2nd Annual Conference on
Communication Networks and Services Research (CNSR 2004)
Fredericton, N.B., Canada, May 19-21, 2004

Performance Evaluation of Service Discovery Strategies in Ad Hoc Networks

Honghui Luo Michel Barbeau
School of Computer Science
Carleton University
Ottawa, Canada
Email: {hluo,barbeau}@scs.carleton.ca

Abstract

Service discovery is an important and necessary component of ad hoc networks. To fit within the context of such networks, a Post-query model with several service discovery strategies named Post-query strategies have been proposed, by Barbeau and Kranakis, which focus on locating services over an ad hoc network. Each strategy consists of a sequence of Post-query protocols executed in rounds. These strategies include: the greedy, incremental, uniform memoryless and with memory Post-query strategies. Inspired by these proposals, we define the conservative Post-query strategy. This paper represents the performance evaluation of these five strategies in combination with the DSR protocol and DSDV protocol based on the simulation results.

1. Introduction

An *ad hoc network* is a multihop wireless network consisting of a set of mobile nodes not requiring any preexisting network infrastructure. In contrast to traditional wired networks, it has limited resource and energy. Service discovery is an important and necessary component of ad hoc networks. Service discovery is defined as the problem of automatically locating different services within a network. Currently, there are some well known service discovery protocols such as Service Location Protocol (SLP) [5], Jini [20], and Service Discovery Protocol (SDP) [2].

SLP is designed for TCP/IP networks. It contains: *User Agent (UA)*, which locates services on behalf of a client; *Service Agent (SA)*, which provides information about a service; and *Directory Agent (DA)*, which caches service information received from SAs and replies to service requests from UAs. SLP can operate in a distributed manner (without a DA) or in a centralized manner (with one or more DAs). The Jini technology consists of: *service provider*, which is the front-end of a service available on a network; *client*, which is a user of services; and *lookup service*, which is a central node where service providers register their services (similar to the DA of SLP). Bluetooth [2] is a short range ad hoc network technology. It employs SDP for locating services provided by or available through other Bluetooth enabled devices. SDP defines the SDP server and SDP client. To find a service, an SDP client unicasts an SDP request to SDP servers one by one. SDP servers either send back SDP responses or error responses.

To fit within the context of ad hoc networks, a Post-query model with several service discovery strategies named Post-query strategies are proposed, by Barbeau and Kranakis [1] based on the distributed match-making paradigm [12][14]. This model focuses on locating services. It defines the ways of how to use the service discovery protocols in ad hoc networks.

In the Post-query model, each server posts its service(s) to other nodes according to the Posting protocol, and each client queries its desired services according to the Querying protocol. A pair of Posting and Querying protocols forms a Post-query protocol. In order to adapt to topological changes over time in an ad hoc network, several Post-query strategies are proposed. A Post-query strategy is a sequence of Post-query protocols executed in rounds. In the context of ad hoc networks, the Posting and Querying protocols are independent. These strategies include: the greedy, incremental, uniform memoryless and with memory Post-query strategies. Inspired by these proposals, we define the conservative Post-query strategy.

With the proposition of these Post-query strategies, a uniform model needs to be constructed to evaluate their performance. As application layer protocols, Post-query protocols cannot work well without the support of the network layer routing protocols. In this paper, we show the performance evaluation of those strategies when they are combined with some typical ad hoc routing protocols in the Post-query model.

The remainder of this paper is structured in the following manner: Section 2 reviews service discovery in ad hoc networks. Section 3 describes the design of the Post-query strategies. Section 4 gives an overview of the simulation environment. Section 5 presents performance evaluation of the five Post-query strategies combined with the DSR protocol [10] and DSDV protocol [18]. Section 6 puts forward conclusions and future work.

2. Service discovery in ad hoc networks

2.1. The Post-query strategies

Barbeau and Kranakis [1] define several Post-query strategies that can be used to locate a service in an ad hoc network in the Post-query model, including the: greedy, incremental, uniform memoryless and with memory strategies. The Post-query strategies are time dependent Post-query protocols that are executed in rounds. They aim at maximizing the probability of success in locating a service and minimizing the waiting time and total number of postings and queryings.

A *Post-query protocol* is defined as a pair of functions (P, Q), where $P(s)$ is the *posting protocol* and $Q(c)$ is the *querying protocol*. Let $1, 2, \ldots, k$ be the k types of services offered in the network and let $K = \{1, 2, \ldots, k\}$. Each server s posts a service of type k_s to a set of nodes N_s according to the algorithm defined in $P(s)$. Similarly, each client c queries a service type k_c from a set of nodes N_c based on the algorithm defined in $Q(c)$. A client successfully locates a service if this service has been posted by some server to a node that it queries. In the context of ad hoc networks, the topology changes as time passes, so a node may be unreachable during the execution of a Post-query protocol. Hence, a Post-query protocol is not by itself sufficient to efficiently adapt to the dynamic changes of the network topology. To solve this problem, Post-query strategies are proposed. They are executed in a sequence of rounds; at each round r a pair of Post-query protocols $(P(s), Q(c))$ are employed. The maximum round R is the upper bound of the number of rounds, after which the nodes terminate their strategies. A Post-query strategy is defined as a sequence $(P_1(s), Q_1(c)), (P_2(s), Q_2(c)), \ldots, (P_R(s), Q_R(c))$ of Post-query protocols. To model the influence of dynamic topology changes, a connection probability p is introduced with $0 \leq p \leq 1$ to indicate the probability of a communication path existing between each pair of nodes in the network. All the strategies use a theoretical algorithm. For each round r, a server s first posts its services according to the posting protocol $P_r(s)$, and a client then queries services based on the querying protocol $Q_r(c)$. In each round, all nodes follow the posting-querying order. The algorithm is executed until the maximum round index R is reached or the queried services are successfully located, whichever comes first. The proposals of each Post-query strategy are as follows:

1. The greedy Post-query strategy: all nodes post to all nodes and all nodes query all nodes in the network using the network broadcast mechanism. This strategy is *non-adaptive* because the execution in each round remains the same.
2. The incremental Post-query strategy: all servers and clients post to and query from a small set of nodes in the first round and increase the sizes of the posting and querying sets gradually.
3. The uniform memoryless Post-query strategy: This strategy is called *uniform* because the same Post-query protocol is executed in each round, and the posting and querying protocols are identical for all the nodes of the network. Servers post all the services they can offer to a random set of nodes. Clients query a random set of nodes. The chosen sizes of the posting and querying sets affect the waiting time, network overload and the probability that a given client $c \in N$ succeeds in finding a service.
4. The *with memory* Post-query strategy: each client builds a cache to store the IDs (IP addresses) of the previously visited nodes. Each new round only involves posting (querying) previously uncontacted nodes.

The simulation results show that, the probability of success within a low number of rounds increases as the connection probability p increases. The probability of success increases with the number of nodes since the number of service offers increases as well. The greedy strategy maximizes the probability of succeeding while requiring the greatest network resources. The waiting time for the greedy strategy is the shortest of all the strategies. The incremental strategy involves the longest waiting time, while it consumes the l east network resources. The waiting time and cost of the uniform memoryless strategy lie between that of the greedy and incremental strategies. When the connection probability p is greater than 0.5, the probability of succeeding is higher than with the greedy strategy.

2.2. Other related work

- Koodli and Perkins [11] define another work in progress approach for service discovery in ad hoc networks. They add extensions to suitable ad hoc routing protocols in order to provide support for service discovery along with routes to those services. In their approach, the association between a service and the IP address of the node hosting the service is known as *service binding*. For table-driven routing protocols, a Service Reply extension is added to a topology updating packet such that service information as well as routing information can be made available immediately. For source initiated on-demand routing protocols, the basic service discovery process adopts the same operations and message format as the *route discovery* process, while it adds a *service request* extension to the route request packet. No simulation results of this approach have been published yet.

- IBM has developed DEAPspace [15][8] that addresses the service discovery problem in single-hop ad hoc networks. It selects the *push model*(in which servers send unsolicited service advertisements). Each node maintains a cache to store all the services offered in the network. Nodes participate in the advertising of services through a broadcast mechanism. Periodic broadcasts are scheduled in a proactive way using an adaptive backoff mechanism. A performance evaluation of DEAPspace shows that this approach has about the same bandwidth as other push model solutions while the time for the discovery of available services is better than other push model solutions, since more service information is being propagated when one node broadcasts its entire cache than when each node of the network only broadcasts the service information which it offers. However, unnecessarily repeated broadcasts occur in DEAPspace, consuming network bandwidth. Moreover, maintaining such a tight convergence in a highly dynamic ad hoc network may not be worth the effort.
- Konark [7] is designed specifically for the discovery and delivery of services in multihop ad hoc networks. As for the service discovery aspect, Konark supports the *push model* and *pull model*. It assumes the multicast support of underlying ad hoc routing protocols. It has developed a new algorithm which attempts to balance the convergence time and network bandwidth use. Compared with the DEAPspace algorithm, when a node receives a service message, it only multicasts the difference between its own relevant services and the service information in the received service message. Hence, network overload is reduced.

3. Design of the Post-query strategies

Figure 1 shows the protocol stack of our design. Each node is either a server or a client in an ad hoc network. For each server, a Post protocol is employed and for each client a Query protocol is employed. We choose the DSR protocol and DSDV protocol as the underlying ad hoc routing protocols.

There are some common design features of these five strategies.

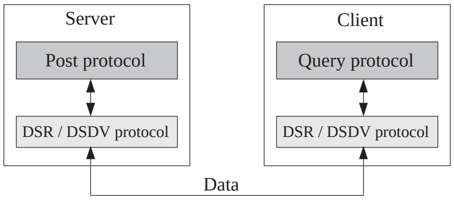

Figure 1. Architecture of protocol stack

- We focus on locating services efficiently in ad hoc networks.
- Each node of the network is required of the maintenance of a cache to store the service information from previous rounds. Thus, after each round, the nodes which receive other servers' postings become potential *servers* for the following rounds. This means that, in the following rounds when these nodes are queried for the services which they cached from previous rounds, they are capable of sending back replies.
- A server can receive other servers post messages and a client can reply other clients query messages.
- When a node receives a query from a client about a service, it sends all the matched service information back to this client. Hence, if this client wants to use this service information later, it has more choices. If this node does not have the desired service, it keeps silent and does nothing.

Based on the greedy strategy of Barbeau and Kranakis, we define the conservative strategy which requires that in each round all servers (clients) post to (query from) their one-hop neighbors using a network one-hop broadcast mechanism.

4. Simulation environment

The *Network Simulator* (NS) is chosen as our simulation tool because it meets our functional requirements. The current version is NS-2 [4]. In NS-2, the Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer protocol. The wireless interface functions similar to the Lucent WaveLAN radio interface [21]. The transmission range is about 250 meters. The signal propagation model combines both a free space propagation model and a two-ray ground reflection model.

50 nodes are simulated. Ten of them are servers and 40 of them are clients. There are 10 kinds of services offered in the network. We assume that all the services follow a uniform distribution. The Random Waypoint Mobility Model[4] is used to generate three different kinds of network scenarios:

- scen300-01: a 300 second pause time and one m/s maximum node speed;
- scen100-10: a 100 second pause time and 10 m/s maximum node speed;
- scen30-30: a 30 second pause time and 30 m/s maximum node speed.

To characterize the mobility of the nodes of an ad hoc network, a parameter called *dynamic ratio* (DR) is introduced. To calculate it, we denote the total number of nodes in the network as N and the total number of link changes of all the nodes during the entire simulation time as LC. The dynamic ratio is then defined as:

$$DR = LC/N$$

High DR means high degree of nodes mobility of an ad hoc network. We define the maximum round number R is equal to 10, as an upper bound of rounds within which servers and clients terminate their execution. If all the clients successfully located their wanted services, the simulation is terminated. Round is a time interval of 15 seconds.

5. Performance evaluation

5.1. Performance metrics

We denote the number of total nodes as N, the number of total clients as N_c, and the number of total successful clients as N_{succ}. For each round r, we denote the number of ServicePost messages as M_{post}, the number of ServiceQuery messages as M_{query}, and the number of ServiceReply messages as M_{reply}. Each successful client c receives 1 to m ServiceReply messages of waiting time $t_1,\ t_2,\ \ldots,\ t_m$.

- Success rate(SR): The ratio (as a percentage) of the number of clients which successfully locate the services, over the total client number. It is calculated using the following formula:
  $SR = N_{succ}/N_c \times 100\ (\%)$
- Number of transmitted messages(NMT): The number of messages transmitted in each round by all the nodes in the network, including ServicePost, ServiceQuery and ServiceReply messages. It is calculated using the following formula:
  $NTM = \sum_{n=1}^{N} \sum (M_{post},\ M_{query},\ M_{reply})$
- Average waiting time(AWT): The minimum time period in seconds, averaged over all the clients, starting from the sending of a ServiceQuery message and ending with the receiving of a ServiceReply message. A client which cannot find the service after r rounds has the waiting time r times round interval seconds. It is calculated using the following formula:
  $$AWT = \frac{\sum_{n=1}^{N_c} min(t_1, t_2, \ldots, t_m)}{N_c}\ (s)$$

For a Post-query strategy, good performance means a high success rate, a short average waiting time and a low number of transmitted messages.

5.2. The simulation results of the greedy Post-query strategy

We choose this strategy to show our methodology of the performance evaluation. The other four Post-query strategies performance will be summarized in Table 1. The greedy strategy requires that all servers post their services to all the nodes, and that all clients query their services from all the nodes in the network in each round. To achieve this goal, a flooding algorithm is used. The idea of a flooding algorithm is that each node tries to forward every message to every one of its neighbors. This results in every message eventually being delivered to all reachable parts of the network. However, some precautions have to be taken to avoid duplicate deliveries and infinite loops. Figures 2 and 3 show that this strategy can achieve a high SR of 100% when combined with the DSR and DSDV protocols, which means all the clients in the network have successfully located their desired services. The flooding algorithm used in this strategy results in a large number of nodes being posted to or queried from within a low number of rounds. The higher the DR of the network is, the lower the SR of this strategy has. As the DR increases, the DSR protocol has more routing overhead, and the DSDV protocol has a lower packet delivery ratio [3] which result in a lower SR. When this strategy is combined with the DSR protocol, it has a lower

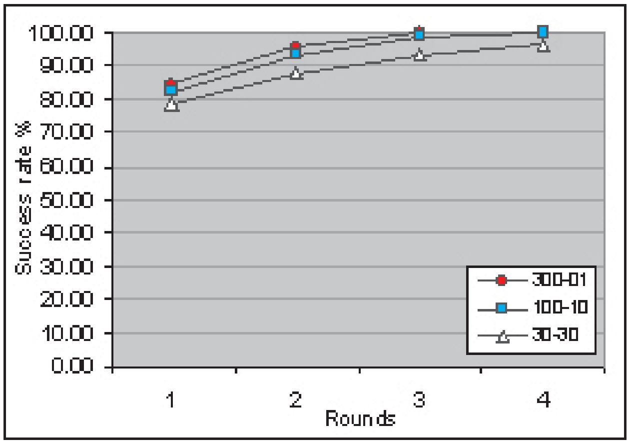

Figure 2. The greedy strategy with DSR success rate

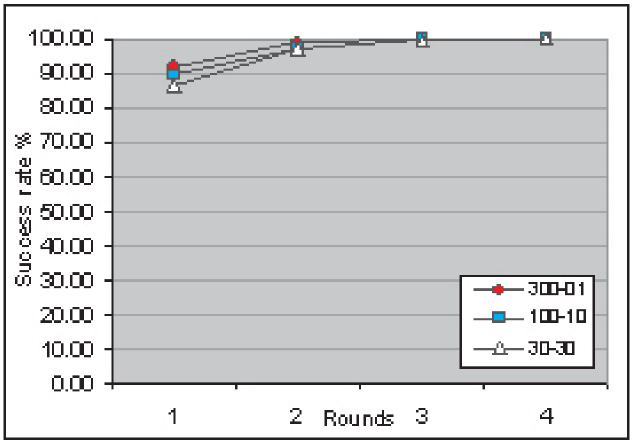

Figure 3. The greedy strategy with DSDV success rate

SR than the one when combined with the DSDV protocol. The flooding algorithm requires the underlying routing protocols to accommodate the heavy routing demands. In such circumstance, the DSR protocol is aggressively adopted and it has more routing overhead than the DSDV protocol. Figures 4 and 5 show the relationships between the NTM and rounds when combined with the DSR protocol and DSDV protocol. We could see that this strategy is costly in terms of NTM. As the DR becomes higher, the NTM becomes lower. When this strategy is combined with the DSR protocol, it has a lower NTM than the one when combined with the DSDV protocol. Figure 6 presents the relationships between AWT and different DRs of the greedy strategy when it is combined with the DSR protocol and DSDV protocol. As the DR increases, the AWT becomes longer. With the DSR protocol, this strategy has longer AWT than the one when with the DSDV protocol. As an on-demand routing protocol, the DSR has to accommodate heavy routing demands of establishing routes from all nodes to all nodes of the network, which is very time-consuming. As a table-driven routing protocol, in the DSDV protocol, a route is prepared before messages are sent, which will save time. Table 1 illustrates the performance comparisons of the five Post-query strategies when combined with the DSR protocol and DSDV protocol. AWT is in seconds. We have five Post-query strategies, each of which can be combined with two types of ad hoc routing protocols. Each strategy, combined with a routing protocol, has three different dynamic ratio network scenarios. Thus, we have 30 combinations, so when we want to adopt a Post-query strategy to an ad hoc network, this table can serve as a reference.

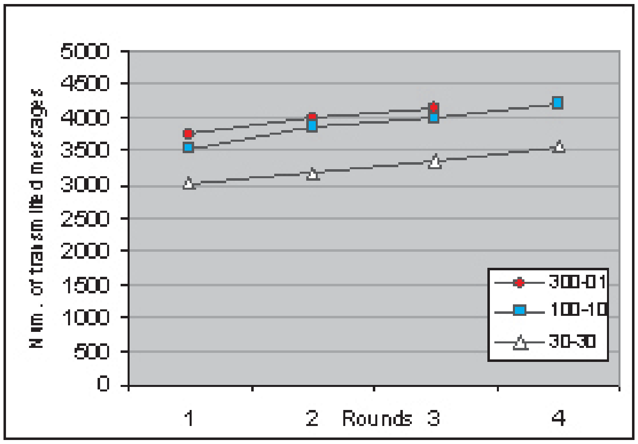

Figure 4. The greedy strategy with DSR num. of transmitted messages

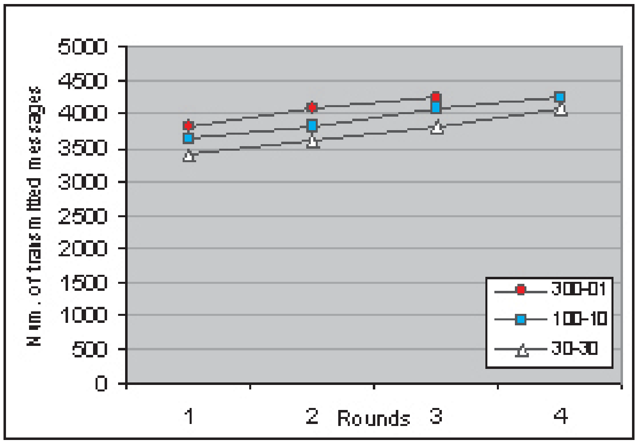

Figure 5. The greedy strategy with DSDV num. of transmitted messages

Before we select a strategy, some analysis should be conducted on the size of the ad hoc network, the available ad hoc routing protocols, the dynamic ratio, the available support for broadcast and unicast communication, the amount of available bandwidth and the specific requirements for the three performance metrics. Based on the parameters we chose, the following conclusions could be drawn.

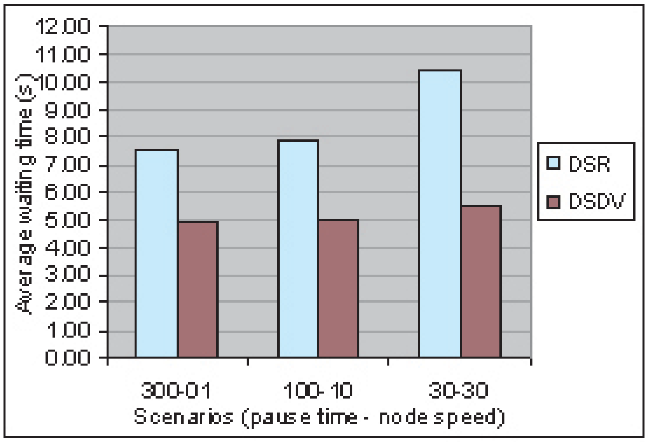

Figure 6. The greedy strategy with DSR and DSDV average waiting time

- The greedy strategy consumes significant network resources to achieve a high SR. It has the highest NTM and shortest AWT of all the five strategies. Therefore, this strategy could serve as a reference to compare other strategies. When requiring a service discovery strategy with a high SR and a short AWT, regardless of NTM, the greedy strategy combined with the DSDV protocol are a good choice.
- The conservative strategy can also achieve a high SR with a low NTM and a short AWT in a high DR ad hoc network. In a low DR ad hoc network, the SR is low because numerous nodes may not be posted to or queried from at all. For a high DR ad hoc network, this strategy combined with the DSR or DSDV protocol are the best two choices.
- The incremental strategy can achieve a high SR with a low NTM, while it has the longest AWT of all the strategies. When requiring a high SR with usage of a low NTM, regardless of AWT, this strategy combined with the DSR protocol will satisfy these requirements.
- In the uniform memoryless strategy, if the sizes of posting and querying sets are relatively low, the NTM is low and in each round almost remains constant so it uses a relatively low amount of network bandwidth. For an ad hoc network which can only spare a dedicated bandwidth for a service discovery strategy, this strategy combined with the DSR/ DSDV protocols can meet this requirement.
- The with memory strategy aims to improve SR. It can achieve a high SR with a low NTM and a long AWT. Comparing this strategy with the uniform memoryless strategy, the SR of this strategy is higher. But the trade-off is that each node is required to build a cache of about 0.2K to store the IDs of previously visited nodes. When requiring a high SR and uses a low NTM and can bear with more memory usage regardless of the AWT, the with memory combined with the DSR protocol are the most suitable choice.

Strategy	RP	DR	Max. SR	Total NTM	AWT
Greedy	DSR	8.06	100%	16215	7.50
		61.48	100%	15601	7.85
		190.72	98.75%	13100	10.42
	DSDV	8.06	100%	16557	4.93
		61.48	100%	15777	5.03
		190.72	100%	14887	5.46
Conservative	DSR	8.06	82.25%	2909	36.55
		61.48	99.75%	5455	20.12
		190.72	100%	3620	8.42
	DSDV	8.06	81.5%	2122	37.62
		61.48	99.5%	4519	27.81
		190.72	100%	3408	10.09
Incremental	DSR	8.06	96%	2900	79.72
		61.48	94.75%	2894	81.32
		190.72	96%	2899	83.76
	DSDV	8.06	97%	2898	83.23
		61.48	87.75%	2856	90.80
		190.72	81.75%	2828	107.15
Uniform memoryless	DSR	8.06	89%	2602	54.55
		61.48	92.5%	2610	54.30
		190.72	87.5%	2602	57.72
	DSDV	8.06	88.25%	2596	59.27
		61.48	83.25%	2577	64.57
		190.72	75.25%	2555	87.64
With memory	DSR	8.06	100%	2845	46.96
		61.48	100%	2833	47.29
		190.72	100%	2832	53.05
	DSDV	8.06	100%	2816	51.89
		61.48	98.5%	2741	56.12
		190.72	97.75%	2686	77.27

Table 1. Performance comparisons of the five Post-query strategies when combined with the DSR protocol and DSDV protocol

6. Conclusion

In this paper we carried out simulations and performance evaluation of five Post-query strategies combined with the DSR protocol and DSDV protocol. Our main conclusions are as follows: The greedy strategy achieves a very high success rate with the lowest average waiting time and highest number of transmitted messages of all the strategies. The conservative strategy can only achieve a high success rate in a high dynamic ratio ad hoc network. The incremental strategy can also achieve a high success rate with a low number of transmitted messages, while it has the longest average waiting time of all the strategies. The uniform memoryless strategy uses a relatively low amount of network bandwidth if the sizes of the posting and querying sets are relatively low. The *with memory* strategy can achieve a high success rate, while each node builds a cache of previously visited nodes. Compared with the uniform memoryless strategy, the success rate of this strategy is improved by 10.34% when combined with the DSR protocol, and by 16.5% when combined with the DSDV protocol.

In the future we could dig deeper to evaluate how their performance is influenced if we vary the sizes of posting and querying sets and round interval. And it would be interesting to study networks in which the services have different probabilities. We may choose other ad hoc routing protocols such as AODV [19], TORA [16][17], OLSR [9] and ZRP [6]. Also we could extend our research according to the heuristic model proposed by Koodli and Perkins [11]. Their approach is to add extensions to suitable ad hoc network routing protocols, in order to find the services and routes to these services at the same time.

7. Acknowledgment

The authors graciously acknowledge the financial support received from the following organizations: Natural Sciences and Engineering Research Council of Canada (NSERC) and Mathematics of Information Technology and Complex Systems (MITACS).

References

[1] M. Barbeau, E. Kranakis, Modeling and Performance Analysis of Service Discovery Strategies in Ad Hoc Networks, Proceedings of International Conference on Wireless Networks (ICWN), Las Vegas, Nevada, 2003.

[2] Bluetooth, Specification of the Bluetooth System, Specification Volume 1, 2001, Specification Volume 2, 2001. http://www.bluetooth.com

[3] J. Broch, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), pp. 85-97, Dallas, TX, October 1998.

[4] K. Fall, K. Varadhan, editors. *ns* notes and documentation. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997. http://www.isi.edu/nsnam/ns/.

[5] E. Guttman, C. Perkins, J. Veizades, M. Day, Service Location Protocol, Version 2, IETF RFC 2608, June 1999.

[6] Z. Haas, M. Pearlman, The Zone Routing Protocol (ZRP) for Ad hoc Networks, IETF Draft - Mobile Ad hoc NETworking (MANET) Working Group of the IETF, November 1997.

[7] S. Helal, N. Desai, V. Verma, C. Lee, Konark - A Service Discovery and Delivery Protocol for Ad-hoc Networks, Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans, March 2003.

[8] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, A. Schade, DEAPspace - Transient Ad-Hoc Networking of Pervasive Devices, Computer Networks, vol. 35, pp. 411-428, 2001.

[9] P. Jacquet, P. Muhlethaler, A. Qayyum, Optimized Link State Routing Protocol, IETF Draft, August 2001.

[10] D. Johnson, D. Maltz, Dynamic Source Routing in Ad-Hoc Wireless Networks, in Mobile Computing, 1996

[11] R. Koodli, C. Perkins, Service Discovery in On-Demand Ad Hoc Networks, Internet-Draft, October 2002.

[12] E. Kranakis, P. Vitányi, A Note on Weighted Distributed Match-Making, in Mathematical Systems Theory, Vol. 25, 123-140, 1992.

[13] D. Maltz, The Effects of On-Demand Behavior in Routing Protocols for Ad Hoc Networks, IEEE Journal on Selected Areas in Communications Special Issue on Mobile and Wireless Networks, pp. 1439-1453, August 1999.

[14] S. Mullender, P.Vitányi, Distributed Match-Making, Algorithmica, Vol. 3, pp. 367-391, 1988.

[15] M. Nidd, Service Discovery in DEAPspace, IEEE Personal Communications, August 2001.

[16] V. Park, M. Corson, A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proceedings of the IEEE Conference on Computer Communications, pp. 1405-1413, Kobe, Japan, April 1997.

[17] V. Park, M. Corson, A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing. Proceedings of IEEE Symposium on Computers and Communication'98, pp. 592-598, Athens, Greece, June 1998.

[18] C. Perkins, P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, Proceedings of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, London, UK, pp. 234-244, August 1994.

[19] C. Perkins, E. Royer, Ad-hoc On-Demand Distance Vector Routing, Proceedings of Second IEEE Workshop. Mobile Computing System and Applications, February 1999.

[20] Sun Microsystems, Technical White Paper: Jini Architectural Overview. http://www.sun.com/jini/, December 1999.

[21] B. Tuch, Development of WaveLAN, an ISM Band Wireless LAN. AT&T Technical Journal, 72(4), pp. 27-33, July/August 1993.