in Proceedings of ICDCS 2003, pp. 500-508, Providence, RI, USA, May, 2003

[16]. Shansi Ren, Lei Guo, Song Jiang, and Xiaodong Zhang, "SAT-Match: A Self-Adaptive Topology Matching Method to Achieve Low Lookup Lantency in Structured P2P Overlay Networks," in Proceedings of the 18th International Parallel and Distributed Processing Symposium, New York: IEEE Press, 2004. pp. 83-91

[17]. Xin Yan Zhang, Qian Zhang, Member, Gang Song, and Wenwu Zhu, "A Construction of Locality-Aware Overlay Network:mOverlay and Its Performance", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 22, NO. 1, JANUARY 2004, pp18-28.

[18]. QIU Tong-Qing, CHEN Gui-Hai, "A Generic Approach to Making P2P Overlay Network Topology-Aware", Journal of Software, Vol.18, No.2, February 2007, pp.381-390.

[19]. Guangtao Xue, Yi Jiang, Jinyuan You, Minglu Li, "A Topology-Aware Hierarchical Structured Overlay Network based on Locality Sensitive Hashing Scheme", in Proceedings of the second on the Use of P2P, GRID and Agents for the Development of Content Networks(UPGRADE-CN'07), June 26, 2007, Monterey, California, USA. pp. 3-8.

[20]. Guoqiang Zhang, "Agent Selection And P2P Overlay Construction Using Global Locality Knowledge", in Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, UK, 15-17 April 2007. pp. 519-524.

[21]. Ratnasamy, S. Handley, M. Karp, R. Shenker, S., "Topologically-aware overlay construction and server selection", in Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM2002). pp. 1190-1199.

[22]. Yunhao Liu, Li Xiao, and Lionel M Ni, "Building a Scalable Bipartite P2P Overlay Network", IEEE Transactions on Parallel and Distributed Systems, Vol. 18, No. 9, September 2007, Pages 1296-1306.