

A Unified Framework for Resource Discovery and QoS-Aware Provider Selection in Ad Hoc Networks *

Jiangchuan Liu¹
csljc@cs.ust.hk

Qian Zhang²
qianz@microsoft.com

Bo Li¹
bli@cs.ust.hk

Wenwu Zhu²
wwzhu@microsoft.com

Jun Zhang³
junzhang@csd.uwm.edu

¹Department of Computer Science, The Hong Kong University of Science and Technology

²Wireless and Networking Group, Microsoft Research, Asia

³Department of Electrical Engineering and Computer Science, University of Wisconsin, Milwaukee

Abstract -With the rising popularity of network-based applications and the potential use of mobile ad hoc networks in civilian life, an efficient resource discovery service is needed in such networks for quickly locating resource providers. In addition, to improve user experience, QoS awareness is also crucial. In this paper, we identify the challenges when basic resource discovery techniques for the Internet are used in mobile ad hoc networks. We then propose a framework that provides a unified solution to the discovery of resources and QoS-aware selection of resource providers. The key entities of this framework are a set of self-organized discovery agents. These agents manage the directory information of resources using hash indexing. They also dynamically partition the network into domains and collect intra- and inter-domain QoS information to select appropriate providers. Simulation results show that our framework improves the QoS delivered to the clients, while the cost and response time are kept at a low level.

I. Introduction

An ad hoc network is generally formed by a set of wireless mobile nodes (hosts). Communication between two network nodes that are not in direct radio range takes place in a multi-hop fashion, with other nodes acting as routers. Ad hoc networks can be used in military and rescue operations, as well as in meetings where people want to share information quickly.

Recently, the rising popularity of network-based applications among end users and the potential use of ad hoc networks in civilian life have led to research interests in resource sharing in large-scale ad hoc networks [25]. With the rapid increase of available resources and accessing requests, a crucial requirement here is that a resource should be located without excessive overhead and long latency. In addition, providing desirable Quality-of-Service (QoS) is an important design objective. Specifically, when there are multiple/replicated providers for the same resource, the best one should be selected according to some QoS metrics to improve user experience. That is, an efficient and QoS-aware resource discovery system is needed.

Most previous work on resource discovery has focused on fixed-infrastructure networks, specifically, the Internet [2,4,6]. However, ad hoc networks have several distinct features which make this more challenging. The most important feature is that the topology of an ad hoc network changes with time. As a result, the design of the routing protocols for ad hoc networks is quite different from that for the Internet. For example, it has been shown that, in this case, re-active (on-demand) routing protocols are usually more efficient and scalable than traditional pro-active (table-driven) protocols [12]. In addition, to be robust in

face of topology changes and node failures, applications for an ad hoc network generally prefer distributed and dynamic control mechanisms to centralized and static mechanisms, though the latter has proven to be efficient for many Internet applications or services, such as the Domain Name System (DNS) [20].

Furthermore, in previous resource discovery systems, the QoS to be delivered to a client is seldom considered. Some systems propose to use client-based probing techniques after discovery [8,18]. However, probing measures the QoS in a very short period. In our simulation, we find that it is not very effective in mobile ad hoc networks because of the mobility and wireless channels variations.

Some discovery standards have been proposed for ad hoc network, such as the Service Discovery Protocol for Bluetooth [23]. However, they are limited to very small scale networks, and do not consider QoS. In this paper, we propose a novel framework concerning resource discovery and provider selection in mobile ad hoc networks with cooperative nodes. This framework is targeted for large-scale networks. It provides a unified solution to the problems of the discovery of resources and the QoS-aware selection of resource providers. Furthermore, it has relatively low discovery latency and cost (in terms of the number of packets for each resource discovery query).

The key entities of our framework are a set of self-organized *Discovery Agents* (DAs), which efficiently integrate three functionalities that are specially designed for mobile nodes: (1) Directory information organization and query; (2) Dynamic domain formation; (3) Intra- and inter-domain QoS information monitoring. The effectiveness of our framework is demonstrated through simulation. The results show that it produces significantly performance gain over the case where QoS is not considered. It also outperforms the case where QoS is considered but is estimated by probing. At the same time, it has relatively low cost and

* Jiangchuan Liu's work is partial supported by a Microsoft Fellowship. Bo Li's work is supported in part by Research Grant Council (RGC) under contracts AoE/E-01/99 and HKUST6163/00E.

response time.

The rest of the paper is organized as follows. Section II provides a brief review of existing resource discovery techniques and identifies the limitations when they are applied to mobile ad hoc networks. Section III proposes our QoS-aware resource discovery framework. Section IV evaluates the performance of our framework. Finally, Section V concludes the paper and discusses some future directions.

II. Existing Work and Our Design Rationale

In this section, we review existing resource discovery and provider selection techniques for the Internet and identify their potential advantages and limitations when they are used for ad hoc networks. Most of these techniques can be classified into the following three approaches.

1. Query flooding [1] and path probing [29,30].

Query flooding is the most straightforward approach for resource discovery. In this approach, a discovery query is sent to all nodes using broadcast. Each node can determine how it will process the query and respond accordingly. Its advantage is flexibility in query processing. However, the broadcast range and frequency need to be carefully controlled because broadcasting to the whole network consumes bandwidth and computation power; both are scarce in an ad hoc network.

Path probing [29] is the basic way for measuring the path QoS between a resource provider and a client. *Ping* probes has been widely used in the Internet environment [29,30] to measure response time. Bandwidth can be measured by the packet-pair technique [31]. Nevertheless, as we said before, probing may not be effective in the highly dynamic ad hoc networks as it measures path QoS for only a short time. In addition, with on-demand routing protocols, probing may initiate route discovery process, incurring high cost.

2. Centralized directory service [6,7,14].

In a centralized directory-based system, directory information of resources, such as meta data and addresses of resource providers, is registered at directory servers. To search the directory information of a requested resource, a client contacts its corresponding directory server. For the Internet, this approach has shown to be very efficient for resource discovery [2]. In fact, the Internet usually use multiple directory servers that are hierarchically organized to improve query responsiveness and scalability [4,6,7]. QoS-awareness can also be easily incorporated into this hierarchy by statically partitioning the network into domains [7].

Centralized server based techniques are also used in provider selection. One example is the use of the Domain Name System (DNS)-based server selection [27], which exploits transparent nature of name resolution to redirect clients to an appropriate server. It relies on clients and their local name server being in close proximity, since redirection is based on the name server originating the request rather than the client. Another example is the IDMaps project[21] which aims at providing a distance map of the Internet from which relative distances between hosts on the

Internet can be gauged, and the closest provider can be located based on map. The architecture of IDMaps consists of a network of instrumentation boxes, called Tracers, distributed across the Internet. Tracers measure distances among themselves and between themselves and regions of the Internet to build the distance map. In [28], the issues of placing a given number of tracers in different topologies are addressed and several heuristics are proposed to improve measurement accuracy in hierarchical topologies with partitioned domains.

However, in mobile ad hoc networks, since there is no fixed topology, maintaining a hierarchal structure of directory or measurement servers is not an easy task. Moreover, statically configured domains do not reflect the dynamic relations of mobile nodes.

3. Decentralized hash indexing [15,16,26].

Decentralized hash indexing has been proposed for resource discovery in peer-to-peer networks. In such a system, there is no special/centralized directory server. Instead, every node provides some directory service. A resource is given a unique key, and a hash function is used to build a deterministic mapping between the key and the node which stores the directory information of that resource. The network and peers are designed in such a ways that, given a key, the corresponding resource can be located very quickly despite the network's size. However, in this approach, each network node could be involved in some queries. In an ad hoc network using on demand routing protocols, if a node has not communicated to other nodes for a certain time, a route discovery process is needed to find a route towards this node, which may incur high cost [10,11]. Furthermore, this approach does not address QoS issues.

Through analyzing the advantages and limitations of the existing approaches, we have arrived at the following design principles for QoS-aware resource discovery in mobile ad hoc networks. First, directory information should be distributed to only a small set of fault-tolerant directory agents. Most messages for discovery are exchanged among these agents to reduce the overhead of broadcast and route discovery. Only low-frequency or controlled broadcast is used to distribute some quasi-static or local information, such as the addresses or locations of the agents. Second, hash indexing can be applied to these agents to reduce query latency. Finally, QoS information should be monitored continuously using distributed mechanism. These principles have lead to our novel framework, described next.

III. A Framework for Resource Discovery and QoS-Aware Resource Discovery

III.A. Framework Overview

Our framework is built on the application layer to provide generic and efficient tools for QoS-aware resource discovery.

In our framework, we assume that all nodes are coopera-

tive and can communicate with each other via some single-hop or multi-hop path. Each node can take one or more of the following three roles:¹

A **Client** that initiates a query for resource discovery and uses resources. There are two basic discovery modes: a *Browsing mode* where a client is looking for all resource providers that have the requested resource, and an *Accessing mode* where a client is looking for a resource provider that could provide the best quality-of-service.

A **Resource Provider (RP)** that provides resources for clients. A RP is also responsible for registering the directory information of its resources and advertising its QoS information to discovery agents.

A **Discovery Agents (DA)** that performs many of the important operations in our framework. First, DAs collectively maintain directory information of the resources using hash indexing. This provides fault-tolerance and fast query response. Second, DAs dynamically partition the whole network into *dynamic domains*. Each DA maintains a separate domain and acts as the home DA of that domain. It monitors the QoS information of the RPs in its domain, and responds to discovery queries from clients in the domain. Third, all registration and query messages are exchanged between DAs. These frequently exchanged messages are also used to continuously estimate peer path QoS, such as the delay between two DA nodes.

III.B. DA Generation and Dynamic Domain Formation

Initially, there are no DA nodes in the network. They are generated through a bootstrapping process as follow. First, one node is elected as the initial DA using a procedure similar to the cluster head selection in the lowest-ID algorithm [22] for ad hoc networks. That is, all eligible nodes broadcast to the whole network about their existence to take part in the election, and the one with the smallest address will win the election. Suppose there are M DAs to be generated (the choice of M will be studied in the next section), the initial DA will then randomly select another $M-1$ nodes to form the DA set, and assign each of them a unique index in the set of $\{2, \dots, M\}$. Specifically, the initial DA has index 1.

After the DAs are generated, their addresses are periodically broadcasted to the whole network at a low frequency. In addition, each non-DA node tries to find the nearest DA as its home DA, and join that DA's domain.

Note that both DAs and other nodes move over time. Hence, the members in a domain changes over time and a dynamic domain formation process is periodically performed for a DA to update its domain members. Here, a non-negative and additive metric is used to measure distance, which can be the number of hops or delay in practice. Based on the properties of shortest paths with this type of metrics [13], we propose a simple distributed algorithm to form dynamic domains, as follows. A DA periodically broadcasts a formation announcement to its neighbor-

ing nodes, which includes the DA's index, expiration time of the announcement, and a distance field. The distance field records the distance between the DA and the node that receives the announcement. Upon receiving an announcement, a non-DA node first checks the value of the distance field, if it is greater than the distance to its current home DA, the node stops forwarding the announcement. Otherwise, it will set that DA as its home DA, and forward the announcement to all its neighboring nodes. To proof the correctness of this algorithm, we must show that (1) any non-DA node should be in a DA's domain; (2) the DA is the nearest DA to that non-DA node. The first property can be proved as follows. Suppose a non-DA node is not covered by any DA's domain, then its neighboring nodes should not be covered by any DA's domain, too. Thus, by using induction on these nodes, we can conclude that either there is no any DA in the network or there is a set of nodes that cannot communicate with the remaining part of the network. Both contradict our basic assumptions. The second property can be proved by the criterion for DA selection in the algorithm.

III.C. Directory Information Organization and Fault Recovery

In our framework, each resource has an attribute known to all intended clients in the network. To register a resource, its provider first issues a registration request to its home DA. The request includes the provider's address, attribute, expiration time, and other directory information. Assume the attribute of the resource is α , a hashing function $H(\cdot)$ is used to produce a index $=H(\alpha)$ in the set of $\{1, 2, \dots, M\}$. The home DA will then distribute the registration request to $DA_\beta, DA_{\beta+1}, \dots, DA_{\beta+K-1}$, and the directory information of the resource will be registered to these DAs.

This organization scheme has several advantages. First, the replicated providers of the same resource always register to the same DAs; hence we can obtain the full list of their directory information from only one DA. Second, the directory information of a resource can be quickly located by using hash indexing. Note that different resources may have the same attribute and their directory information will thus be stored in the same DAs. Hence, our framework does not preclude the use of fuzzy search in a DA, such as wildcard-based search. Third, this scheme provides fault tolerance if K is greater than 1. Suppose the nodes are homogeneous with failure probability p , the number of replications, K , should be set to $\log_{1-p} A$ where A is the availability requirement for the directory information. When a DA is found failed by another DA in the discovery query process (it will be discussed in Section III.E in detail), the latter will broadcast a DA selection message to the network. Non-DA nodes that are willing to take the place of the failed DA will respond to this message and the one with the minimal last-known distance to the failed DA will be selected. Assume the index of the failed DA is i , the directory information can then be recovered from a subset of $DA_{i-K+1}, DA_{i-K+2}, \dots, DA_{i+K-1}$.

¹A node can take one or more functions. Specifically, a node that takes DA functions is called a DA node, or DA in short, and other nodes are called non-DA nodes.

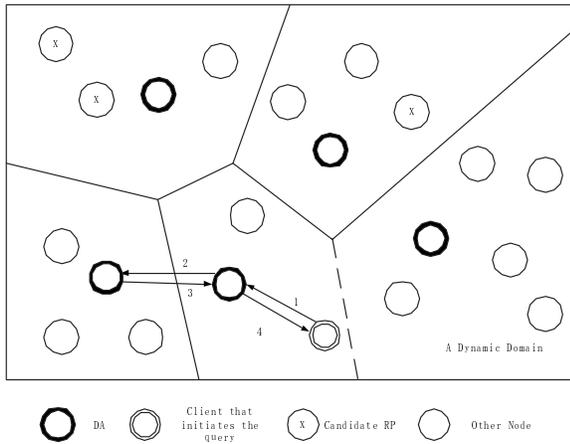


Figure 1: Operations for resource discovery, browsing mode.

III.D. QoS Information Collection and Prediction

A DA is also responsible for QoS information collection and prediction. Note that the requirements of QoS are highly application-specific. Hence, our framework provides generic QoS information to different application to achieve a flexible solution. Specifically, the first type of QoS information is application-level QoS, including the CPU usage and available memory of a RP. A RP periodically provides this information to its home DA. The second type is path QoS between two nodes. In this paper, we consider the path delay (packet latency) between two nodes, which is one of the most useful path QoS metrics for many applications [21]. However, other path metrics, such as bandwidth, can also be incorporated into the system. We assume the clocks of all DAs are synchronized by some global time service, such as the Universal Time Coordinate (UTC) service provided by the Global Position System (GPS) [24], and a message exchanging between DAs carries a timestamp. Thus, DAs can predict their peer path delay by an Autoregressive Moving Average (ARMA) predictor [17], which uses the packet latency calculated from those frequently exchanged messages. For non-DA nodes, we do not directly measure their path QoS by exchanging probing message between RP-Client pairs. This is because, first, probing may trigger high cost route discovery operations if two nodes seldom communicate with each other, and second, the time for using a resource is usually much longer than the time for probing, and a short time probing may give a different estimation comparing to the statistical behavior of a path. Hence, instead of using probing, we use an approximation method. We assume that the nodes in a dynamic domain are QoS-similar, and use the home DA as a representative. The path QoS of two non-DA nodes is approximated by the path QoS of their home DAs. When there are enough DAs that move independently, the error of this approximation is expected to be small, as shown in Section IV.

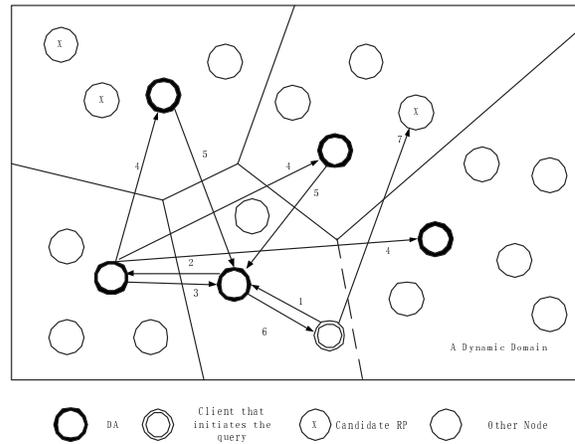


Figure 2: Operations for resource discovery and QoS-aware provider selection, accessing mode.

III.E. Discovery Query

In our framework, resource discovery and selection is done in two phases. The first is directory query for searching the resource directory information in the DA set. Figure 1 shows an example. Starting from the client's home DA (denoted as hDA) to which a query is submitted (Step 1 in Figure 1), if no cached record matches the query, hDA will calculate the hashing index of the resource, β , to decide the qualified DA set, $DA_\beta, DA_{\beta+1}, \dots, DA_{\beta+K-1}$. The query is then forward to the DA which is in the qualified DA set and is nearest to hDA (Step 2). If this DA fails, hDA will try to forward the query to the next nearest DA in the qualified set, until the query is successful. In the browsing mode, a full list of RP candidates (the providers that have the requested resource) is returned to hDA (Step 3), and then to the client that initiates the query with browsing mode (Step 4).

The second phase is QoS query, which is for accessing mode only. It needs to compare the QoS provided by all RP candidates, and select the best one. Towards this end, hDA should query all DAs that are home DAs for the candidates. In the current version, we use a parallel search strategy, see Figure 2. The DA that has the directory information sends a QoS query to all the DAs using multicast, or multiple-unicast if the underlying routing protocol does not support multicast (Step 4). The query includes the index of hDA, the list of the RP candidates, and the type of QoS of interest. If there are one or more candidates in a DA's dynamic domain, the DA will respond to hDA by providing the addresses of the candidates and the corresponding QoS information (Step 5). The QoS of all RP candidates are then compared by hDA according to the requirement of the client, and the result is returned to the client (Step 6). Finally, the directory information of the best candidate is returned to the client (Step 6) which accesses resource using appropriate protocols (Step 7).

IV. Performance Evaluation

In this section, we present our initial simulation results. Our main objective for the study is to investigate whether our framework enhances QoS-awareness in resource discovery. Another objective is to see whether our framework exhibits satisfactory performance in terms of query responsiveness and overhead.

IV.A. Simulation Environment

We simulate our framework using the LBNL network simulator ns-2 [19]. Our simulated network consists of 150 mobile nodes, whose initial positions are chosen from a uniform distribution over an area of 1000 m by 1000 m. Random waypoint [11] is used as the mobility model. The nodes' moving speeds are uniformly chosen from 10 to 72 km/hr. The IEEE 802.11 protocol is used as the MAC layer protocol. Each wireless channel has 2 Mbps bandwidth and a circular radio range with 250 m radius. For routing, we use the Ad hoc On-demand Distance Vector (AODV) protocol [10].

We assume that there are 100 different resources in the simulated environment. The popularity of each resource, measured in number of requests per minute, is randomly distributed in 1 to 5 requests per minute. For each discovery query, the client that initiates the query is randomly selected in the network. Each resource is served as a Constant Bit Rate (CBR) streaming application of 28 Kbps and lasting 30 seconds. In the experiments, the path QoS of interest is the average packet latency. We choose this metric because it is relatively easy to estimate, and luckily the most generally useful [21].

For the sake of comparison, we also simulated a traditional framework in which locating a resource and selecting a provider are considered as two separate issues. The resource discovery method is centralized directory-server based. When there are replicated providers, a client sends 15 consecutive packets to each provider to estimate path delay, and selects the one with the minimum average delay.

IV.B. Query Latency and Cost

In the first set of simulation experiments, we investigate the query latency and cost of ours and the traditional. We vary the number of replicated providers, N_p , for each resource, from 1 to 10. We find that the average query latency is nearly independent of N_p in both frameworks, and also independent of M , the number of DAs in our framework. This is because that QoS queries to different DAs or probes to different providers are sent simultaneously. Table 1 lists the query latency in different phases, including directory query and QoS query. It can be seen that, in our framework, the average latency for directory query is higher than that in the traditional framework. This is because, in the traditional framework, a client need to contact a directory server only, while in our framework, a client need to contact not only the home DA but also the DA that stores directory information. However, the latency of QoS query in our framework is much lower than that in the traditional

| Framework | Directory Query | QoS Query | Overall |
|-------------|-----------------|-----------|---------|
| Traditional | 287 | 1375 | 1662 |
| Proposed | 395 | 354 | 749 |

Table 1: Comparison of discovery latencies (ms).

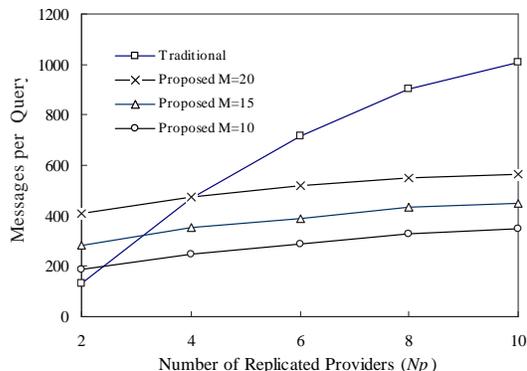


Figure 3: Comparison of discovery costs.

framework. As a result, the total latency in the traditional framework is about 2.2 times of ours. This is because QoS query in our framework involves only querying to all DAs, and the latency is thus bounded by the time-out factor between the home DA and the furthest DA. On the other hand, probing involves not only the transmission of a packet from a client to a provider, but also several cycles of this process.

Figure 3 shows the average cost in terms of the number of messages (packets) transmitted in the network for each discovery. We observe that, in our framework, the cost is also nearly independent of the number of replicated providers because the QoS queries always sent to all the DAs regardless of the number of providers. On the contrary, in the traditional framework, since the client need to probe every resource provider, the cost increases nearly linearly with the increasing number of replicated providers.

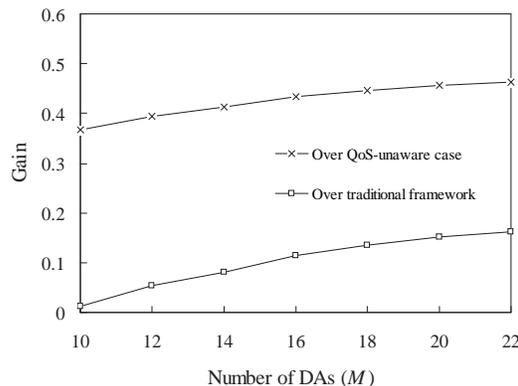


Figure 4: The performance gains of the proposed system over the traditional system and the QoS-unaware case.

When there are more than 4 replicated providers, the traditional framework incurs much higher cost than ours. Note that in the QoS query phase of our framework, it is possible to use some heuristic algorithm to avoid querying DAs whose domain does not cover any qualified provider. We will study this approach in our future work.

IV.C. Quality of Service

In this set of experiments, we studied the QoS-awareness of the two frameworks. The metric of interest is our framework's performance gains over the traditional framework and the QoS-unaware case (the resource provider is randomly selected), where the gains are calculated by normalizing the reduced packet latencies.

Figure 4 shows the results with $N_p=5$. The number of DAs varies from 10 to 20. We can see that performance gain increases with the increase of the number of DAs. This is because the expected area of a dynamic domain decreases when there are more DAs, and hence, in this case, it is more accurate to approximate the path QoS between two nodes using their home DAs as representatives. Specifically, when enough DAs are deployed, the gain of our framework is up to 45% compared to the QoS-unaware case, and 15% to that of the traditional framework. Moreover, the gain tends to saturate when there are more than 20 DAs in this 150-node network. Hence, from Figures 3 and 4, a choice between 15 and 20 DAs provides satisfactory performance in terms of both discovery cost and accuracy. This number is much smaller than the total number of nodes.

V. Conclusions and Future Work

In this paper, we first identified the limitations when basic resource discovery techniques are used in mobile ad hoc networks. We then proposed a novel framework that is specially designed for QoS-aware resource discovery in ad hoc networks. Our framework jointly considers the problems of resource discovery and QoS-based provider selection. The key entities in our framework are a set of self-organizing discovery agents. The agents adopt an efficient hash indexing method to store directory information. They also monitor QoS information continuously and predict path QoS on behalf of other nodes to reduce overall cost and improve accuracy.

Preliminary simulation results showed that our framework enhances QoS-awareness comparing to a traditional framework which uses centralized directory service and client-based probing. In addition, our framework incurs lower query latency and cost. We will conduct more experiments in our future work to investigate the behavior of this framework, such as potential oscillation among providers [18], scalability in large scale networks, and performance with other QoS metrics.

References

- [1] R. Iannella, *Internet Resource Discovery Issues*, Available at archive.dstc.edu.au/RDU/reports/QuestNet95.html
- [2] *Decentralized Resource Discovery in Large Peer-based Networks*, Available at cubicmetercrystal.com/alpine/discovery.html
- [3] Z. Fu and N. Venkatasubramanian, *Combined Path and Server Selection in Dynamic Multimedia Environments*, in Proceedings of ACM Multimedia 99, October 1999.
- [4] R. van Renesse, *Scalable and Secure Resource Location*, in Proceedings of IEEE Hawaii International Conference on System Sciences, January 2000.
- [5] M. S. Corson, *Issues in Supporting Quality of Service in Mobile Ad Hoc Networks*, in Proceeding of IWQOS 97, May 1997.
- [6] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, and R. Katz, *An Architecture for a Secure Service Discovery Service*, in Proceedings of ACM MOBICOM 99, September 1999.
- [7] D. Xu, K. Nahrstedt, and D. Wichadakul, *QoS-Aware Discovery of Wide-Area Distributed Services*, in Proceedings of IEEE International Symposium on Cluster Computing and the Grid, May 2001.
- [8] S. G. Dykes, C. L. Jeffery, and K. A. Robbins, *An Empirical Evaluation of Client-Side Server Selection Algorithms*, in Proceedings of IEEE INFOCOM, March 2000.
- [9] E. M. Royer and C.-K. Toh, *A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks*, IEEE Personal Communications, pp. 46-55, April 1999.
- [10] C. E. Perkins and E. M. Royer, *Ad Hoc On-Demand Distance Vector Routing*, in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999.
- [11] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, in Mobile Computing, edited by T. Imielinski and H. Korth, Chapter 5, Kluwer Publishing Company, 1996.
- [12] S.-J. Lee, *Routing and Multicasting Strategies in Wireless Mobile Ad Hoc Networks*, Ph.D. Dissertation, University of California, Los Angeles, 2000.
- [13] R. J. Wilson, *Introduction to Graph Theory*, 3rd Edition, Longman Scientific and Technical Publisher, New York, 1987.
- [14] E. Guttman, C. Perkins, J. Veizades, and M. Day, *Service Location Protocol, Version 2*, RFC 2608, June 1999.

- [15] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H Balakrishnan, *Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications*, in Proceedings of ACM SIGCOMM 01, September 2001.
- [16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, *A Scalable Content-Addressable Network*, in Proceedings of ACM SIGCOMM 01, September 2001.
- [17] R. Wolski, *Forecasting Network Performance to Support Dynamic Scheduling*, in Proceedings of the 6th IEEE International Symposium on High Performance Distributed Computing, 1997.
- [18] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, *A Novel Server Selection Technique for Improving the Response Time of a Replicated Service*, in Proceedings of IEEE INFOCOM 98, April 1998.
- [19] The LBNL Network Simulator, ns-2, available at <http://www.isi.edu/nsnam/ns/>.
- [20] P. Mockapetris, *Domain Names C Concepts and Facilities*, RFC 1034, November 1987.
- [21] P. Francis, S. Jamin, C. Jin, D. Raz, Y. Shavitt, and L. Zhang, *IDMaps: A Global Internet Host Distance Estimation Service*, IEEE/ACM Transactions on Networking, October 2001.
- [22] A. Ephremides, J. E. Wieselthier, and D. J. Baker, *A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling*, in Proceedings of the IEEE, Vol. 75, No.1, pp. 56-73, 1987.
- [23] The Official Bluetooth Website: <http://www.bluetooth.com/>
- [24] Notes on NAVSTAR GPS Operations. Available at tycho.usno.navy.mil/gpsinfo.html.
- [25] K. Konishi, *Report on Large Scale Networking Workshop*, March 2001. Available at www.jp.apan.net/meetings/0103-NGI-workshop
- [26] R. Jain, Y.B. Lin and S. Mohan, *A Caching Strategy to Reduce Network Impacts of PCS*, IEEE Journal on Selected Areas in Communications, Vol. 12, No. 8, pp. 1434-1444, October 1994.
- [27] A. Shaikh, R. Tewari, and M. Agrawal, *On the Effectiveness of DNS-based Server Selection*, in Proceedings of IEEE INFOCOM, April 2001.
- [28] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, *One the Placement of Internet Instrumentation*, in Proceedings of IEEE INFOCOM, March 2000.
- [29] R. Carter and M. Crovella, *Server Selection using Dynamic Path Characterization in Wide-Area Networks*, in Proceedings of IEEE INFOCOM 97, April 1997.
- [30] M. Crovella and R. Carter, *Dynamic Server Selection in the Internet*, in Proceeding of the Third Workshop on the Architecture and Implementation of High Performance Communication Subsystems, August 1995.
- [31] S. Keshav. *A Control-Theoretic Approach to Flow Control*, in Proceedings of ACM SIGCOMM'91, September 1991.

Biography

Jiangchuan Liu received the B.S degree (cum laude) in Computer Science from Tsinghua University, Beijing, P.R. China, in 1999. Currently, he is working towards the Ph.D degree in Computer Science at the Hong Kong University of Science and Technology. In 2000 and 2001, he had Internships with Microsoft Research, Asia, working in video multicast over the Internet and QoS-aware resource discovery, respectively. He is a recipient of a Microsoft research fellowship, and has won top honors in regional and national programming contests. His current research interests include video transmission over wired and wireless networks.

Qian Zhang received the B.S., M.S., and Ph.D. degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science.

She joined Microsoft Research China in July 1999 as Associate Researcher in Internet Media Group and now is Researcher of Wireless and Networking Group. Dr. Zhang has published about 40 refereed papers. She is the inventor of several pending patents. Her current research interest includes multimedia delivery over wireless, Internet, next generation wireless networks, P2P network/ad hoc network. Currently, she is actively participating in TCP/IP header compression in ROHC WG in IETF. She is the principle contributor of the IETF ROHC-TCP WG draft.

Bo Li received the B.S. and M.S. degrees in the Computer Science from Tsinghua University, Beijing, P. R. China, in 1987 and 1989, respectively, and the Ph.D. degree in the Electrical and Computer Engineering from University of Massachusetts at Amherst in 1993. He worked for IBM Networking System, RTP, NC before joining Hong Kong University of Science and Technology.

Bo Li has been fairly active in research community, having served and serving as editors and guest editors for over 10 journals in IEEE and ACM, and been involved in organizing 30 some conferences in IEEE and ACM, in particular IEEE Infocom.

Wenwu Zhu joined Microsoft Research China in November 1999 as Researcher in Internet Media Group and now is Research Manager of Wireless and Networking Group. Prior to his current post, he worked at Bell Labs, Lucent technologies as Member of Technical Staff from July 1996 to October 1999. While he was at Bell Labs, he performed research and development in the area of video conferencing, Internet video, and video streaming over IP networks. Dr. Zhu has published over 100 refereed papers. He is the inventor of more than a dozen of pending patents. His

current research interest is in the area of wireless/Internet multimedia communication and networking.

Dr. Zhu is member of Eta Kappa Nu. He is Guest Editor for the Special Issue on Wireless Video in IEEE Trans. On Circuits and Systems for Video Technology. He served as Guest Editor for the Special Issue on Streaming Video in IEEE Trans. On Circuits and Systems for Video Technology. Dr. Zhu are members of Technical Committees of Video Signal Processing and Communications, and Multimedia System in IEEE Circuits and System Society. He also serves member of Multimedia Communication committee in IEEE Communications Society. He has contributed to IETF ROHC WG draft on robust TCP/IP header compression over wireless links. He received the Best Paper Award in IEEE Trans. on Circuits and System for Video Technology in 2000.

Dr. Zhu received the B.E. and M.E. degrees from National University of Science and Technology, China, in 1985 and 1988, respectively. He received the M.S. degree from Illinois Institute of Technology, Chicago, IL, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1993 and 1996, respectively, all in Electrical Engineering. From August 1988 to December 1990, he was with Graduate School, University of Science and Technology of China (USTC) , and Institute of Electronics, Chinese Academy of Sciences, Beijing.

Jun Zhang received his B.S. in electrical and computer engineering from Harbin Shipbuilding Engineering Institute, Harbin, China, in 1982 and was admitted to the graduate program of the Radio Electronic Department of Tsinghua University. After a brief stay at Tsinghua, he came to the U.S. for graduate study on a scholarship from the Li Foundation, Glen Cover, New York. He recieved his M.S. and Ph.D. both in electrical engineering from Rensselaer Polytechnic Institute in 1985 and 1988, respectively. He joined the faculty of the Department of Electrical Engineering and Computer Science and currently is a professor. His research interests include image processing and computer vision, signal processing and digital communications. He has been an associate editor of IEEE Trans. Image Processing.