

An Efficient and Robust Service Discovery Protocol for Dynamic MANETs

Mohammad Nazeeruddin, Gerard Parr, Bryan Scotney

School of Computing and Information Engineering
University of Ulster, Coleraine, Northern Ireland.
Email:nazeer, gparr, bryan@infc.ulst.ac.uk.

Abstract. Automatic service discovery is essential for the usability of self-configuring Mobile Ad Hoc Networks (MANETs). Existing service discovery solutions are either directory based or directory-less. The directory based approaches are fast and reliable but have the complexity of maintaining directories in the dynamic MANET environment. On the other hand, the directory-less approaches use broadcasts which are very expensive in MANETs. This paper proposes and evaluates a new service discovery protocol called MANET Service Location and Discovery (MSLD), which is integrated with a stateful auto-configuration protocol called DHAPM. Because of this integration, MSLD reuses the robust directory structure of DHAPM, which allows MSLD to acquire the advantages of the directory based service discovery protocols without any additional complexity of directory maintenance. The performance evaluation shows that MSLD has low service discovery latency and high service availability with minimum communication overhead.

1 Introduction

The tremendous growth of portable handheld devices (e.g. cell phones, iPAQs, PDAs) with basic networking capabilities (Bluetooth, GPRS, 802.11) has opened new research directions in the area of networking [1]. Mobile Ad hoc Networks (MANETs) provide the necessary capabilities to form a rapid, self-organizing, infrastructureless network among mobile wireless peers without any prior planning. One of the key reasons behind the formation of a MANET is to effectively utilize the services provided by the individual peers (nodes or hosts) in the network. Nevertheless, to effectively exploit the services provided by the peers, a simple robust mechanism for discovering services is needed. Service location or discovery protocols offer a convenient and effective framework for the networked hosts to advertise their services to other hosts and access the information about the existence, location and configuration details of the other networked services [2]. Services on a device include accessible software components, hardware components and data which other devices may need. For instance, in military communications data from several heterogenous devices, offered by different services, should be integrated to discover meaningful trends [1].

Because of several inherent limitations of mobile devices and wireless networks, enabling service discovery is a challenging task. Thus, the MANET Ser-

vice Discovery (SD) protocols should have the following features. (i) Low overhead: mobile nodes are limited-resource devices and hence MANET protocols should use minimal device resources. (ii) Robustness: MANET protocols should tolerate intermittent network connectivity (which arises because of node mobility) and device failures. (iii) Versatility: any network enabled device can be part of a MANET. So the SD protocols should be able to work with a wide variety of devices. (iv) Scalability: MANETs have diverse application ranging from small personal networks to diverse military networks. Thus, the MANET protocols should work seamlessly in different networks.

In this paper we propose a service discovery protocol for MANETs and show its integration with DHAPM [11] [12], a dynamic host auto-configuration protocol for MANETs. The proposed service discovery protocol is lightweight, robust and can adapt dynamically to the network conditions.

2 Related Work

Theoretically the service discovery problem can be solved by using either a directory based (centralized) approach or a directory-less (distributed) approach.

In a directory based approach, a directory of the available services in the network is maintained by a single host or a group of hosts. Service agents (Servers) register their services along with the service access details in the directory and clients query the directory to obtain the service descriptions.

In the directory-less approaches, as there are no centralized directories, the service messages should to be distributed to all nodes. Depending on distributed message type, the service discovery can be done in two modes. (i) Push mode: servers broadcast (or multicast) frequent service advertisements to all hosts and clients passively cache the services and select the service which is of interest. (ii) Pull mode: clients broadcast service requests and servers that offer the requested service reply to the request.

The traditional service discovery protocols designed for wired networks (like SLP, Jini, UDDI), do not account for the latency and packet loss issues associated with the several intermediate wireless links connecting multi-hop nodes and intermediate node mobility [8]. Moreover, the traditional protocols do not take account of the device heterogeneity and resource constraints. In general, there has been an agreement among researchers that the traditional protocols cannot be used in MANETs [3, 8].

Some new SD protocols for MANETs were developed specifically for a physical layer technology, like Bluetooth's service discovery protocol and IrDA's information access protocols [3]. Some other protocols were developed as part of high-level distributed application technology (e.g. SD mechanisms defined in JXTA and OSGi) and other approaches (e.g. INS and one.world [3]) developed as part of complex pervasive computing architecture. Though these approaches are efficient in the specific environment for which they were developed, they are not suitable for general heterogenous MANETs.

The specific ties to a particular environment can be eliminated by designing SD solutions at the application layer of the protocol stack. As MANETs

are infrastructure-less networks, the obvious choice for the SD protocols is a directory-less architecture and hence several new directory-less protocols (Konark, DEAPspace, PDP [3], etc) have been proposed for MANETs. The common problem in the directory-less (pull-based or push-based) approaches is that they employ broadcasting or multicasting both of which add substantial communication overhead [7]. Also, these approaches suffer from scalability problems in larger networks [13].

The communication overhead can be minimized by using distributed service directories. In [7] a service discovery solution is proposed based on the service directories which are deployed dynamically. It was shown that this type of solution minimizes the generated traffic. However, the directory based approaches have an additional overhead of selecting the directory servers and maintaining the directories.

The proposed MANET Service Location and Discovery (MSLD) protocol also employs distributed directories to achieve robustness against directory failures. MSLD is integrated with DHAPM and uses the DHAPM address agents (AAs) to perform the role of service directories. Thus, MSLD does not have any additional overhead of selection and maintenance of directories while having low communication overhead and the robustness offered by the directory-based approaches.

3 Integration with DHAPM

Dynamic Host Auto-configuration Protocol for MANETs (DHAPM) [12] is a stateful host auto-configuration protocol based on multiple dynamically selected Address Agents (AAs), which deliver the important functions of auto-configuration. Each Address Agent (AA) manages a disjoint block of sequential IP addresses which it can assign to a new node without consulting any other AA in the MANET [10]. Each AA maintains two tables - an address table and an agent table. The Agent table keeps the information about all other AAs in the MANET. The Address table keeps a record of all the assigned IP addresses and the details of nodes to which they are assigned. Node details include hardware address, node status, and a boolean bit [12]. AAs periodically synchronize with the other AAs by sending incremental address table updates.

DHAPM has robust mechanisms for initiating and maintaining the AAs. The AAs in DHAPM can be easily extended to support service discovery tasks. The following are the required amendments to DHAPM for providing Service Discovery related tasks:

- Service directory: A service directory holds a list of available services including the details of the service providers and service attributes. DHAPM already maintains hostnames and IP addresses in the address table for auto-configuration and name resolution¹. So the same table can be supplemented with an extra "services" column as shown in Table 1. Since the address table

¹ Name resolution related tasks are provided by a separate protocol called MNS.

is sorted with respect to IP addresses, the whole table should be searched to find all the matching services for a given requested service type, which is inefficient. For this reason a separate sorted table called "service directory" is maintained. The service directory is derived from the address table and sorted with respect to service names (types) for efficient service lookups.

- A mechanism for querying/administering database: Once a service directory is available, DHAPM should provide a mechanism to query and administer this database. To carry out this task, a new module called MANET Service Location and Discovery (MSLD) is added to the DHAPM protocol. MSLD is described in Section 4.

Table 1. Modified Address Table Accommodating Service Information

IP	MAC	Name	Service(s)	Status	Suitable for AA	ALT
1	Mac1	PC1	srv1@attr1=val1;attr2=val2	0	TRUE	3600
2	Mac2	PDA2	srv1@attr1=val1;attr2=val2	4	TRUE	600
3	Mac3	PC3		1	FALSE	200
.....
253	Mac253	LT	srv10@attr1=val1, srv13@attr1=val1	0	TRUE	500
254	Mac254	PDA	srv4attr1=val1;attr2=val2;attr3=val3	1	TRUE	400

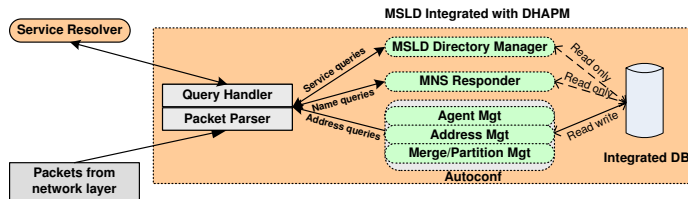


Fig. 1. MSLD Integrated with DHAPM

The integration of SD protocol in DHAPM brings the following benefits:

- MSLD has no additional overhead of Directory Manager (DM) selection and maintenance while having the benefits of a distributed directory based approach.
- MSLD service directory can use the existing information from the address table instead of re-requesting from the services.
- Basic service registration and service deregistration can be done during the address allocation and address relinquishment process respectively.
- MSLD module runs as a separate thread of DHAPM process instead of an independent process which saves the system resources.

The integrated architecture of DHAPM with service discovery module is shown in Fig. 1. DHAPM query handler receives all queries including service discovery and name resolution queries. Depending on the type of a query, the DHAPM query handler forwards it to an appropriate module. The integrated database (address table, agent table and service directory) is managed by autoconf module and the other modules have a Read-Only (RO) access to the database.

4 MANET Service Location and Discovery (MSLD)

MSLD has three components: Service Resolver (SR), Service Manager (SM) and Directory Manager (DM).

Service Resolver (SR): Service resolver is a software routine which works on behalf of the user applications to acquire information about the available services with their attributes and configuration details. SR retrieves the service information in two modes depending on whether a DM is accessible or not: (i) Unicast mode (U_C mode): SR operates in this mode when it has information about one or more DMs. In this mode service discovery requests (SDQs) are unicasted directly to a DM and DM unicasts the service discovery replies to the requesting SR. By default SR operates in this mode. (ii) Broadcast mode (B_C mode): SR temporarily switches to this mode when it cannot communicate with any DM. In B_C mode SR broadcasts SDQs in the MANETs. If any device is offering the requested service then it unicasts a service discovery reply (SDY) with the service details to the requesting SR. Moreover, DMs also reply to the broadcasted requests indicating their availability for any matching entries from their service directory. If SR receives a reply from a DM, then it switches back to U_C mode.

SR caches the retrieved service details for a certain lifetime called Cache Life Time (CLT). If the same request is issued again before the elapse of CLT seconds, then SR replies immediately from the cache instead of sending a service discovery request (SDQ) to a DM. Once the CLT of a service expires, the service information becomes obsolete and is deleted from cache.

Service Manager (SM): The service manager manages all services offered by a device. It serves two purposes: (i) communicates with a DM on behalf of all services² available on the devices; the services register their information (service attributes and configuration) directly with the local SM instead of independently registering with a DM; (ii) replies to the broadcasted SDQs, which is essential for the smooth functioning of service discovery protocol in the absence of DMs.

Using a SM as an intermediate broker not only optimizes the communication overhead but also relieves service developers from the burden of writing additional code to interact with DMs.

Directory Manager (DM): A MSLD directory manager also performs two tasks: (i) collects and maintains the information about all the services available in the MANET in the form of a service directory; (ii) resolves the received service discovery requests using the service directory.

The DM receives SDQs from either a local SR or remote SRs running on the other hosts. Whenever DM receives a SDQ, it just checks its service directory for the requested service type. If it finds one or more entries then it returns the matching set. Otherwise it returns a null set with an error code. When the DM

² services and service agents (SAs) are interchangeably used in this document

is bombarded with several simultaneous SDQs, it buffers the SDQs and serves each request using a First In First Out (FIFO) policy. If the buffer limit, which is determined based on the individual DM capacity, is exceeded then the DM drops the new requests.

4.1 Service Management and Lookup

This subsection describes the details about the MSLD service directory management and service lookup.

Service Naming Convention: Any networked service may be encoded in a service Uniform Resource Locator (URL). MSLD employs compact service URLs tailored for MANETs using similar syntax and semantics defined in [9]. Having a compact service URL not only optimizes the storage requirements of SD but also saves the energy needed while transmitting and receiving service URLs. A service URL of MSLD protocol can be encoded in the following form:

< srvttype >: // < addrspec >; < attrlist >.

The service type (*< srvttype >*) describes the type of the service. It contains an abstract name (e.g. printer) and is sometimes supported by a concrete type (printer:lpr) [9]. An address specification (*< addrspec >*) in the service URL is the hostname or IP address of the service provider. An attribute list (*< attrlist >*) is a string containing all the attributes of a service separated by semicolons. Each attribute and value are expressed as "*attribute = value*". CLT is also added as an attribute. The following are some example service URLs:

printer://host1.edu;duplex=TRUE

videochat://host2;authentication=KERBEROSV4;speed=5fps

It should be noted, however, that MSLD is not tied to any particular service description syntax, and any other service description formats can be used instead of the proposed format.

Service Registration: Whenever a service is started on a device the corresponding service agent (SA) registers itself with the local Service Manager (SM). The SM coordinates the service registrations with a DM.

The initial service registration is done as part of the new address request process. Whenever a new node requests an address, it also informs the AA³ about the services hosted by it (including the *< srvttype >* and optional *< attrlist >*). The address specification part of the service URL (*< addrspec >*) is omitted because AA can fill that part itself. AA acknowledges the service registration when it actually assigns an address to the new node. Each registered service has a lifetime, which defines how long the service entry will be valid. If the SA does not specify a lifetime then the address lease time (ALT) of the host will be used as its default lifetime. Services should be renewed (or re-registered) before the service lifetime expires.

A node can also register its services any time after the initial registration using a *SrvReg* message and DM acknowledges the registration using a *SrvRegAck*

³ It should be noted that DM is an integral part of AA. For the sake of clarity the term AA is used whenever referring to the specific autoconf tasks and the term DM is used when referring to the service discovery tasks.

message. In addition, devices may need to change the registered service information (service attributes) from time to time. This can be done by sending the updated service information to its DM using a *SrvUpd* message. DM replaces the existing service details with the new details and acknowledge the updates using a *SrvUpdAck* message.

Service Lookup: Whenever a service discovery request (SDQ) is received from user application, Service Resolver (SR) first checks whether the requested service query has some matching entries in the local MSLD cache. If some matches are found with valid CLT then the SR immediately returns the matching entries to the application with an indication that the matching set is from the local cache. If the application is not interested in the cached entries, it may re-request SR to query a DM for the updated service information.

If no matches are found in the local cache or if the application requested specifically to query a DM, then the SR constructs a SDQ message, forwards the SDQ message to its DM (AA), and starts a SDQ Timer. If a service discovery reply (SDY) is received from the DM, then SR extracts the results from the reply message and forwards them to the requested application. In case the received SDY contains a null set, SR returns an appropriate message to the application. Conversely, if no response is received from the DM and the timer expires then SR re-sends the request to its DM until a response is received or the number of retries exceeds r_{min} . If there is no response even after r_{min} retries then the resolver forwards the request to the DM_{pid} ⁴ (AA_{pid}) with a probable departure notification of its DM (AA) and restarts the SDQ Timer. If the DM_{pid} also does not respond then the resolver switches to B_C mode (i.e. directory-less mode).

In B_C mode SR broadcasts SDQs so that service managers (SMs) can directly reply to the SDQs. When the SR first switches from U_C mode to B_C mode, it broadcasts the old SDQ only one time and for the subsequent queries SR retries a request for r_{min} times. If the broadcasts also fail to elicit any responses then the SR returns an error message to the application and terminates the request process. If SR receives a reply from a DM at any point in time then it switches back to U_C mode to reduce the excessive overhead due to the broadcasts.

Service Deregistration: When a node intends to leave the MANET it sends an address relinquish message (*AddrRelReq*) to its AA. After receiving this message from the departing node, the AA marks the nodes's record, including any services offered by the departing node, as invalid. In the case of abrupt node departures, records will be automatically deleted after the address lease time (ALT) of the host expires. This task is handled by autoconf module, so there is no additional overhead on MNS. Also, a service agent can deregister any of its services at any time using a *SrvDeReg* message.

5 Mathematical Analysis

In this section the communication overhead of different alternative approaches is compared with MSLD protocol.

⁴ AA_{pid} is the node which generated MANET PID. For details refer to [10]

Message overhead of the directory-based approaches: An ideal directory based approach has a directory and all devices in the network register their services in this directory. The directory announces its presence periodically by flooding announcement messages. Clients unicast the SDQs to the directory and the directory unicasts the replies to the clients. Assume N_χ is the average number of SDQs, N_{reg} is the average number of service registrations, and N_{cng} is the average number of service updates during the each directory announcement period (T_{ad}). Then the message overhead incurred during T_{ad} is:

$$B_C + 2 * N_\chi * U_C + 2 * N_{reg} * U_C + 2 * N_{cng} * U_C \quad (1)$$

where B_C and U_C is the message overhead of a broadcast and a unicast message respectively.

Message overhead of the directory-less approaches: In the push-based directory-less approaches the service agents periodically announce their services and clients cache the announced service information. If N_{sp} is the average number of service providers then the message overhead incurred during T_{ad} is:

$$N_{sp} * B_C \quad (2)$$

In the pull-based approaches clients flood the SDQs and service agents unicast the replies. If α is the average number of service agents providing the similar services then the message overhead incurred during T_{ad} is:

$$N_\chi(B_C + \alpha * U_C) \quad (3)$$

Message overhead of the MSLD: MSLD maintains a distributed directory, which is periodically synchronized to achieve robustness against node failures. As MSLD is integrated with DHAPM, directory synchronization is done automatically as part of AA synchronization. Only the AA synchronization packet will have extra *service* data. Moreover, the initial service registration is done as part of the address request process. Only the registration of services which started after the initial service registration and service updates will be sent separately. If ζ is the directory synchronization overhead and N_{reg}^- is the average number of late service registrations during the period (T_{ad}) then the message overhead incurred during T_{ad} is:

$$\zeta + 2 * N_\chi * U_C + 2 * N_{reg}^- * U_C + 2 * N_{cng} * U_C \quad (4)$$

Comparing Eqs. 1 and 4, as $\zeta \ll B_C$ and $N_{reg}^- \leq N_{reg}$, it is evident that the overhead of MSLD (integrated with DHAPM) is always less than directory based approaches.

In the case of directory-less approaches, both push and pull based approaches employ flooding for SD which is very costly in MANETs. Although the overhead of flooding can be minimized by using the restricted broadcast (restricting broadcast to few hops), still its overhead is much greater than a unicast message [6]. Thus, comparing Eqs. 2 and 3 with Eq. 4, it is evident that MSLD overhead is

less than both the push and pull based directory-less approaches.

Comparison with routing-based approaches: It has been argued in the literature that the service discovery message overhead can be reduced by piggybacking the service discovery messages with the underlying reactive protocol messages. It was demonstrated in [5] through extensive simulations that pull-based approaches minimize the overall broadcasts when SD messages are integrated with the reactive routing protocol messages. This sub-section analyzes the overhead of MSLD and directory-less approaches when such an optimization technique has been supported by the underlying routing protocol.

As the forward and the reverse routes to the service provider are established during the service discovery process itself, the overall routing and SD message overhead per SDQ will remain the same in directory-less approaches and is given by $B_C + \alpha * U_C$. However, in MSLD the overall overhead will be different depending on whether a route is already available or not. The following three cases are possible in MSLD:

- Case 1: when the route to any DM is not known: in this case, the client sends the route discovery broadcast integrating SDQs. The client receives unicast replies containing service information from the DMs and service managers. So the overall overhead is similar to the pull-based approach ($B_C + \alpha * U_C$).
- Case 2: when the route to a DM is known: in this case, the client unicasts a SDQ to the DM and the DM unicasts the reply. Then the client selects a service provider (SP) and contacts it. If a route to the SP is already known then the overall overhead per SDQ is: $\zeta/N_\chi + 2 * U_C + 2 * N_{eng} * U_C/N_\chi \approx 2 * U_C$ (for large values of N_χ). This implies that MSLD still has lower overhead.
- Case 3: when the route to DM is known and SP route is unknown: in this case, the additional route discovery overhead excluding SP contact overhead ($B_C - U_C$) should be added to the overhead in case 2. Therefore, the overall overhead is $B_C + U_C$. Comparing with the pull-based system overhead (Eq. 3), the overhead of MSLD is lower only if $\alpha > 1$. Otherwise the pull-based approach has better performance. Nevertheless, when the requested service is not available in the network (i.e. $\alpha = 0$), MSLD needs only $2 * U_C$ and pull-based approach needs $r * B_C$. This means MSLD is more efficient.

From the above analysis, it is evident that the pull-based approaches integrated with a routing protocol may have lower overall communication overhead only in case 3 when $0 < \alpha < 1$. In all the other cases MSLD has better or similar performance than the pull-based approaches. In short, MSLD is still an efficient solution even when the service discovery messages are piggybacked on the routing messages.

6 Simulation Based Analysis

From the mathematical analysis it is evident that MSLD has lower communication overhead than the alternative approaches. In this section, we evaluate

MSLD performance based on two other performance metrics (service discovery latency and service availability) using the ns-2 simulator. All simulations were executed for a duration of 300 seconds. We considered 50 mobile terminals with transmission range of 250m randomly distributed over the simulation surface. To simulate dynamic MANETs we used the random way point mobility model with max speed of 4m/s, min speed of 2m/s and zero pause time. In the simulations, all the service discovery requests are uniformly distributed over 150 seconds. All the nodes in the MANET generate an equal number of requests.

The two service discovery architectures simulated are MSLD and directory-less (distributed) architectures. The primary reasons behind the simulation of directory-less architecture are (i) it is an obvious alternative to MSLD; (ii) MSLD switches to directory-less mode in the absence of DMs.

Service Discovery Time (Latency): The service discovery latency of a protocol is the time needed to resolve a service type to a valid service binding and contacting the service agent successfully through the obtained service binding.

To evaluate the latency, we have executed several simulation scenarios where clients are 1-6 hops away from the DM and server node. For each hop the mean (μ) and standard deviation (σ) of service discovery resolution time, service agent contact time and total binding time were calculated. The obtained results are summarized in Tables 2 (a) and (b) for MSLD and directory-less protocols respectively. Notice that latency increases with the increase in the distance (number of hops). This is because of the accumulated packet forwarding delays at the intermediate nodes. However in the case of MSLD, the total latency is less than 50ms even if the client is 6 hops (around 1500m) away from both DM and SP which is negligible. On the other hand, the directory-less approaches have higher latency because of the broadcasting delays.

Table 2. Latency (ms) Vs Hops

(a) MSLD						(b) Directory-less					
Hops	μ_{sdy}	σ_{sdy}	μ_{sp}	σ_{sp}	μ_{total} σ_{total}	Hops	μ_{sdy}	σ_{sdy}	μ_{sp}	σ_{sp}	μ_{total} σ_{total}
1	3.26	(0.25)	3.43	(0.22)	6.69 (0.34)	1	8.24	(3.50)	4.54	(2.25)	12.78 (4.31)
2	6.73	(0.37)	6.87	(0.35)	13.61 (0.52)	2	14.97	(4.22)	7.73	(0.79)	22.71 (4.10)
3	10.21	(0.44)	10.43	(0.45)	20.64 (0.63)	3	22.74	(5.14)	11.40	(1.30)	34.13 (5.33)
4	13.73	(0.54)	13.95	(0.52)	27.68 (0.73)	4	30.34	(5.93)	14.33	(1.82)	44.67 (6.41)
5	17.27	(0.58)	17.44	(0.60)	34.71 (0.83)	5	37.72	(6.34)	17.49	(0.63)	55.22 (6.42)
6	20.74	(0.63)	20.96	(0.62)	41.70 (0.87)	6	45.38	(7.25)	22.24	(2.69)	67.62 (8.30)

Service Availability: Service availability is one of the important performance metrics to measure the service discovery protocol performance. The service availability of the protocol can be defined as the ratio of successful service bindings to the total number of requests generated. A successful service binding

means the successful resolution of the service type and a successful contact with the server [5].

Table 3 summarizes the failed service discovery queries (F_{sdqs}), failed service bindings ($F_{bindings}$) and service availability of both MSLD and directory-less approaches for different network areas and SDQ rates. Observe from the table that the total number of failures ($F_{sdqs} + F_{bindings}$) increases with the increase in the network size, resulting in a lower service availability. This is because as the network size increases the node density decreases and hence there is an increased chance of node isolation/separation from the MANET.

An important point to be noted from this is that MSLD has minimal (almost zero) service type resolution failures, which shows the significance of maintaining a distributed service directory. However, service binding failures ($F_{bindings}$) are much higher in MSLD. This is because for a successful service type resolution, the client just needs one accessible DM; but the service binding with the server may fail because of node mobility (both server and client mobility) during the time of service registration and at the moment the client application seeks to contact the service agent. The probability of such failures increases when the node density decreases.

On the other hand, in the directory-less approaches the server itself resolves the service discovery request, so there is a low probability that the server will be inaccessible to the client after a successful service type resolution and hence there are fewer $F_{bindings}$. Nevertheless, the initial service resolution is more difficult in the directory-less approaches as it merely depends upon the availability/accessibility of the server providing such a service.

Although there are differences in F_{sdqs} and $F_{bindings}$ values, on average the overall service availability is higher in MSLD than in directory-less approaches.

Table 3. Service Availability Vs Network Size

Network Area (m^2)	$\Sigma SDQs$ (SDQ Rate/sec)	MSLD			Directory-less		
		F_{sdq}	$F_{bindings}$	SA	F_{sdq}	$F_{bindings}$	SA
250x250	2000 (13.33)	0	0	1.0000	0	0	1.0000
	4000 (26.67)	0	0	1.0000	1	0	0.9998
500x500	2000 (13.33)	1	43	0.9780	27	24	0.9745
	4000 (26.67)	2	66	0.9830	56	15	0.9822
750x750	2000 (13.33)	3	99	0.9490	57	25	0.9590
	4000 (26.67)	2	169	0.9573	458	177	0.8413
1000x1000	2000 (13.33)	7	143	0.9250	76	66	0.9290
	4000 (26.67)	4	238	0.9395	513	189	0.8245

7 Conclusion

The existing service discovery protocols including the recent solutions tend to add substantial overhead (either for maintaining directories or for flooding the

messages) which reduce their applicability in resource-constrained MANETs. The solution proposed in this paper - MSLD, overcomes this limitation by employing distributed service directories. The proposed protocol can also be seamlessly integrated with any directory-based protocols to reduce directory maintenance overhead. An example scenario showing the integration of MSLD with DHAPM (a robust auto-configuration protocol previously proposed and evaluated by us) was described in this paper and the efficiency of this integrated protocol was investigated through mathematical analysis and simulations.

References

1. D. Chakraborty, A. Joshi and Y. Yesha, "Integrating service discovery with routing and session management for ad-hoc networks," *Ad Hoc Networks*, Vol 4, Issue 2, March 2006, Pages 204-224.
2. E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol Version 2", RFC 2608, July 1997.
3. C. Campo, C. Garcia-Rubio, A. Marin Lopez and F. Almenarez, "PDP: A light-weight discovery protocol for local-scope interactions in wireless ad hoc networks," *Computer Networks*, In Press, Corrected Proof, Available online 19 January 2006.
4. C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF Mobile Ad Hoc Networks Working Group, IETF RFC 3561, July 2003.
5. P. E. Engelstad, Y. Zheng, "Evaluation of Service Discovery Architectures for Mobile Ad Hoc Networks," the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS05) (2005)
6. L. M. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, *IEEE INFOCOM*, Volume 3, April 2001 Pages:1548 - 1557.
7. F. Sailhan, V. Issarny, "Scalable Service Discovery for MANET," Third IEEE International Conference on Pervasive Computing and Communications, 2005. PerCom 2005, Page(s):235 - 244.
8. C. K. Toh, "Ad Hoc Mobile Wireless Networks. Protocols and Systems", Prentice Hall PTR, New Jersey, 2002, Chapter 12, Pages: 231-242.
9. E. Guttman, C. Perkins, and J. Kempf, "Service Templates and service: Schemes", RFC 2609, June 1999.
10. M. Nazeeruddin, G. Parr and B. Scotney, "Fault-tolerant Dynamic Host Auto-configuration Protocol for Heterogeneous MANETs", Proc. of 14th IST Mobile & Wireless Summit, Dresden, Germany, 19-23 June 2005.
11. M. Nazeeruddin, G. Parr and B. Scotney, "A New Stateful Host Auto-configuration Protocol for Digital Battlefield MANETs", Proc. of IEEE MILCOM, Atlantic city, NJ, 17-20 Oct 2005.
12. M. Nazeeruddin, G. Parr and B. Scotney, "DHAPM: A New Host Auto-configuration Protocol for Highly Dynamic MANETs", Accepted for publication in *Journal of Network and Systems Management*, Oct 2006.
13. J. Tyan and Q. Mahmoud, "A Comprehensive Service Discovery Solution for Mobile Ad Hoc Networks," *Journal of Mobile Networks and Applications*, 10, 2005, pages: 423-434.