International Journal of Education and Science Research
Review
E-ISSN 2348-6457
Volume-II, Issue-6 December- 2015 P-ISSN 2348-1817
www.ijesrr.org Email- editor@ijesrr.org

A Study for the Preference Configuration of MANETS Along With an Evaluation Report on Their Service Location Protocol

Heena
Department of MCA
Chandigarh University
Gharuan

ABSTRACT:
Due to the fundamental role that Mobile Ad Hoc Networks (MANETs)play in future pervasive computing environments, providing service discovery in such networks is essential. Compared to wired infrastructure-based networks, wireless ad hoc networks levy a number of unique challenges to the problem of service discovery.

1.0 INTRODUCTION:

In MANETs, heterogeneous nodes are constantly moving, entering and leaving the network, hence continually changing the network's topology. Ad hoc networks must be fully decentralized with no single controlling entities such as directories and lookup tables. Often nodes cannot communicate directly with each other, forcing routes between nodes to transverse several hops. Finally, mobile devices have limited battery and processing power, and therefore cannot afford to perform complex tasks. It is fundamental that we not only provide a Service Discovery Protocol (SDP) in MANETs, but also highlight its assets and deficiencies. Essentially, the most accurate way to evaluate a protocol is to test it under environments that are as realistic as possible. Each protocol can ultimately out-perform another ina particular networking scenario. This necessitates the availability of reference configurations for a fair comparison of MANET SDPs. Evaluating SDPs using this common benchmark facilitates recognizing the strengths and weaknesses of the protocols in the various realistic networking contexts. Such benchmark also Serves as a common ground for a fair comparison of SDPs

2.0 LITERATURE REVIEW:

The Bluetooth SDP was developed by the Bluetooth Special Interest Group (SIG). The protocol targets Bluetooth capable devices. A client queries services directly
By type and attributes all matching services respond, thus allowing the client to locate services. There are no directories involved due to the ad hoc nature of the environment. Since the protocol does not offer functionality to access these services, other complementing protocols may be used.Service Location Protocol (SLP) [2]: SLP is the IETF solution to service discovery. Services are represented by Service Agents (SAs), while User Agents (UAs) mediate for users and clients. Although directories, rep-resented by Directory Agents (DAs) are optional, they are required for scalability and efficiency. String based querying is used to search for services. Alternatively, services could be browsed. Universal Resource Locators (URLs) are used to access services. Scope awareness is sup-ported, and can be utilized to search for services within a particular scope. Universal Plug and Play (UPnP) [5]: UPnP, initiated by Microsoft, relies on a suite of widely available web protocols and open standards in order to provide a powerful solution to the problem of service discovery. UPnP is based upon the TCP/IP protocol stack. New clients and services joining network acquire an IP address either via a DHCP server or Auto-IP. In UPnP, directories, called control points, are optional. Services announce their presence using the Simple Service Discovery Protocol (SSDP), which in turn uses Hypertext Transfer Protocol (HTTP) via User Datagram Protocol (UDP). Service names and attributes are described using the Extensible Mark-up Language (XML). A URL pointing to the XML files included in the advertisement messages. To invoke and control a service, the Simple Object Access Protocol (SOAP) is used. The General Event Notification

Architecture (GENA) is used to format status messages, which are sent when the service status changes. Clients and di-rectories interested need to register with services to receive these status messages. Moreover, UPnP is capable of providing URLs for user interfaces, allowing users to easily control as well as view the status of service. DEAP space [7]: DEAP space is proposed by IBM Research to address the problem of service discovery in single-hop ad hoc networks. The goal is to minimize the time needed to discover new devices in a network without incurring a significant increase in bandwidth and power requirements. The protocol uses the proactive push method for flooding service announcements. However, each device not only announces the service it provides, but also announces all those it is aware of. This increases responsiveness, since a lost advertisement will not significantly affect the "world-view "that the devices possess. Moreover, a new device joining the network does
not need to wait for all other devices offering services to advertise in order for it to acquire a complete overview of the available services. Konark [11]: Konark, proposed by Helalet. al., is a service discovery and delivery protocol designed for large wireless multi-hop ad hoc networks. The discovery process is completely distributed. Each device includes a Konark Application that facilitates human control of service discovery functions, such as advertising and service invocation. The Konark application sits on top of managers and service registries, where the service discovery process and data structures are managed. The protocol uses XML-based templates to describe services Service invocation is achieved via a micro-HTTP server present on each device, and which is based on SOAP. All service discovery functionalities can be accessed via a set of Application Program Interfaces (APIs), which actual services can easily be equipped with. Koodli and Perkins [12]: Koodli and Perkins proposed the integration of the service discovery problem with the mobile ad hoc network routing proto-cols. Extensions to both table-driven and on-demand routing protocols are added. In the first case, service information can be made available along with the route information stored in the routing tables. In the second case, service discovery uses the same operations used in route discovery by adding a service request extension to the route request packet. Li and Lamont [13]: Li and Lamont proposed a lightweight service discovery mechanism for mobile ad hoc pervasive environments. They integrate service discovery with the routing layer. This integration allows the protocol to automatically identify the appropriate service discovery model for the current network configuration; that is detect whether a directory exists or not, as well as decide whether to pursue active or passive discovery. Service discovery is achieved by extending the messaging mechanism of the Optimized Link State Routing (OLSR) layer. The topology information obtained from the routing layer is used to enhance the protocol's capability in accommodating topology change and improve its scalability for hierarchical networks. MARE [8]: MARE uses a combination of mobile agents and tuple spaces to address the problem of service discovery in highly dynamic ad hoc net-works. The goals of the protocol are to achieve a near consistent view of a constantly changing ad hoc network without reliance on a single central entity, and to reduce the amount of communication between the parties. The authors report results showing that the number of messages exchanged while achieving a consistent view using MARE is much less than that of SSDP and SLP. This is achieved by using agents and moving them to the point of operation rather than transmitting information backend forth across the network. Agents may also aggregate information about few services all at once (e.g. aggregates all information pertaining to all services of the same service provider at once). The author's doac knowledge that the proposed MARE lacks security mechanisms Zhu et al. [14]: Zhu et al. propose another service discovery protocol that is also based on proxies, but customized for ad hoc environments. The protocol solves some of the security and trust concerns that evolve in ad hoc networks when accessing unfamiliar public services. The protocol presumes a non-pure ad-hoc environment, one that has access to other network connections. Services are assumed to be connected to their service providers via the Internet. Clients must communicate with their trusted proxies using the Internet connection that the services possess with their service providers. The Internet is used to setup trust relationships and exchange security keys between the communicating partners while leaving the ad-hoc environment intact.

3.0 PROBLEM STATEMENT:

This work aims to adapt, implement, evaluate, and improve the Service Location Protocol (SLP) Version 2 in the context of Mobile Ad Hoc Networks (MANETs). Additionally, this work aims to develop a framework based on reference configurations by which the performance of Service Discovery Protocols (SDPs) could be realistically evaluated in MANET environments.

4.0 SERVICE LOCATION PROTOCOL FOR MANET:

Service Location Protocol (SLP) is the Internet Engineering Task Force (IETF) standard for enabling network-based applications and users to automatically locate services in Local Area Networks (LANs) with cooperative administrative control. The current Version 2 of the protocol, documented in RFC 2608 [2], was promoted to the Internet Standards Track in June 1999.In SLP, service discovery is achieved using three entities: User Agents (UAs), Service Agents (SAs), and Directory Agents (DAs). Services are represented by SAs, while UAs mediate for users and applications. SAs broadcast service information to DAs (passive discovery), or reply to service requests (active discovery). UAs request service information (active discovery), or listen to service broadcasts from DAs (passive discovery). Directories represented by DAs, are intermediate centralized service information repositories, which cache service information from SAs, and satisfy UAs requests accordingly. Similarly, Das also operate in either an active or a passive fashion. Although DAs are optional, they are required when scalability or efficiency is desired. String based querying is used to search for services. Universal Resource Locators (URLs)are used to access services; they provide users and applications with necessary service information to enable them to directly contact the discovered service. Scope awareness is supported, and can be utilized to search for services within a particular administrative scope.

5.0 ADAPTATIONS TO SLP:

The original Service Location Protocol (SLP) description is documented in RFC 2608 [2]. The original SLP specification is mainly intended to function within LANs under cooperative administrative control. In our work, we have adapted SLP Version 2 to Mobile Ad Hoc Network (MANET) environments; we called the adapted protocol Service Location Protocol for MANET (SLP Manet).SLP Manet is hence not a new protocol, instead it provides a subset of the original SLP specification that is adaptable to MANET environments. As SLP Manet implements all the SLP Version 2 required features described in [2]. All features not implemented in SLPManet were described as optional in the RFC specification of SLP, and were left out because they do not suit MANET environments. The most notable of these excluded features are: Directory Agents (DAs): Directory Agents are centralized stores for ser-vice information. Directory agents could exist in LANs to enhance performance and capability of SLP. DAs do not lend themselves to MANET since such networks must operate without requiring the existence of certain pre-existing and continuously existing nodes. As a consequence of the absence of DAs, only active discovery can take place. Authentication Blocks: Agents are neither configured to generate authentication blocks nor do they verify them. This optional feature is needed to prevent control of services by an adversary. Since security is not the goal of our research, we did not implement this functionality in SLP Manet.

Optional SLP messages: Messages such as Attribute Request and Service Type Request were not implemented. In a MANET context, such optional features add complexity and consume more resources. The focus of outwork is to implement the required core of the SLP protocol, which provides enough features to implement a service discovery mechanism for MANET environments.

Service Agent Advertisement User Agents can not solicit SA Advertisements if they have been configured with a <scope-list>. UAs solicit SA Advertisements only when they are explicitly configured to use User Selectable scopes in order to discover the scopes that SAs support.

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+

| Service Location header (function = SAAdvert = 11) |

+-+
| Length of URL | URL \
+-+
| Length of <scope-list> | <scope-list> \
+-+
| Length of <attr-list> | <attr-list> \
+-+
| # auth blocks | authentication block (if any) \
+-+

The SA responds only to multicast SA discovery requests which either include no <scope-list> or a scope which they are configured to use. The URL is "service: service agent://"<addr> of the SA, where <addr> is the NS-2 address of the SA. The SAA dvert may include a list of attributes the SA supports. This attribute list should be kept short so that the SAAdvert will not exceed the path MTU in size. However, since attributes are not used in SLP Manet, the lengthand string field of the <attr-list> are set to zero and null respectively. The SAAdvert contains no SAAdvert authentication blocks in our implementation since digital signature verification was not implemented in SLP Manet.

6.0 SCOPES:

Scopes are sets of services. The primary use of Scopes is to provide the ability to create administrative groupings of services. A set of services maybe assigned a scope by network administrators. A client seeking services is configured to use one or more scopes. The user will only discover those services which have been configured for him or her to use. By configuring UAs and SAs with scopes, administrators may provision services. Scope strings are case insensitive. The default scope string is "DEFAULT". Scopes are the primary means an administrator has to scale SLPManet deployments to larger networks. SLP Manet messages which fail to contain a scope that the receiving Agent is configured to use are dropped. Every SrvRqst (except for SA discovery requests)and SAAdvert message must include a <scope-list>.Analysis of Existing Caching Approaches We identify three major disadvantages with the caching approaches identified above:

1. Both approaches described involve cross-layering through integrating routing with service discovery, thus the OSI layered approach is sacrificed. Consequently, in [28], the SLP headers must be extended to include next hop information for routing from client to responded nodes.
2. The first approach has the potential drawback of lowering the number of discoverable services as the authors of [28] have identified.
3. The latter approach can also induce increased flooding of messages due to the distribution of service replies. In MANET scenarios where a service's not like to be used repeatedly, or the service advertisement lifetime is short, this method will result in decreased efficiency. The distribution of service replies in this case will only result in useless flooding.

7.0PROPOSED EXTENDED CACHING APPROACH:

We propose a simple caching optimization. All intermediate nodes thatare also participating as clients or User Agents (UAs) in the service discovery process shall cache service information found in the Service Replies (SrvRplys) that they relay. This way, if one of these intermediate UAs wish to use a service that was previously cached, it need not issue a new Service Request (SrvRqst).UAs must not generate SrvRplys for cached service information.

Therefore, in the discovery scenario previously described, and depicted in when C2 requests service X, C1, N1 and N2 will all rebroadcast the SrvRqst as usual? However, when N1 or N2 need service X, they will not send out a SrvRqst since service X is locally cached (assuming that the cached service URL has not yet timed out).The proposed modification to SLPManet:

1. Increases the performance and efficiency of SLPManet. Increased caching at UAs results in reduced flooding of messages since less SrvRqsts need tobe issued. Reduced message flooding translates into less

Network bandwidth consumption similarly, increased cache hits means that the average discovery latency is reduced, and the overall discovery success rate should improve.
2. Does not sacrifice the layered OSI approach to networking. No routing information is needed by the service discovery application hence; SLP Manet headers need not be amended.
3. No extra distribution of information is needed. Service information is learned by intermediate UAs nodes for free. The only cost is that of caching. Therefore, scenarios with short service advertisement lifetimes, or where services are not likely to be re-used will not result in decreased protocol efficiency.
4. The number of discoverable services is not lowered as much as in [27].All intermediate nodes still re-broadcast service requests even if they have cached hits.
5. The modification conforms to the original specification of SLP Manet; UAs issue Srv Rqsts while only SAs respond with Srv Reply. However, a possible drawback to caching is that in the event that a network topology change occurs, a cached service may no longer be the optimal (orclosest) service. then accessing service X provided by S2 is probably more optimal than the already cached service (i.e. service X provided by S1). This is a general problem associated with designs that do not have access to routing information (i.e. non-
Cross layer designs). Therefore, it is essential that the service URL lifetimes are representative of the rate of network topology change.

7.0EVALUATION OF SLPMANET:

We used Network Simulator Version 2 (NS-2) developed by the Virtual Inter Network Test bed (VINT) project at the University of California at Berkley for our simulations. NS-2 is an open-source discrete event simulator targeted at networking research. NS-2 provides substantial support for simulation of Transmission Control Protocol (TCP), routing, and multicast protocols over wired and wireless (local and satellite) networks [32]. Tools for supporting mul- tihop wireless networks with models for physical, data link, and Media Access Control (MAC) layers is provided by the Monarch research group at Carnegie-Mellon University. NS-2 is an object oriented simulator, written in C++, with an Object Tool Command Language (OTcl) interpreter as a front-end. The simulator supports a class hierarchy in C++ and a similar class hierarchy within the OTcl interpreter. The two hierarchies work in harmony with each other. From the user's perspective, there is a direct correspondence between a class in the OTcl hierarchy and one in the C++ hierarchy. Users create new simulator objects using OTcl scripts. These objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the C++ hierarchy. User instantiated objects are mirrored through methods defined in the corresponding C++ class. This split-programming approach and one-to-one mapping between hierarchies allow users to easily configure, setup, and control simulation parameters through OTcl scripts without the need of manipulating and recompiling C++ objects. In NS-2, each mobile node on a flat ground uses an Omni-directional antenna with transmitter and receiver gain equal to 1. The transmitters' and receivers' antennas are 1.5 m high. The wireless interface mimics the commercial 914 MHz Lucent Wave LAN Direct-Sequence Spread Spectrum (DSSS) radio [33]. It is implemented as a shared-media radio with nominal bit rate of 2 Mb/s. The system is assumed to be an ideal one, with system loss equal to 1. The signal propagation model combines both a free space propagation model and a two-reground reflection model. When a transmitter is within the reference distance of the receiver, the free space model with a signal attenuation of 1/r2 is used. Otherwise, the ground reflection model, where the signal falls off as 1/r4, is used. The crossover point is around 86.14 m. The transmitted signal power is around 0.2818 W, while the correct-received signal threshold is around 3.652e- 10 W. Therefore, the transmission range is in the range of 250 m. The MAC layer protocol used is the IEEE 802.11 Distributed Coordination Function (DCF). Each node maintains a network interface priority queue whose job is to queue packets until the MAC layer can transmit them. The queue' sleight is 50 packets. Routing packets possess higher priority, and therefore are always inserted at the head of the queue. In order to implement Service Location Protocol for MANET (SLPManet), it was required to use a MANET routing protocol that supports multicasting or broadcasting as well as unexacting. We used an efficient scalable Broadcast routing protocol for

MANET (BCAST), developed by Kunz [34]. An NS-2 implementation of BCAST is publicly available at [35]. However, BCAST does not support unexacting. SLPManet requires that a Service Agent (SA) uncast Service Replies (SrvRplys) back to the requesting User Agent (UA). In order to work around this obstacle, we made each UA join a BCAST group of its own. SAs would then send Service Replies (SrvRplys) to the one-node BCAST group of the requesting UA. For instance, a UA residing at node 4 would join a BCAST group of its own, the address of the BCAST group is 0xE000004. When the SA receives Service Requests (SrvRqsts) from node 4, they would broadcast their SrvRplys to the single-node group 0xE000004. The penalty of this setup is the exchange of extra routing messages by the BCAST routing agents. Since routing performance and evaluation is not the goal of this work, this will not affect the performance or validity of our results. It is noteworthy to mention that our SLPManet implementation will work with any MANET routing protocol that supports both multicasting and unexacting. The above work-around will only take effect if BCAST was defined to be the underlying routing protocol.

In order to implement our extended caching modification of SLPManet, we made each UA join every other UA BCAST group (in addition to its own BCAST group). Hence, any SrvRply that is being uncast back to the querying UA will also be picked up and cached by the intermediate UAs. We did not impose a limit on the size of the node's cache. However, a URL entry is removed from the cache as soon as it times out. Hence, the number of entries cached at a UA or a SA at any one time was relatively small. SLPManet requires packets to be transported via UDP. Since a UDP agent allocates and sends network packets, all the information needed for application level communication (i.e. needed by the UA and SA of the SLPManet application) should be handed to the UDP agent as a data stream. Unfortunately, the UDP implementation in NS-2 allocates packets that only contain a header stack. Therefore, we had to modify the UDP implementation so that they correctly handle application level data sent to and from the SLPManet application agents. We called the SLPManet-compatible UDP agents UDPSlp.

8.0 EXPERIMENTAL SETUP:

We determined configurations for each of the scenarios composing our benchmark. A test for each of the 10 applications was crafted according to the configurations. NS-2 implementations of the mobility models provided by [29], [30], and [31] were used to generate 10movement.SR nodes (refer to Table 3 on page 67) serving as Service Agents (SAs) are picked randomly from a uniform distribution on [0,NS). Similarly, UR nodes serving as User Agents (UAs) are picked randomly from a uniform distribution on [0, NS). However, a node that has already been chosen cannot be picked again. This signifies that there can be at most 1 UA or 1 SA per node. The rest of the nodes, NS − SR − UR, do not play a role in the service discovery process except relaying packets. The number of Unique Services (US) available in a given scenario is the number of Service Agent nodes less the number of duplicated services, US = SR − SD + 1. Therefore, US − 1 Service Agents offer different services, while SD server agents offer identical services. Research in [29] shows that the initial random distribution of mobile nodes is not representative of the manner in which nodes distribute themselves when moving. There is a high variability in the number of nodes within a mobile node transmission range during the first 600 seconds of simulation. This problem may have severe consequences in some of the mobility models, such as the random waypoint model. In order to solve this problem, we run simulations for 2000 seconds, and discard the initial 1000 seconds of simulation. In fact, the service discovery process does not start until after 1000 seconds of simulation

Time has elapsed. There are a total of 1000 requests per simulation. The inter-arrival time for requests is exponentially distributed in the range [1000, 2000) seconds with a mean at 1360 seconds. Every time a User Agent queries a service, it randomly requests a service from the US unique services available. Each User Agent, in the range [0,UR), issues an approximately equal number of requests (i.e. 1000/UR requests each). In every UR requests, SQ requests are issued simultaneously. Since it is more reasonable in MANET environments to find one matching service, versus finding as many matching services as possible, the max Search parameter is left to its default "false" value. Furthermore, SLPManet timing

Parameters, specifically, CONFIG RETRY and CONFIG MC MAX, were set to their default values of 2 seconds and 15 seconds respectively. Similarly, the Maximum Transmission Unit (MTU) for SLP Mane messages, excluding UDP headers, is kept at its default value of 1400 bytes.

9.0PERFORMANCE METRICS:

Note that a new "discovery transaction" is said to be initiated whenever a UA requires a service. A "discovery transaction" is said to be "successful" if either the service that the UA requires was already locally cached (i.e. cache hit), or at least one SrvRply was successfully received by the querying UA in reply to one or more (if re-transmitted) SrvRqst(s).We define below four metrics we used in our evaluation of SLPManet.

1. Overall discovery success: the percentage of discovery transactions that was successful by the end of the simulation period. This metric measures the overall performance and throughput of the protocol. The overall discovery success is composed of:

(a) Discovered services: The fraction of the successful discovery transactions that were a result of newly discovered services (i.e. was a result of a successfully received SrvRply to a SrvRqst).

(b) Cached services: The fraction of the successful discovery transactions that were a result of matching services already cached at the querying UA (thus the discovery transaction did not lead to broadcasting SrvRqsts).

2. Service lookup bandwidth of successful discovery transactions: bandwidth consumed by SLPManet messages from the time a service is first requested by a UA until the first matching service information is available at the UA. This metric measures the efficiency of the protocol. We measure:

(a) Peak bandwidth: The maximum of all successful discovery transactions' bandwidth consumption.

(b) Average bandwidth: The mean of all successful discovery transactions' bandwidth consumption.

(c) Standard deviation: Measures the dispersion in service lookup bandwidth of all successful discovery transactions.

3. Aggregate service discovery bandwidth: bandwidth consumed by the whole lookup process. This metric measures the efficiency of the protocol, and demonstrates the degree by which the protocol is conservative of network resources. Aggregate bandwidth is composed of:

(a) Useful bandwidth: Total bandwidth induced by all successful discov- ery transactions.

(b) Wasted bandwidth: SrvRqsts that do not yield SrvRplys, or where the SrvRplys are not received by the querying UA result in wasted bandwidth.

4. Service lookup latency of successful discovery transactions: time elapsed from the time a service is first needed by a UA and the first matching service information is available at the UA. This metric measures the responsiveness of the protocol. We measure:

(a) Peak latency: The maximum of all successful discovery transactions' lookup latencies.

(b) Average bandwidth: The mean of all successful discovery transactions' lookup latencies.

(c) Standard deviation: Measures the dispersion in service lookup latencies of all successful discovery transactions.

10.0EVALUATION OF SLPMANET:

Since the performance of SLPManet is likely to be sensitive to movement patterns, we ran BENCH Manet with different sets of mobility files (i.e. ten movement files per benchmark run) until the standard deviation of the samples became fairly small for the majority of the benchmark tests. The number of necessary samples was thus determined to be 10. In the subsequent sections, all metrics and quantities resemble the mean of the 10 samples. We also calculate the 95% confidence interval for each of these metrics, and illustrate the confidence intervals by means of vertical y-bars on the bar-charts.nOverall Service Discovery Success

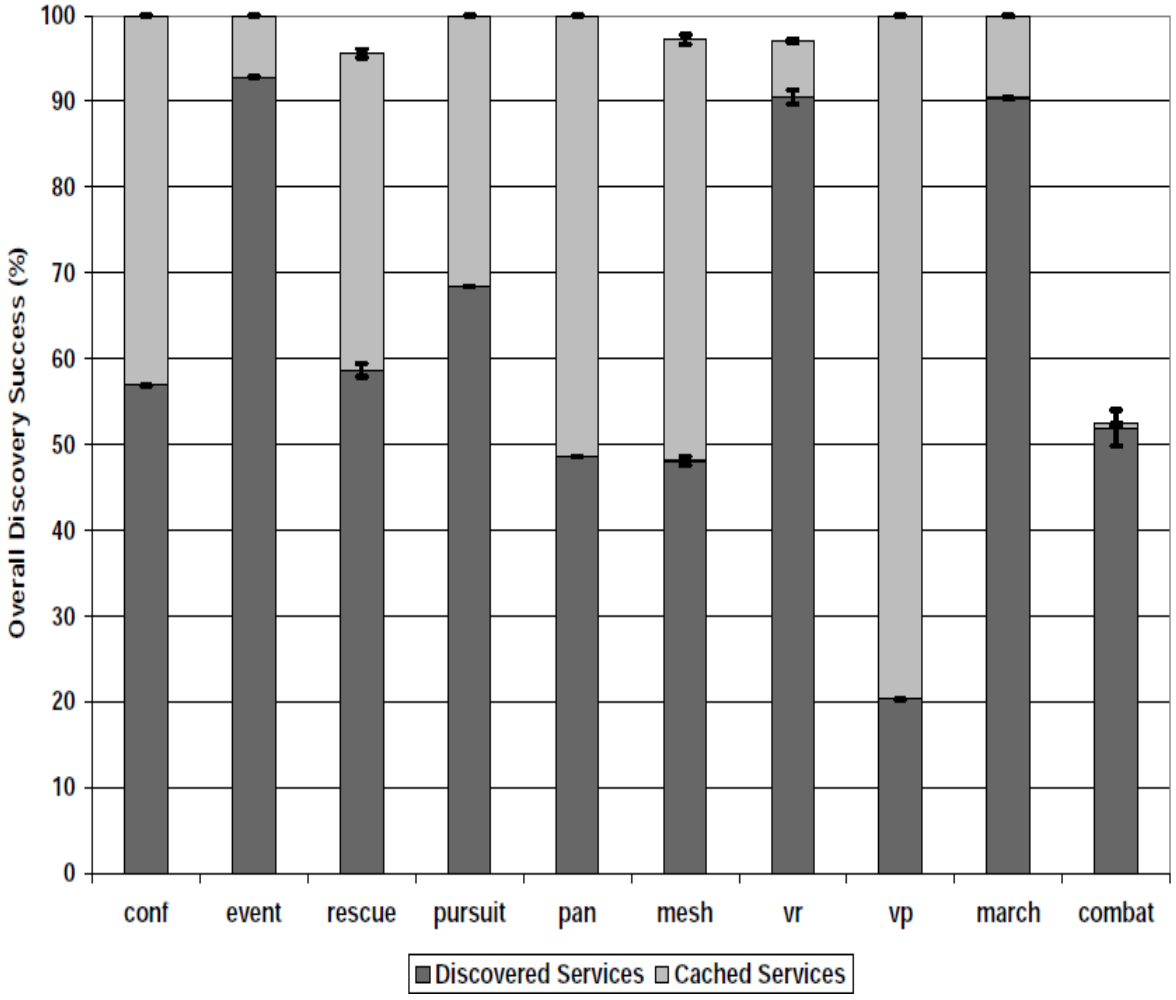

Figure 1: SLPManet overall service discovery success

Figure 1 shows the overall discovery success percentage of SLPManet across the 10 different scenarios in BENCH Manet. The bar-chart also shows the proportions that cached services and newly discovered services contributed toward the overall success. It is notable that 9 out of the 10 benchmark scenarios achieve an overall discovery success greater than 95%. 6 of these 9 scenarios achieve a discovery success of 100%, meaning that every time a service was needed by a UA, either a SrvRply was successfully received, or matching service information was already cached at the UA.

11.0 CONCLUSIONS:

In this paper, a service discovery protocol, Service Location Protocol (SLP) Version 2 [2], has been adapted to suit Mobile Ad Hoc Networks (MANETs). The adapted protocol, Service Location Protocol for MANET (SLP Manet), was implemented using Network Simulator Version 2 (NS-2). Various existing and potential applications of MANETs were carefully examined in order to determine unique mobility and networking characteristics for each different class of applications. A benchmark, Benchmark for MANET (BENCH Manet), consisting of a suit of tests, each of which is based on the determined reference configurations of the various representative MANET applications, was developed. Hence, BENCH Manet facilitated a thorough performance evaluation of SLP Manet. The performance evaluation of SLPManet shows that the overall success percentage for most of the tests in BENCH Manet exceeded 95%. However, scenarios with a high degree of mobility, and fast changing network topologies, such as combat, performed poorly in terms of overall discovery success.

REFERENCES:

1. J. Macker and I. Chakeres, "Mobile Ad-hoc Networks (manet) IETF Char- ter." http://www.ietf.org/html.charters/manet-charter.html.
2. E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Pro- tocol, Version 2," Request for Comments Standards Track 2608, Internet Engineering Task Force, June 1999.
3. Sun Microsystems, Jini Device Architecture Specification, Version 2.0. http://www.sun.com/software/jini/specs/jini20.pdf.
4. "Salution Architecture Overview."White Paper, 1998.http://www. salutation.org/whitepaper/originalwp.pdf.
5. Universal Plug and Play Forum, Universal Plug and Play Device Architecture, Version 1.0, June 2000. http://www.upnp.org/download/UPnPDA10 20000613.htm.
6. Bluetooth SIG, Specification of the Bluetooth System, Volume 1, 2001. http: //www.bluetooth.com.
7. M. Nidd, "Service Discovery in DEAPspace," IEEE Personal Comm., pp. 39–45, August 2001.
8. M. Storey, G. Blair, and A. Friday, "MARE: resource discovery and con- figuration in ad hoc networks," Mobile Networks and Applications, vol. 7, pp. 377 – 387, 2002.
9. F. Zhu, M. Mutka, and L. Ni, "Splendor: A secure, private, and location aware service discovery protocol supporting mobile services," in Proceeding of the 1st IEEE Annual Conference on Pervasive Computing and Commnications, pp. 235–242, IEEE Computer Society Press, March 2003.
10. [F. Zhu, M. Mutka, and L. Ni, "Classification of Service Discovery inPerva- sive Computing Environments," Technical Report MSU-CSE-02-24, Michigan State University, East Lansing, 2002.
11. S. Helal, N. Desai, V. Verma, and C. Lee, "Konark - A Service Discoveryand Delivery Protocol for Ad-hoc Networks," in Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC), vol. 3, pp. 2107–2113, March 2003.
12. R. Koodli and C. Perkins, "Service Discovery in On-Demand Ad Hoc Networks," Internet-Draft, Internet Engineering Task Force, October 2002.
13. L. Li and L. Lamont, "A Lightweight Service Discovery Mechanism for Mobile Ad Hoc Pervasive Environment Using Cross-Layer Design," in Proceeding of the Third IEEE International Conference on Pervasive Comput-ing and Communications Workshops (PERCOMW'05), pp. 55–59, March 2005.
14. F. Zhu, M. Mutka, and L. Ni, "Facilitating secure ad hoc service discovery in public environments," in Proceedings of the 27th Annual International Conference on Computer Software and Applications (COMPSAC), pp. 433–438, November 2003.
15. M. Balazinska, H. Balakrishnan, and D.Karger, "INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery," in Proceeding of the International Conference on Pervasive Computing (Pervasive), (Springer-Verlag, Zurich, Switzerland), 2002.