



An Evaluation of Mobility Effect on Tiny Service Discovery Protocol for Wireless Sensor Networks

Ahmed Salim

Department of Mathematics,
Faculty of Science, Zagazig University,
Zagazig, P. O. Box 44519, Egypt.

Department of Mathematics,
Faculty of Science and Arts, Al-mithnab,
Qassim University, P. O. Box 931, Buridah 51931,
Al-mithnab, Kingdom of Saudi Arabia.

asalim@zu.edu.eg

Akram El Khatib

Mathematics Department
Suez Canal University
Ismailia, Egypt

akram_elkhatib@hotmail.com

ABSTRACT

In mobile sensors environment, nodes change their positions this leads to crucially affects service discovery of wireless sensor networks (WSNs). There for the accuracy of most service discovery can be limited to just small area of movement, or demands considerable maintenance efforts in term of neighbornodesallocation. A large waiting time of wireless sensor applications spent in node discovery, as nodes need to periodically advertise their presence and be awake to discover other nodes for services. The optimization of waiting time, which is generally a hard task in static wireless sensor network s, is even harder in mobile wireless sensor networks, where the neighboring nodes also change over time. In this paper, the effect of node mobility on the performance of Tiny Service Discovery Protocol (TinySDP)in WSNs has been analyzed. In order to measure and evaluate the performance of TinySDP in mobileWSNs (MWSN), three major metrics of evaluation has been considered such as, Success ratio, Number oftransmitted messages and Average waiting time. Simulation results show that the success ratio and average waiting time of TinySDP in mobile WSNs decreased and the number of transmitted message increased.

Indexing terms/Keywords

Wireless Sensor Network (WSN); Service Discovery Protocol; Mobility Models; Mobile Wireless Sensor Network (MWSN).

Academic Discipline and Sub-Disciplines

Computer Science, Engineering, Information Technology.

SUBJECT CLASSIFICATION

Service Discovery.

TYPE (METHOD/APPROACH)

Analysis and Modeling

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol. 14, No. 1

www.ijctonline.com, editorijctonline@gmail.com



1. INTRODUCTION

The vast advancements in technology in general and in wireless communications specifically have given us the ability to mass-produce small, low-cost sensors that can connect to each other wirelessly. The sensors once deployed, whether in a random or a pre-engineered way will connect to each other and form a wireless sensor networks. WSNs are made of a large number of sensors deployed in a certain area. The sensors would transform physical data into a form that would make it easier for the user to understand. WSNs technology is growing rapidly and becoming cheaper and easier to afford, allowing different kinds of application usage of such networks. WSNs provide the right infrastructure of a large class of applications. WSNs can be used for a wide variety of applications dealing with monitoring (health environments, seismic, etc.), control (object detection and tracking), and surveillance (battlefields surveillance) [41–47]. Recent applications, however, include Mobile Sensor Networks (MSNs) where sensor nodes move freely throughout the network, carried by people, robots, or vehicles. The mobility of nodes has raised issues in handling dynamic link changes or mobility patterns that have not been considered in conventional sensor networks. In particular, the mobility of the nodes requires related algorithms to be stateless because link states between mobile nodes are valid for only a short duration. In addition, the underlying communication protocols in MSNs need to be scalable and robust to endure dynamic environments. Many technical challenges are, therefore, newly addressed in MSNs. Service discovery protocols are network protocols that allow automatic detection of devices and services offered by these devices on a computer network. Clients, the nodes that need a service, perform a discovery step, which typically initiates (limited) flooding of the network to discover nodes offering appropriate services. In some cases, clients may directly seek the needed services themselves; in others, they may contact one or more service catalogs, which maintain directories of available services. A discovery attempt generally classifies the service by type and may optionally include requirements such as a manufacturer, serial number, or other service attributes. Once the service providing node is found, it is important to ensure that the nodes can use the services efficiently [6]. Unfortunately, existing service discovery protocols (such as Service Location Protocol (SLP) [8], Jini [4], Microsoft Universal Plug and Play (UPnP) [3], DEAPspace [36], Gossip-based Discovery [11], and Pervasive Discovery Protocol (PDP) [12]) that used in MANET (Mobile Ad-hoc Network) are too resource intensive to be used in sensor networks.

A sensor node provides services consisting of the sensed information, based on the types of sensors available, along with context, such as location, which makes that information useful. Taking the unified approach, in which a node in the sensor network is viewed as providing a set of location-based services, forms the basis of new location aware service discovery frameworks specifically tailored to sensor networks. The Tiny Service Discovery Protocol (TinySDP) [6], as its name suggests, is a lightweight protocol that allows discovery and selection of services in WSN. In this paper, we analyze the TinySDP protocol with mobile WSN since adding mobility to sensor networks can significantly increase the capability of the sensor network by making it resilient to failures, reactive to events, and able to support disparate missions with a common set of sensors. The rest of the paper is organized as follows: In Section 2, we discuss background studies and related research. In Section 3, will describe TinySDP and discuss mobility models. Our problem formulation is covered in Section 4. In Section 5, the experimental simulation posted in details with both network and simulation setup. In Section 6, we conclude our work.

2. RELATED RESEARCH

C. K. Toh [35] has outlined different service discovery architectures for managing service information on MANETs. In terms of service discovery, a MANET node may act as a client (or service requestor) that wants to discover a type of service, a server (or service provider) that wants to make its services available to other MANET nodes, or a service coordinator (SC) that assists with service discovery. SCs are nodes that hold a central repository for caching service descriptions and service bindings. Existing Service Discovery Protocols classified into two categories: The commercial scientific researches protocols, which are available in large scales. The focus of researches has been recently on enhancing efficiency and scalability of the existing protocols. Some protocols use peer-to-peer distributed hash table, which can be prolonged at least based on a network of city such as in INS/Twine [32] and Superstring [26]. Simple Service Discovery Protocol (SSDP) [3], Toh [35], Jini [4], Service Location Protocol (SLP) [8] are not working properly in all networks. They usually begin with a predetermined network. Toh [35] suggests different service discovery architectures to manage service information on ad-hoc networks for example, semi-centralized architecture is fully distributed and hybrid. SSDP provides a mechanism for HTTP clients and HTTP resources to discover each other in a local area network. It is used in Microsoft UPnP (Universal Plug and Play) [3] support plug and play to add a service that diabetes under the computers running Windows operating system.

UPnP devices are fully transparent and have standard connection technology and peer-to-peer discovery and configuration. UPnP specification, the Microsoft implementation and sample source code available at WinHEC [3]. Jini is a service discovery technology based on Java [4]. In Jini, lookup services provide catalogs of available services to clients. Upon initialization, Jini services register their availability by uploading proxy objects to one or more of these lookup services. Once a client has contacted a lookup service, it can search for interesting services and then download the corresponding service proxy objects. However, Jini's dependability on Java makes it unsuitable for sensor networks. Service Location Protocol (SLP) [8] is the automatic detection of resources in networks to Internet Protocol. SLP is an independent protocol language. Thus, protocol specification can be implemented in any language, which is based on the detection mechanism of service attributes that rely on different ways of describing the service. SLP infrastructure consists of three types of agents. DEAPspace [36] is a decentralized discovery algorithm targeted at wireless ad-hoc single-hop networks. It uses a pure push-based approach for service discovery. The time is slotted into intervals and service information is broadcast during these intervals. The broadcasts are single-hop and are not transmitted beyond the local transmission range of the device in question. This is not suitable for wireless sensor networks.

Gossip-based discovery [11] is a service discovery protocol for mobile ad-hoc networks. The device collects information about the services on the net, listen to each service announcement, and the demand and response messages in the network. Service Registry on any device of local services as well as services that can be stored by other vendors. Pervasive Discovery Protocol (PDP) [12] concentrates on service discovery in ad-hoc networks, where mobile devices communicate via wireless links without any fixed infrastructure. It is a fully distributed protocol that uses both push and pull techniques. In PDP, a device announces its service only when other devices request the services. The services in the PDP messages are defined using a URL scheme similar to the one used in the Service Location Protocol (SLP) [8] and transmitted using UDP or TCP. This introduces too much overhead to implement these protocols on wireless sensor nodes. The simple discovery of services in terms of service name resolutions mechanism presented based on a name resolution [16], studied in parallel to the normal Internet service DNS SRV discovery [38]. This allows the server IP address and the transport protocol to be used must be resolved to implement the service. The type of transportation service on behalf of the controller [37] is encoded. Demand for service discovery can use this combination without adding significant complexity network built. Tiny Service Discovery Protocol (TinySDP) [6] is a lightweight protocol that allows discovery and selection of services in a network. In this paper, we focus on the components of the protocol relevant to my research and analyze TinySDP in mobile WSNs. The protocol allows services to advertise their availability and clients to browse the network, determining in which service exist and the characteristics of those services. The network nodes need little or no static configuration for discovering services.

3. BACKGROUND

3.1 Description of TinySDP

The Tiny Service Discovery Protocol is a lightweight protocol that allows discovery and selection of services in a network [6]. The protocol supports a framework by which clients (or users) needing a service are able to contact those services (see Figure 1). The clients are other sensor nodes or mobile devices like PDAs, laptops, etc. The protocol uses a distributed approach in which information is cache in the network rather than in some central server. Any advertisements or requests are sending to the network rather than to a particular node acting as a server. The process of advertising messages distributes the data in the network, increasing its access, thus making it possible to find the data quickly. When the service request reaches a node that has service information in its cache, it forwards the request to the node that provides the service. The service provider then replies directly to the client.

The TinySDP defines a packet size of 29 bytes. We have a 15 byte application header and 14 bytes of data in our packet. As depicted (see Figure 2), the protocol header carries the basic information about the packet.

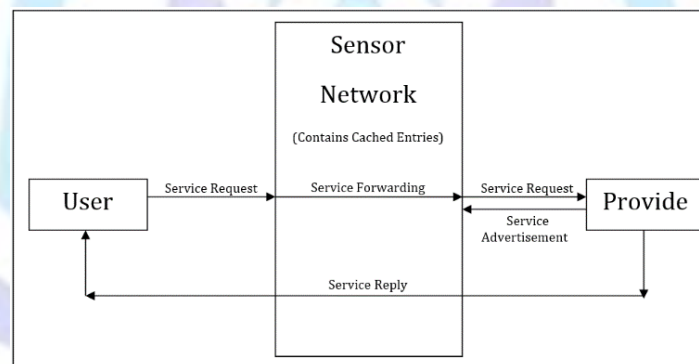


Fig 1: TinySDP Framework.

The packet type specifies whether the packet is an advertisement, request, reply, or acknowledgement. Within 8-bit field, we can define 256 different packet types. The 16-bit service type field specifies the service whose information the packet is carrying. However, some standardization of the service values is required to ensure that a particular value means the same service to all the nodes. For instance, if we define value 310 for temperature then this value will remain the same for every application. Some of these values are for defining standardized service types, leaving the rest for application specific services. For generic message exchanges not concerning any specific service, the protocol requires a service type of zero.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	

- Bytes 0 : TinySDP Version
- Bytes 1 : Packet Type
- Bytes 2-5 : Source Node
- Bytes 6-9 : Destination Node
- Bytes 10-11: Service Type
- Bytes 12-13: Scope (0 == all scopes)
- Bytes 14 : TTL (Max Hop Count)

Fig 2: TinySDP Packet Format.

Every packet has a hop count (the time-to-live field). When hop count reaches zero, the packet discarded. Limiting the lifetime of a packet using the hop count prevents old packets from moving around in the network. Services grouped together using scopes. A scope is a set of sensors making up an administrative group. It can indicate a group of nodes within a particular location, within an administrative domain, or grouped by some other category. For instance, we may

want to control the rights of nodes by allowing them access to services on some nodes and not on others. If we want a node to look for services in all domains, we use ascope of zero.

3.1.1 Advertisement of Messages Approaches

To advertise to dedicated nodes that stores a list of all services in the network. However, this scheme has a downside. These servers/directories will need to be equipped with extra resources (power, storage, computation capability) if it works properly for a long time and not to become a bottleneck in the network. For scalability and generality, the assumption of using a central server avoided.

3.1.2 Route Resolution

TinySDP requires a multi-hop routing protocol by making a few assumptions while routing the packets. All nodes know their location relative to the coordinate system [5], the nodes know their neighbors [9], they have a sense of directionality [29] and the source site can find the coordinates of the destination node with the location service [20], [34]. The implementation uses the Trajectory Based Forwarding (TBF) [19], (see Figure 3), as a routing protocol. It is a hybrid based on source routing and forwarding Cartesian [38].

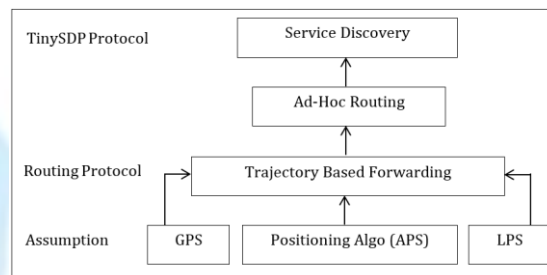


Fig 3: Routing Support in Service Discovery.

3.1.3 Service Resolution

Let us now look at the request process. As already mentioned, the client sends out a service request along a particular path toward the edges of the network. When a node receives a service request packet, it checks whether it provides the service or not. There are two variants depending on whether or not it does:

1. If it provides the service, it sends back to service reply to the source/client. If he cannot meet the client requirements completely, then it forwards the service request along its trajectory, but also offers to provide limited service.

For example, if a client is looking for a specific color printer of resolution, it sends a service request specifying these parameters. A node in the network that receives this request can provide a black-and-white printout with the same resolution. This node can forward the service request further, and then it sends back a reply to the client offering its services.

2. If it does not provide a service, it checks whether it is advertising the service in its cache. Again, there are two possibilities:
 - If it has an entry in the cache, it forwards the service request to the service provider.
 - If it does not have the cache entry, it checks whether the packet meant for itself.
 - If the packet was destined for the local node, indicating this is the last node along the path of the request, it sends a negative acknowledgment to the source node.
 - If the packet not meant for the local node, then it forwards the packet along its trajectory.

3.1.4 Data Structures

All nodes store some basic information, including information about nodes services, its neighbors, and advertisements received from other nodes. We have included only the most essential information in the cache entries. Every node uses three data structures for storing information. Two additional tables are required for use with RPC in service discovery: an RPC client table and an RPC server table.

1. Advertisement Table.
2. Service Table.
3. Neighbor Table.
4. RPC Client Table.
5. RPC Server Table.

3.2 Mobility Models

The Random Waypoint model, which is a very simple mobility model, has been presented in [7] based on pause time between changing *direction/speed*. Background a random point in the simulation area with a uniformly distributed speed between [*minSpeed*, *maxSpeed*]. After arriving to the destination again waits for the same period of time (*pausetime*) before moving to a new place. In [1] and [18], there are common problems with simulation studies using Random

Waypoint model due to poor choice of velocity distribution, uniform distribution. If *minSpeed* is zero, such velocity distribution leads to a situation where average speed approaches zero and at the stationary state each node stops moving. Random Direction and walk nodes change their *speed/direction* every time slot. In this model, new direction from θ is chosen randomly between $(0, 2\pi)$. The speed chosen from uniform (or Gaussian) distribution. In this model node reaches boundary it bounces back with $(\pi-\theta)$. The following figure described about the movement behavior of a node in random waypoint mobility model (see Figure 4).

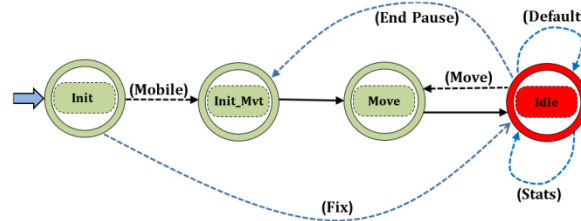


Fig 4: Random Waypoint Mobility Process Model

3.2.1 Random Waypoint

It is a real need for developing a deeper understanding of mobility models and their impact on Tiny Service Discovery protocol (TinySDP) performance. The movement pattern of mobile wireless sensor nodes is described by mobility model. The mobility model describes the pattern of movement, location, velocity and acceleration over time. Since mobility patterns may play a significant role in determining the TinySDP performance, it is desirable for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way. Thus, when evaluating TinySDP, it is necessary to choose the proper underlying mobility model. It is not appropriate to evaluate the applications where nodes tend to move together using Random Waypoint model (RWPM) [7] or whatever else. The Random Waypoint model and its variants are designed to mimic the movement of mobile wireless sensor nodes in a simplified way. Because of its simplicity of implementation and analysis, they are widely accepted.

4. Problem Description

Service discovery in mobile WSN is a challenging issue. The nodes in a WSN offer spontaneous and variable connectivity. In addition, the proximity of a given service as well as the kind and the number of services vary unpredictably with time. The mobile nodes can also provide services that may be of interest to other mobile nodes. Therefore, there is always a need for having a mechanism that can discover and utilize services provided by other devices. The service discovery techniques used for this purpose. Service Description is also an important aspect of a service discovery protocol as well as proper descriptions facilitate the searching of a service. Usually these aspects are an intrinsic part of a service discovery protocol. Restriction applied to that aspect of service discovery that is concerned with finding the location of a particular service in mobile WSN. For that, our aim is to analyze the behavior of TinySDP in mobile WSNs. In order to analyze the results, we first define a few metrics [2]. We denote the total number of nodes as N . The Number of clients which successfully locate the Service as (N_{succ}) and the number of Service Advertisement Messages as (M_{adv}) . The number of Service Request Messages as (M_{req}) and the total number of messages sent as (M_{total}) . Each successful client (C) receives message from 1 to m service reply Messages of waiting time (t_1, t_2, \dots, t_m) .

- **Success Ratio (SR):** The ratio (Calculated as a Percentage (%)) of the number of nodes that successfully locate the service, over the total client number. It is calculated by the following equation:

$$SR = N_{succ} / N \times 100 (\%)$$

- **Number of Transmitted Messages (Mtotal):** The total number of messages transmitted for a period of the simulation. This is a useful parameter for estimating the efficiency of the Protocol. It is calculated by the following equation:

$$M_{total} = \sum_{n=1}^N (M_{req} + M_{adv})$$

- **Average Waiting Time (AWT):** The minimum period in seconds, of time, averaged over all customers, from the message transmission service request and ending with the reception of reply message. It is calculated by the following equation:

$$AWT = \frac{\sum_{n=1}^N \min(t_1, t_2, \dots, t_m)}{N}$$

5. Simulation Results

In this section, we discuss the simulator used for paper, the simulation environment and draw conclusions from the results. In this paper, we use TOSSIM [55], the simulator for TinyOS. All simulation had conducted with TinyOS version (1.1.11) and TOSSIM version (1.1) TOSSIM is the TinyOS simulator. TOSSIM captures network behavior at a high fidelity while scaling to thousands of nodes. TOSSIM compiles directly from TinyOS code, and is by default automatically compiled (make pc). It runs the same code that runs on sensor network hardware. TOSSIM uses very simple but surprisingly powerful abstractions for its wireless network. The network is a directed graph, in which each vertex is a node, and each edge has a bit error probability. Each node has a private piece of state representing what it hears on the radio channel. TOSSIM provides an excellent tool to study and analyze the algorithm design by providing a controlled and reproducible environment, and by enabling access to tools such as debuggers.



5.1 Network Setting

First, a network with 50 simulated. Then, we extended number of nodes to 100. The simulation performed in two-state experiment, 50 nodes and 100 nodes. We choose two-state experiment to study modified effect and to highlight on results. All nodes are assumed to have the same resources in terms of power and memory. There are 11 kinds of services offered in the network. We assume that all services follow a uniform distribution. We used a random topology and Random Waypoint Model for the network. We have assumed a convex dense network with MTinySDP. We use a mobile radio model; all nodes have full connectivity to their neighbors. Using a more realistic radio model that includes errors would increase the number of messages that need to be transmitted, but should not alter the relative performance of the various protocols.

5.2 Simulation Setting

Simulation has four combinations with respect to number of nodes and network mobility. First trial, we considered 50 nodes with TinySDP in static WSN. Second trial, we considered 50 nodes with TinySDP immobile WSN. These two trails address the effect of mobility in small network. Third trial, we considered 100 nodes with TinySDP in static WSN. Forth trial, we considered 100 nodes with TinySDP in mobile WSN. These two trails address the effect of mobility in large network. Each three minutes of the real time of a node, the node starts executing TinySDP. According to metric calculations, advertisements are sent out when the network. In first state we have 50 nodes in our network, a few of the service request packets get lost if all of the 50 nodes start sending out their service request together at network startup. In second state we have 100 nodes in our network, a large of the service request packets get lost if all of the 100 nodes. In order to ensure that this does not happen, the nodes send out their advertisement packets only in certain slotted time intervals each 1 minute. A timer that outputs the collected information into the screen. Simulator should run for a time of 100 minutes to get the best reading of sending out request packets. The parameters are listed in (Table 1).

Table 1. Network and Simulation Parameter

Network And Simulation Parameter	
Total Number Of Node	50, 100
Mobility Model	Random Waypoint
Speed	1-3 m/s
Pause Time	1-10s
Area Scenario	100 X 100m
Transmission Range, R	15m
Initial positions	Uniform distributed
Simulation duration	100 minutes
Node is initialized	At a Random Location

5.3 Results Analysis and Discussion

We have to analyze and compare results of TinySDP with static WSN with results of TinySDP with mobile WSN. Since we are assuming a stationary network once and then assuming dynamic network, we compare results with those results appeared in [6] that assume a nearly stationary and dynamic network. A side-by-side comparison of performance results is as shown in the following (Table 2) once with 50 nodes and another with 100 nodes. The main result of our experiment has shown that the TinySDP with MWSN has lesser values in performance metrics in terms of the Total Number of Transmitted Messages (M_{total}) and Success Ratio (SR) and lesser values in Average Waiting Time (AWT) for discovering service is a measure of the high performance metrics of TinySDP with static WSN. In Static WSN, TinySDP has almost the same success ratio. This indicates that number of nodes has no effect on the performance in static WSN. In the other side, we can see clearly that success ratio directly affected by both mobility of nodes and number of nodes.

Table 2. Performance Comparison

Comparison	Routing Protocol	N	Nsucc	Madv	Mreq	Max SR	Mtotal	AWT
TinySDP with Static WSN	TBF	50	49	203	218	98%	421	0.5275
		100	98	441	406	98%	847	0.5113
TinySDP With MWSN	TBF	50	31	271	340	62%	611	0.3518
		100	38	497	757	38%	1254	0.4101

We visualized results in four variant figures. Performance comparison of all metrics and then Service Request Messages, Service Advertisement Messages and Total Number of Messages Sent between TinySDP with static WSN and TinySDP with MWSN.

Figure (5) illustrated that TinySDP with static WSN advances TinySDP with MWSN in overall performance comparison with both 50 and 100 nodes network. TinySDP with MWSN has sent more total messages than TinySDP with static WSN. TinySDP with static WSN sent about 400 total messages in 50 nodes network and about 800 total messages in 100 nodes

network, but TinySDP with MWSN sent over that number of messages about 600 messages in 50 nodes network and about 1200 total messages in 100 nodes network. In all cases of the both scenario of service requests, total number of messages indicates that lost messages in WSN because of mobility. In static WSN, Success ratio in both 50 and 100 nodes network has get high score about 98 present. In the mobile WSN, service discovery has a serious decrease, as number of nodes gets bigger. MWSN with 50 nodes scored about 60 present but with 100 nodes scored 30 present. This present of success ratio is considered very low and is not accepted for service discovery.

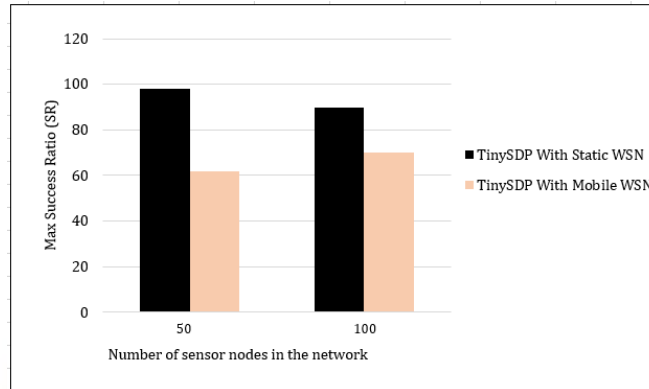


Fig 5: Success Ratio (SR) of TinySDP with static WSN and TinySDP with MWSN with different number of sensor nodes

Figure (6) describes the relationships between the number of service request messages and the number of nodes in both classical and TinySDP with MWSN. As we can see clearly from this figure, the superiority of TinySDP with static WSN over TinySDP with MWSN. We notice a convergence between them when number of nodes are in range of 10 to 35 nodes. TinySDP with static WSN wildly advances TinySDP with MWSN after number of nodes exceeds 35. Thus, we can say in TinySDP with MWSN, the number of service requests messages has a direct proportionality relationship with number of nodes with large number of nodes. This gives us a clear indication of performance drop in TinySDP with MWSN as number of nodes increase.

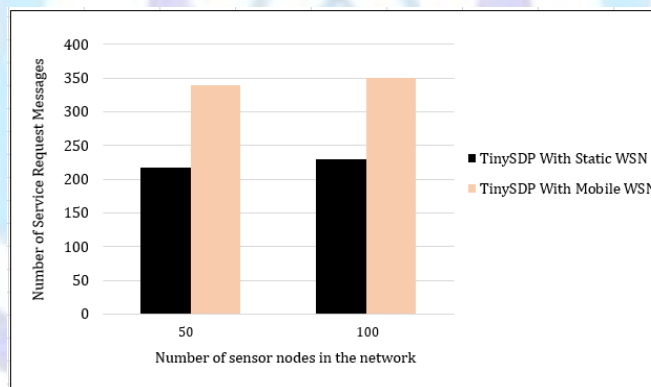


Fig 6: Performance Number of Service Request Messages of TinySDP with static WSN and TinySDP with MWSN with different number of sensor nodes

Figure (7) describes the relationships between the number of service Advertisement messages and the number of nodes in both TinySDP with static WSN and TinySDP with MWSN. Advertisement process is a periodic process and has no relations with number of nodes or its mobility. Since this loosely relationship, we notice almost no difference between TinySDP with static WSN and TinySDP with MWSN in advertisement comparison. We can describe the advertisement as steady and stable over time and the increase of number of nodes.

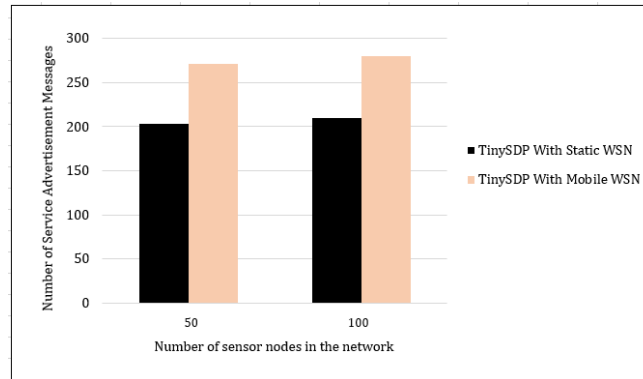


Fig 7: Performance Number of Service Advertisement Messages of TinySDP with static WSN and TinySDP with MWSN With different number of sensor nodes

In Figure (8), Total Number of Messages Sent between TinySDP with static WSN and TinySDP with MWSN has a clear gap. The advance is clearly goes towards TinySDP with MWSN. We notice that the step between TinySDP with static WSN and TinySDP with MWSN is almost fixed until number of nodes reaches 35 nodes. After that, we notice that the step is widely changes towards TinySDP with static WSN.

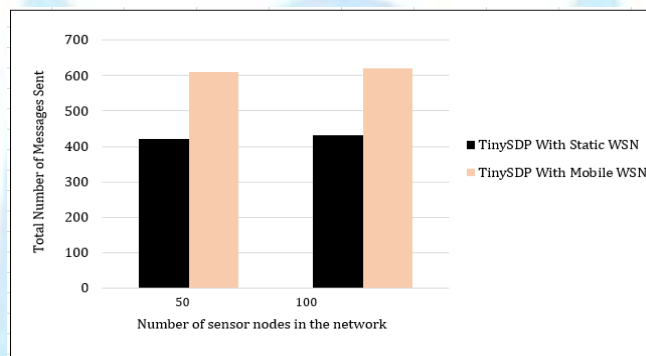


Fig 8: Performance Total Number of Messages Sent of TinySDP with static WSN and TinySDP with MWSN with different number of sensor nodes

In Figure (9) describes the relationships between the Average Waiting Time (AWT) from the message transmission service request and the number of nodes in both classical and TinySDP with MWSN. As illustrated in (see Figure 9), the AWT of TinySDP with MWSN is much better than TinySDP with static WSN. In the case of the number of nodes in the network less than 50 nodes. However, in the case of exceeding the numbers of the nodes of 50, in the case of TinySDP with static WSN outweigh the MWSN. Therefore, TinySDP with MWSN, have a direct relationship to the value of the average waiting time with the number of nodes in the network.

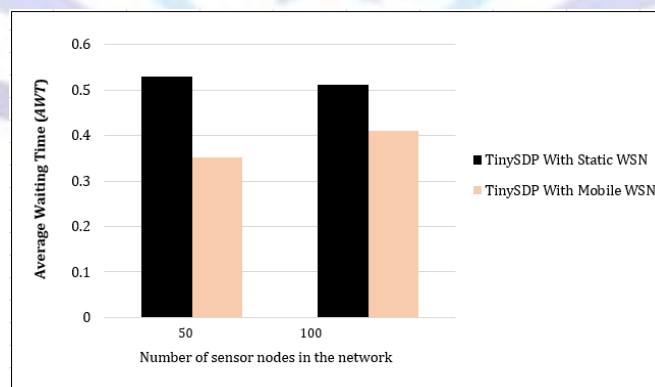


Fig 9: Average Waiting Time (AWT) of Message Transmission of TinySDP with static WSN and TinySDP with MWSN with different number of sensor nodes

6. CONCLUSION

In this paper, evaluate the TinySDP with static WSN for performance of service discovery in mobile WSN. The Random Waypoint model, which is a very simple mobility model and very convenient with real movement of physical sensors had adapted. This work holds a comparison of TinySDP with mobile and static MWSN. The three major metrics Success ratio, Number of transmitted messages, Average Waiting time was considered. The result showed that the number of messages



almost fixed until number of nodes reaches large number of node. TinySDP has a strange behavior in mobile WSN with large number of nodes. By comparing all metrics first with 50 nodes then with 100 nodes, we concluded that TinySDP is not suitable for MWSN especially with large number of nodes also the step is widely changes towards TinySDP. In addition, in TinySDP with MWSN, the number of successful service discovery has a direct proportionality relationship with number of nodes with large number of nodes. This gives us a clear indication of performance drop in TinySDP with MWSN. The suggested modification over the TinySDP based on the principle of multiple behavioral modes of operations. TinySDP should be able to change behavior in different situations in order to improve certain metrics like reducing communication delays, improving deliverability. The suggested modification to TinySDP tackles three general aspects. First, each node marks its mobility and sends advertisement of about it to targeted neighbor nodes. Second, the awareness of vicinity mobile neighbor nodes and network topology that allows to benefit from both the push and pull mode. Third, new mechanism for sending services advertisement that reduces case of unknown service. By tackling these three aspects, the protocol can change its behavior for the better performance. To be noticed, suggested modification of TinySDP is open to improvement.

REFERENCES

- [1] Random waypoint model. <https://www.netlab.tkk.fi/esa/java/rwp/rwp-model.shtml>. Retrieved: 2013-05-01.
- [2] B. Parbat, A. K. Dwivedi, and O. P. Vyas. "Data Visualization Tools for WSNs: A Glimpse". International Journal of Computer Applications, Volume 2, Issue 1, ISBN:0975-8887, Pages:14-20, May 2010.
- [3] PnP Forum. "UPnP Device Architecture References Specification, V2.0". Available (online) at [Http://www.upnp.org](http://www.upnp.org).
- [4] Sun Microsystems. "Jini Network Technology". Available (online) at [Http://www.jini.org](http://www.jini.org).
- [5] X. Li, N. Santoro, I. Stojmenovic. "Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks". IEEE Transactions on Computers, Volume 58, Issue 9, Pages:42-50, September 2009.
- [6] Kaur, M, Bhatt, S. ;Schwiebert, L. ; Richard, G.G. "An Efficient Protocol for Service Discovery in Wireless Sensor Networks", GLOBECOM Workshops, 2008 IEEE, Pages:1-6, Dec 2008.
- [7] Bhavyesh Divecha, Ajith Abraham, et al, "Impact of Node Mobility on MANET Routing Protocols Models", Journal of Digital Information Management, Volume 5, Number 1, pp. 19-24, September 2007.
- [8] C. Toh, G. Guichal, D. Kim, and V. Li. "Service Location Protocols for Mobile Wireless ad-hoc Networks". Inderscience Publishers, Volume 2, Issue 4, Pages:250-262, June 2007.
- [9] R. Madan, and S. Lall. "An Energy-Optimal Algorithm for Neighbor Discovery in Wireless Sensor Networks". Mobile Networks and Applications, Volume 11, Issue 3, Pages:317-326, June 2006.
- [10] D. Chakraborty, A. Joshi, Y. Yesha and T. Finin, "Toward Distributed Service Discovery in Pervasive Computing Environments," IEEE Transactions on Mobile Computing, vol. 5, no. 2, IEEE CS Press, pp.97- 112, February 2006.
- [11] C. Lee, S. Helal, and W. Lee. "Gossip-Based Service Discovery in Mobile ad-hoc Networks". IEICE Transactions on Communications, Pages:2621-2624, September 2006.
- [12] C. Campo, C. Garcia-Rubio, A. M. Lopez, and F. Almenarez, PDP: A lightweight discovery protocol for local-scope interactions in wireless ad hoc networks Computer Networks, 50(17), 3264-3283, December 2006.
- [13] Eylem Ekici, Yaoyao Gu and Doruk Bozdog. Mobility-based communication in wireless sensor networks. IEEE Communications Magazine, July 2006.
- [14] A. Varshavsky, B. Reid and E. Lara, "A Cross Layer Approach to Service Discovery and Selection in MANETs," Proc. Second International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2005), November 2005.
- [15] V. Lenders, M. May and B. Plattner, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," Proc. Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05), Vol. 1, pp. 120-130, June 2005.
- [16] P. Engelstad, and Y. Zheng. "Evaluation of Service Discovery Architectures for Mobile Ad-Hoc Networks". Wireless On-demand Network Systems and Services Conference, Pages:2-15, January 2005.
- [17] J. Tyan and Q. Mahmoud, "A Network Layer Based Architecture for Service Discovery in Mobile Ad-Hoc Networks," Proc. 17th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2004), vol. 3, pp. 1379-1384, May 2004.
- [18] Jungkeun Yoon, Mingyan Liu, and Brian Noble, "Random waypoint considered harmful", Technical report, Electrical Engineering and Computer Science Department, University of Michigan lecture notes on, vol.2, pp. 1312-1321, April 2003.
- [19] D. Niculescu, and B. Nath. "Trajectory Based Forwarding and its Applications". Technical Report DCS-TR-488, Department of Computer Science, Rutgers University, ISBN: 1-58113-753-2, Pages:260-272, September 2003.
- [20] D. Niculescu, and B. Nath. "Ad-hoc Positioning System (APS) Using AoA". In INFOCOM, IEEE Expert Magazine, Pages:1734-1743, April 2003.
- [21] S. Helal, N. Desai, V. Verma and C. Lee, "Konark- a Service Discovery and Delivery Protocol for Ad-Hoc Networks," Wireless Communications and Networking, vol. 3, IEEE Press, pp. 2107- 2113, March 2003.



- [22] M. Klein, B. Konig-Ries and P. Obreiter, "Service Rings - A Semantic Overlay for Service Discovery in Ad-Hoc Networks," Proc. 14th International Conference on Database and Expert Systems Applications (DEXA'2003), IEEE CS Press, pp. 180-185, September 2003.
- [23] M. Klein, B. Konig-ries and P. Obreiter, "Lanes - A lightweight overlay for Service Discovery in Mobile Ad Hoc Networks, 3rd Workshop on Applications and Services in Wireless Networks (ASWN2003), July 2003.
- [24] U. Kozart and L. Tassiulas, "Network Layer Support for Service Discovery in Mobile Ad Hoc Networks," Proc. IEEE INFOCOM-2003, April 2003.
- [25] F. Zhu, M. Mutka and L. Splendor, "A Secure, Private and Location-aware Service Discovery Protocol Supporting Mobile Services," Proc. First International Conference on Pervasive Computing and Communication (PerCom'03), ACM Press, pp. 235-242, 2003.
- [26] R. Robinson, and J. Indulska. "Superstring: A Scalable Service Discovery Protocol for the Wide-Area Pervasive Environment". IEEE International Conference on Networks, Pages:699-704, October 2003.
- [27] F. Bai, N. Sadagopan, and A. Helmy, Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks, in Proceedings of IEEE Information Communications Conference (INFOCOM 2003), San Francisco, Apr. 2003.
- [28] P. Levis, N. Lee, TOSSIM: A Simulator for TinyOS Networks, Computer Science Division, University of California Berkeley, California, 17 September 2003.
- [29] A. Nasipuri, and K. Li. "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks". 1st ACM International workshop on Wireless Sensor Networks and Applications. ISBN:1-58113-589-0, Pages:105-111, September 2002.
- [30] D. Chakraborty, A. Joshi, T. Finin and Y. Yesha, "GSD: A Novel Group-Based Service Discovery Protocol for MANETs," Proc. 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN), pp. 140-144, September 2002.
- [31] O. Ratsimor, D. Chakraborty, A. Joshi and T. Finin, "Allia: Alliance-Based Service Discovery for Ad-Hoc Environments," Proc. 2nd International workshop on Mobile Commerce (WMC'02), ACM Press, pp.1-9, September 2002.
- [32] M. Balazinska, H. Balakrishnan, and D. Karger. "INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery". In International Conference on Pervasive Computing (Pervasive), London, UK, pages: 195210, August 2002.
- [33] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. Wireless Communication and Mobile Computing (WCWC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Vol.2 (no. 5):pp. 483502, 2002.
- [34] D. Niculescu, and B. Nath. "Ad-hoc Positioning System (APS)". In GLOBECOM, IEEE Expert Magazine, Pages: 2926-2931, January 2001.
- [35] C. Toh. "Ad-Hoc Mobile Wireless Networks. Protocols and Systems". Prentice Hall PTR, ISBN0:0130078174, Pages:231-242, December 2001.
- [36] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade, "DEAPspace: Transient Ad-Hoc Networking of Pervasive Devices", Computer Networks: Special Edition on Pervasive Computing 35 (4):411428, March 2001.
- [37] A. Gulbrandsen, P. Vixie, and L. Esibov. "A DNS RR for Specifying the Location of Services (DNSRRV)". RFC 2782, Internet Engineering Task Force (IETF), February 2000.
- [38] J. Li, J. Jannotti, D. De, D. Karger, and R. Morris. "A Scalable Location Service for Geographic ad-hoc Routing". In Proceedings of ACM Mobile Computing and Networking, ISBN: 1-58113-197-6, Pages: 120-130, August 2000.
- [39] L. Cheng and I. Marsic, "Service Discovery and Invocation for Mobile Ad Hoc Networked Appliances," in the CD-ROM Proc. 2nd International Workshop on Networked Appliances (IWNA 2000), December 2000.
- [40] A. D. Birrell and B. J. Nelson, Implementing Remote Procedure Calls ACM Transactions on Computer Systems, 2(1):3959, February 1984.
- [41] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," in Workshop on Data Communication, pp. 20-41, Latin America and the Caribbean, Costa Rica, April 2001.
- [42] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263-270, Seattle, Washington, USA, August 1999.
- [43] Ahmed M. Khedr and Walid Osamy, and Dhrama P Agrawal, "Perimeter Discovery in Wireless Sensor Networks," J. Parallel Distrib. Comput., vol. 69, pp. 922 - 929, 2009.



- [44] Ahmed M. Khedr and WalidOsamy, "Tracking Mobile Targets Using Random Sensor Networks," TheArabian Journal for Science and Engineering, vol. 32, No. 2B pp. 301-315, October 2007.
- [45] Ahmed M. Khedr and WalidOsamy, "A Topology Discovery Algorithm For Sensor Network UsingSmart Antennas," Computer Communications Journal, vol. 29, pp. 2261-2268, 2006.
- [46] Ahmed M. Khedr and WalidOsamy, "Target tracking Mechanism for Cluster Based Sensor Networks",Applied Mathematics and Information Science Journal, vol. 1(3) pp. 287- 303, 2007.
- [47] Ahmed M. Khedr, "New Mechanism for Tracking a Mobile Target Using Grid Sensor Networks," Computing and Informatics, vol. 28, pp.1001 1021, 2008.

