Kahloot et al. The Arab Journal of Sciences & Research Publishing, Vol. 2 - Issue (3): 2016, 4, 21 P. 97-114;
Article no: AJSRP/ K29116.

Modifying Tiny Service Discovery Protocol (MTinySDP) to overcome mobility issues

Khalid M. Kahloot[1]. Akram A. ElKhatib[2]. Ahmed A. Salim[3].

[1] Islamic University of Gaza, Gaza Strip, Palestine. **Email**: kkahloots@hotmail.com

[2] Suez Canal University,Egypt. Gaza Strip, Palestine. **Email:** akram_elkhatib@hotmail.com

[3] Suez Canal University,Egypt Ismailia, Egypt **Email:** asalim@hotmail.com

ABSTRACT

Emerging wireless communication standards and more capable sensors and actuators are pushing the development towards wireless sensor networks (WSNs). Deploying a large number of sensor nodes requires a high level framework enabling the devices to present themselves and the resources they hold. The device and the resources can be described as services. With a service discovery protocol, resources, e.g. samples of data produced by the sensor, could be provided to the application developers through high level interfaces. Service discovery protocols enable software, i.e. services and service users, to dynamically advertise and find available services on the network.

The Tiny Service Discovery Protocol (TinySDP), as its name suggests, is a lightweight protocol that allows discovery and selection of services in WSN. Here, we have analyzed 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.

In this paper, we perform analytical simulations in terms of Success ratio, number of transmitting messages and average waiting time. Simulation results show that the performance of TinySDP in mobile WSNs (MWSNs) has been decreased. As a result, we present Modified TinySDP (MTinySDP) for MWSNs. The simulation results show that MTinySDP outperforms TinySDP in terms of Success ratio, number of transmitting messages and Average waiting time for mobile and static WSN.

Indexing terms/Keywords

Wireless Sensor Network (WSN), Service Discovery Protocol (SDP), Mobility, TinySDP, TOSSIM, TinyViz

Academic Discipline And Sub-Disciplines

Information Technology, Networking, Protocols

SUBJECT CLASSIFICATION

Wireless Sensor Network

1. INTRODUCTION

Wireless sensor networks have several features that set them apart from the classical wired and wireless networks. These differences make the large body of research in Service Discovery protocols, are not directly applicable to WSNs. WSNs is One of the main features of the application specific nature. The nodes in WSNs cooperate to jointly address one or a few well-defined tasks or applications. This has a significant impact on the functionality of the Service Discovery , which traditionally have created a binding between the requested service/resource description and the Identification (ID) of the nodes of a network that provides such a service [4].

The semantic routing protocols that work without a global unique identifier (ID's) make such a direct binding unnecessary. Here the main task of the Service Discovery component will support the ability to reconfiguration of the network and to build repository of the available semantic attributes. These stores will facilitate the cooperation between the nodes in the network and will support the interaction with the human operator, and the interaction with the external networks. Of course, some specific tasks require a centralized WSNs communication (ID) [10].

Regardless of the ID-centric nature, implementation of sevice dicovery should be streamlined and should contribute to the overall energy efficiency of the protocol stack [7]. The number of generating messages is a primary factor determining the scalability of the protocol. Thus, every attempt should be made to reduce it. The tight integration between the routing and the functionality of sevice dicovery has been suggested as a promising approach to this ending. Exchange available information on the status and reduced control of overhead can significantly increase the scalability of the implementation makes it suitable for limited resources WSNs configuration.

Information retrieval of available services depends on the type of storage system. This type of network storage is one of the most important classification criteria, as it directly affects the performance of the SDPs in terms of scalability, mobility support and resource awareness [7] [11]. Depending on the type of network, various storage systems may be developed. For example, in ad-hoc networks, storage can be non-existent due to increased mobility, while wide area networks, it is compulsory to have intermediate storage, although it can increase substantially the complexity of the design.

It distinguishes the following main types of storage. First, Centralized approach is optimal in terms of the speed of data access and low traffic, although it creates a single point of failure. SLP with a Directory Agent [3] and Jini [18] are examples that have a central server for information storage. They use the server only for information retrieval, the actual communication between the client and the server being done in a peer-to-peer manner.

Second approach is Unstructured Distributed. In unstructured distributed storage systems, and relies on communication broadcast or multicast mechanisms[4]. This technique is common for protocols designed to work in local area networks and ad hoc networks. Several methods are used to obtain and maintain the service data. Service providers flood the network with service advertisements. Clients flood the network with discovery messages. Nodes cache the service advertisements. Nodes overhear in the network traffic and cache the interesting data.

However, the Structured distributed storage techniques are commonly found in the context of large networks [11], where the storage solutions mentioned earlier do not scale well. Directory nodes organize themselves in an overlay structure that allows them to route the discovery messages in a limited number of hops [17]. The structured distributed systems are classified into three categories: hierarchical, flat and hybrid.

The remaining contents of this paper have been organized as section 2 that describes the related research efforts of our problem. Proposed approachin section 3 , introduces our Modified Tiny Service Discovery Protocol (MTinySDP). Methodology and explanation of algorithms is presented in section 4. Simulation and results descussion is talked about in section 5. Finally, conclusion of our work is section 6.

2. RELATED WORK

In this section, we provide an extensive description of the existing service discovery protocols. Dynamic ad-hoc service discovery is one of the most intensively explored research topics in the past few years. Some protocols use registry-based approaches, while others discover services via multicast, query propagation, or distributed caching schemes. We outline a few service discovery protocols here. As will be seen, existing Service Discovery Protocols are too resource intensive, using a lot of network bandwidth and memory. Most of these are better suitable for Mobile Ad-hoc Networks (MANETs) and would require a substantial redesign to be suitable for WSN.

2.1 Service Discovery Protocols

Discovery protocols provide features to spontaneously lookup services with a low communication overhead. Note that, the connotation of service in the traditional context refers to a network service like a printer or ssh server; the traditional view of service discovery within a local area network is different from the usage scenarios in multi-hop wireless networks will be discussed.

2.1.1 Service Lookup Protocol (SLP)

An ETF (Internet Engineering Task Force) which provides a standard for network applications [3], tools for service discovery in a distributed environment. SLP is a tool for discovering services and advertising on IP networks and intensively uses TCP and UDP protocols. SLP allows applications to discover the existence, location, and characteristics of desired services and allows you to services to advertise their capabilities. Components of SLP are three main software objects (Agents). User Agent (UA) which performs service and discovery on behalf of client software. Service Agent (SA) which advertises the location and attributes on behalf of services. And Directory Agent (DA) which aggregates information about available services in the network.

SLP defines two modes of operation the system with (DA) and without (DA). Depending On presence or absence of (DA) in SLP, two separate architectures are possible, whose models. The presence of (DA) reduces the traffic in the network, since the multicast messages are avoided. The Disadvantage of an architecture with (DA) is the more complicated infrastructure, increased cost and degraded system reliability due to the dependence on introduced (DA's).

2.1.2 Sun’s Project Jini (Jini)

A Java-based framework for heterogeneous networks and software components [18]. Jini benefits from Java’s class loading capabilities to enable very flexible interface specifications for services. Service discovery is provided by the Jini Lookup Service, which resides on a designated node to which other nodes may send advertisements or lookup messages. Cascading of multiple lookup services is possible to enable discovery outside the local domain. Jini imposes several constraints on the participants as it relies on Java and Remote Method Invocation (RMI). Although Jini has often been termed an ad-hoc networking protocol because of its ability to spontaneously integrate new services and applications, it does not provide a suitable support for wireless sensor networks, because of its heavy weight RMI based protocols and its centralized structure.

2.1.3 The Universal Plug-and-Play (UPnP)

A set of protocols for service discovery under development by the Universal Plug and Play Forum [22], an industry consortium led by Microsoft (UPnP 2000). UPnP standardizes the protocols spoken between clients (called control points in the UPnP lingo) and services rather than relying on mobile code, like Jini. Device and service descriptions are coded in XML, and a number of protocols for local auto configuration, discovery, advertisement, client/service interaction, and eventing – Auto-IP, SSDP (Simple Service Discovery Protocol), SOAP Simple Object Access Protocol), and GENA (General Event Notification Architecture) are included in the specification. These protocols tend to be based loosely on existing standards such as HTTP .

2.1.4 Simple Service Discovery Protocol (SSDP)

It provides a mechanism that allows HTTP resources and HTTP clients to discover each other in a local area network [9]. It is used in Microsoft’s UPnP (Universal Plug and Play) architecture. When a service first comes online, it announces its presence in the network by sending a multicast message. All the clients that hear this

multicast cache the information. Any clients that come online after this announcement can discover the desired service by sending out a discovery request. As a result, there is no steady stream of messages being sent out. Only when an event occurs is the messages sent. This event-driven approach, however, makes the protocol more complex.

2.1.5 Gossip Based Discovery

this is a service discovery protocol for Mobile Ad-Hoc Networks [6]. A device collects information about services in the network by listening to all the service advertisement, request, and response messages in the network. Service advertisement is made periodically by sending a multicast message in the network. Other devices in the network cache this information. A service advertisement typically includes not only the services that the node itself provides, but also the services it knows that other devices provide. A service registry on each device is a tree-structured service classification with generic services at the top, and more specific ones as we move down the tree levels. It stores local services as well as services offered by others. This protocol updates the service registry based on every advertisement, request, and response message that it hears. This could be a challenge for resource constrained sensor networks.

2.1.6 The Geographic Hash Table (GHT)

A system for data-centric storage (DCS) on sensors [17]. Under DCS, Sensed the data is stored in the node specified by the sensed data related that name. GHT hashes keys to the geographical coordinates, and stores a pair key value when the sensor node geographically closest retailer whose key. Distributing the sensed data in the network using hashes offers its own advantages. It maintains persistence of data as nodes fail or move. In addition, it offers scalability. As the network size increases, the load of (key, value) pairs spreads evenly over the network.

2.1.7 The Pervasive Discovery Protocol (PDP)

Concentrates on service discovery in ad-hoc networks [5], where mobile devices communicate through wireless connections without any sturdy infrastructure. It is a fully distributed protocol that uses both push and pull techniques. In the PDP, a device announces its services only when other devices request the services. Service announcements are broadcast to all the devices within range, all of which will get to know about the new service simultaneously at that moment, without having to actively query for it. Devices have a cache where these announcements are stored. Service announcements include not just the services offered by the device itself, but also all the known services of the requested type.

2.1.8 Post Query Model

In the post query model, each server posts its serves to a set of nodes according to the posting protocol and each node queries its desired services according to the querying protocol. A pair of Posting and Querying protocols Considered a post query protocols. In order to cope with the topological changes in the time warp in an ad-hoc network, and the implementation of these protocols in rounds. As part of ad-hoc networks, the Posting and Querying protocols be independent. The proposed of all post query strategy defined in [12]. These include five post query strategies:

Greedy: all nodes post to all nodes and all nodes query all nodes of the network using the network broadcast mechanism. This strategy is non-adaptive and does not change with each round.

Conservative: based on the greedy strategy, we define a conservative strategy, this strategy requires that in each round all servers (nodes) post to (query from) all their one-hop neighbors using a one-hop broadcast communication mechanism. It is a non-adaptive, round invariant strategy.

Incremental: all servers and clients post to and query a small set of nodes in the first round and progressively increase of nodes posted to and queried from. The strategy aims at conserving valuable resources in the network.

Uniform Memoryless: in the strategy, servers post to a random set of nodes and nodes query a random set of all nodes.

Uniform with Memory: in the strategy, the nodes build a cache of nodes they visited and services they detect in the previous round.

2.1.12 The Tiny Service Discovery Protocol (TinySDP)

A lightweight protocol that allows discovery and selection of services in a network [2]. Services protocol allows for advertising availability clients and browses the network, determining in which services exist and the characteristics of those services. The network nodes need little or no static configuration for discovering services. The protocol supports a framework by which clients (or users) needing a service are able to contact those services. Clients are other sensor nodes or mobile devices like laptops, PDAs, etc.

The protocol uses a distributed approach in which information is cached in the network rather than in some central server. Any advertisements or requests are sent 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.

2.2 Distributed Service Discovery Protocol

Distributed Service Discovery Protocol (DSDP) is a distributed service discovery architecture, which is based on the virtual backbone to identifying and registering the services available within the dynamic network topology [9]. DSDP consists of the formation of a virtual backbone, as well as distribution of service registrations, requests, and replies. The dynamic virtual backbone is formed from a subset of the network nodes, Such that each node in the network are either part of the backbone or single hop far from at least one of the backbone of nodes.

The nodes in the virtual backbone act as service brokers and form a mesh structure that is interconnected by virtual links. Each non-backbone node is associated with at least one service broker in the backbone. Services have to be registered to at least one service broker in the backbone. When a node requests a service, it sends a request message to its service broker, wherefrom the messages is forwarded further in the backbone, which has the distributed knowledge of all available services in the network. Broadcasting of such messages would inefficiently waste network resources, which is crucial in shared wireless mediums.

F. Sailhan, et al. [8] the proposed service discovery protocol designed to broad mobile ad-hoc networks (i.e., comprising at least about 100 nodes). In this architecture discovery, Are distributed directories and publishing dynamically for scalability. The discovery architecture is structured as a virtual network. The virtual network is consists of a subset of the nodes of the MANET acting as directories. These directories represent a backbone of nodes responsible for performing service discovery. Is published even one directory at least inaccessible to most in a specified number of hops, H.

3. PROPOSED APPROACH

3.1 Overview

We introduce an alternative method to request and register services in TinySDP in order to provide a better responsively service discovery, which based on distributed directory in active mobility environment in WSN. In the modification of TinySDP, a node will retain only a partition of the service directory for its neighbor nodes as part of the overall service directory contains network services. Each node only contacted with three closest neighbors according to Choosing Neighbors Algorithm explained below. By connecting to three neighbors only, we acquire two advantages. First, this reduces number of transmitted messages in the network.

Second and most important, this will from a topology with at least six diagonal cells. After formation of the cell holding at least six neighbor nodes, a partial service directory will be built in each node, which makes the network operates as an distributed partial hex cells. An active process of monitoring mobility of the node is running periodically. The impact of this modification of TinySDP on mobility nodes is that the mobility node will automatically regulated within a hex cell as a part of the large network. Thus, partial service directory will cover only this part network. When a mobile node moves individually and want to discover a service, it does not need to send hundreds of messages to get to the node containing the service directory on the other side of the network. This node will council only nodes in this part of network.

Each node works as a service directory/provider at the same time. Thus, no needs to use a central with centralized directory. Mobile node only need to send messages to a limited number of adjacent nodes that are potential hold information to get the services of registered in the node directory on the part of the network. This limitation will improve the hit rate of access to the service and improves the performance of the network. Thus, we get better performance while reducing the messaging overhead on network due to updates of the continuously moving nodes.

Our modification over the classical TinySDP protocol is based on the principle of multiple behavioral modes of operations by which modification TinySDP will be able to change its behavior in different situations. The modification over mobile TinySDP considers five main points:

- First, to monitor the network topology and continuously changes the behavior due to node mobility and to process of sending and receiving messages in the mobile network.
- Second, to be aware of vicinity neighbors and network topology that allows it to benefit from both the push and pull mode.
- Third, to introduce a new mechanism for sending services advertisement between the mobile nodes in the network.
- Fourth, to specify the location of nodes and the network area also choosing the best routing and shortest path in order to identify the nearest three neighbor nodes to create a hex cell that going to exchange the directories.
- Fifth, to consider three operational mode, ADVERTISING mode, UPDATING mode, and IDLE mode, to control the service advertisement continually among the node.

3.2 Functionalities and Improvements

3.2.1 Functionalities

The functionalities of Modification of TinySDP have been added to MTinySDP in order to be applied on MWSN where each node works as a service directory provider. Hence, all mobile nodes within the network are involved in the process of updating, advertising, choosing neighbors and service requesting. The functionalities can be presented as the adaptation of Service request/replay will be indirectly enhanced by integrating distributed partial service directory to a three number of adjacent nodes, rather than a random trajectory advertisement in TinySDP, which increases the total number of messages, sent across the network and reduces success rate.

The mobility of the nodes of the network forces us to continuously monitor the state of the network service directory. Therefore, it is required to develop a method to update the information periodically. In TinySDP the continually monitoring process exists only in the startup of the network. Add a new metric for calculations the physical distance between the nodes. This leads to the selection of the three closest neighbors of node by SDP_NBR metric, knowing that those two metrics serve a mobile environment for the existence of significant changes verses constant movement of the nodes. However TinySDP does not take into account this point.

The Declaration of the services is periodically according to the three operating modes proposed ADVERTISING, UPDATING, and IDLE Mode, unlike the TinySDP where the adverts were in network on startup or in case of adding a new node, so this not suitable mobile network. Maximizing the utilization of push and pull mode in hybrid way. In which new neighbors will be pushed to this table by TBF routing protocol. With push mode, service is pushed from the predefined three closest nodes. Periodically, with pull mode, advertisements are gathered from the predefined three closet nodes especially when we use that mode together.

3.2.2 Improvements

By improving the hit rate of access the service, performance of the network will improves. As an aim, reduces the load on the network because of the contract messaging updates are constantly moving. The improvements will be as follows:

1. **Service Advertisement**

Time to advertise in old Approach: To start at network startup and When a new node enters the network. While in new Approach: Periodically according to Modes Switching Algorithm. Advantages are to be aware of the mobility of neighbor nodes and to keep advertisement table up-to-date periodically.

2. **Service Request**

Time to request in old Approach is on demand, when a node needs to find a service, it sends out a service request packet. While in New Approach, To define a process for service request for each of the neighbors. Service requests will be executed in separated parallel process. Service request process will not intersect with updating and advertising, which make MTinySDP operates in multiprocessing mechanisms. Advantages are to add multiprocessing, will increases the availability of service request/reply and service request/replay will be indirectly enhanced distributed partial service directory.

3. **Service Reply**

In old Approach, when a node receives a service request, it checks if it meets the service requirements of the requesting node. While in New Approach, Service replay will be executed in separated parallel process. This makes MTinySDP operates in multiprocessing mechanisms.Advantages are to add multiprocessing will increase the availability of service request/reply and Service request/replay will be indirectly enhanced distributed partial service directory.

3.2.3 Routing Resolution

TinySDP is a multi-hop protocol. A routing scheme was needed that would help in the service discovery process. Although the service discovery is a middleware protocol, lying above the network layer, we argue that some level of TBF routing support is necessary for service discovery to be efficient. We will make a few assumptions due to the variety of manufactures, mote fabrication technics in addition to the conditioning of the motes according to the target environment. These assumptions consider routing the packets:

- All nodes know their location with respect to a coordinate system. This information can be obtained by use of some positioning system [21].
- A neighbor discovery protocol runs in the lower layers. So all the nodes are aware of who their neighbors are and their location [7].
- The nodes have an approximate sense of directionality. This is useful, when a node initiates a service request or service advertisement [15].
- All nodes can find the coordinates of the destination node from a location service [13], [21].
- All nodes in WSN use Trajectory Based Forwarding (TBF) as routing protocol.
- TBF protocol is a hybrid of source-based routing and Cartesian forwarding.
- TBF protocol removes the overhead at each node to maintain and update the routing information.

Since routing in a WSN without this basic knowledge is nontrivial, we believe these assumptions are reasonable. We use Trajectory Based Forwarding (TBF) as our routing protocol. It is a hybrid of source-based routing and Cartesian forwarding [16]. Like source-based routing, the path is indicated by the source, but without actually specifying all the intermediate nodes. Such as Cartesian forwarding, and the decisions taken at each node are greedy, but do not depend on the distance to the destination – the measure is the distance along the desired trajectory. TBF removes the overhead at each node to maintain and update the routing information. In addition, it trades off computation for communication. Considering the four orders of magnitude difference between the cost of sending a wireless, packet and executing an instruction [20], TBF saves precious resources.

3.2.4 Push / Pull Mode Mechanism

Originally, the TinySDP uses both push mode and pull mode. The new approach defines a mechanism to reach full utilization. Mobile node will utilize both of them and switch one to another. In Push Mode, Mobile node uses push mode to advertise services and to send both service table and neighbor table. The node is informing its neighbors about its own provided services. According to our algorithm, the three closest nodes neighbors will receive information through pushes. Figure (1) shows information exchange with push mode.

While in Pull Mode, Mobile nodes uses pull mode to request the advertisement table and neighbors table. The node is asking its neighbors about service provided by them and their neighbors. According to our algorithm,

the three closest nodes neighbors will send information through pulls. Figure (2) shows information exchange with pull mode.

3.2.5 Distributed Partial Services Directory

Each node works as a service directory/provider at the same time. Thus, no needs to use a central with centralized directory. Each node will preserve a service directory contains the advertised services of only its neighbors. Implicitly, the Service Directory is partitioned and distributed through nodes. The structure of service Advertisement table is storing information of services. All other structures are described as follows:

1. **Service Table:**

- The service table maintains the list of services that the node provides.
- The table stores some standard values for a service (time duration, maximum, minimum, or average value).
- The services table pushed to three closest nodes neighbor by push Mode as shown in figure (1)

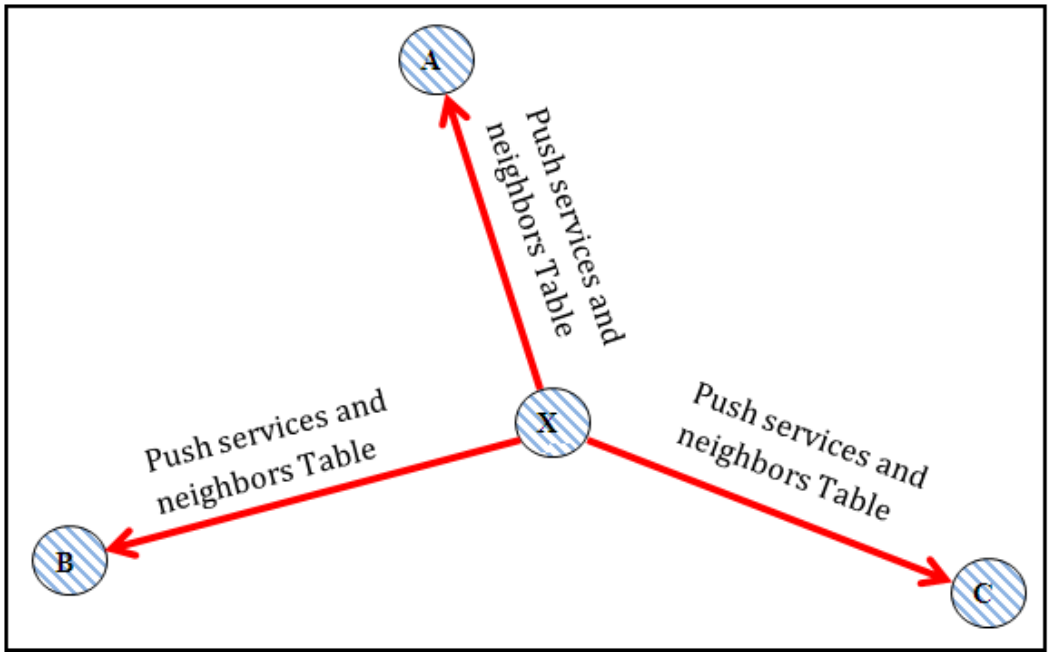

Figure (1): Push Mode Mechanism.

2. **Neighbor Table:**

- The table contains all neighbors' addresses.
- The next node is chosen from this table per the rules of the routing protocol.
- For TBF, the node finds the neighbor towards the destination along a trajectory and then sends the packet.
- The neighbor table pulled to three closest nodes neighbor by pull Mode as shown in figure (2).
- Implicitly, each node knows how to contact its neighbors and neighbors of neighbors and so on.

3. **Advertisement Table:**

- The advertisement table contains the latest advertisements that the node has pulled.
- Periodically, each node pulls advertisements from its three neighbors only.
- Advertisement table is pulled from the three closest nodes neighbors by pull mode as shown in figure (2).

3.2.6 Update Mechanism for Service Directory

1. **Service Table:**
- Only when new service is added to this node. This service will be available after adding.

2. **Neighbor Table:**
- New neighbors will be pushed to this table by TBF routing protocol.
- TBF registers all Cartesian forwarding.

3. **Advertisement Table:**
- Periodically, with pull mode, advertisements are gathered from the predefined three adjacent nodes.
- Considering that, these three nodes performed service directory updates before separately.
- Eventually, a node will get advertisements from neighbors, neighbors of neighbors, and neighbors of neighbors of neighbors and so on.
- No duplication advertisement will be registered.
- The advertisement will be deleted automatically after it expires.

3.2.7 Hex Cell Formation

- A separated parallel process for monitoring mobility is active all the time.
- When a node wants to request a service, it sends service request to the three neighbors only.
- The node selects closest neighbor node by Cartesian position and asks to join its cell.
- If this neighbor has already connected to three nodes, then rejection will be sent back.
- In this case, the node will try next closest neighbor node by Cartesian position.
- The node will try until receiving acceptance to join on cell.

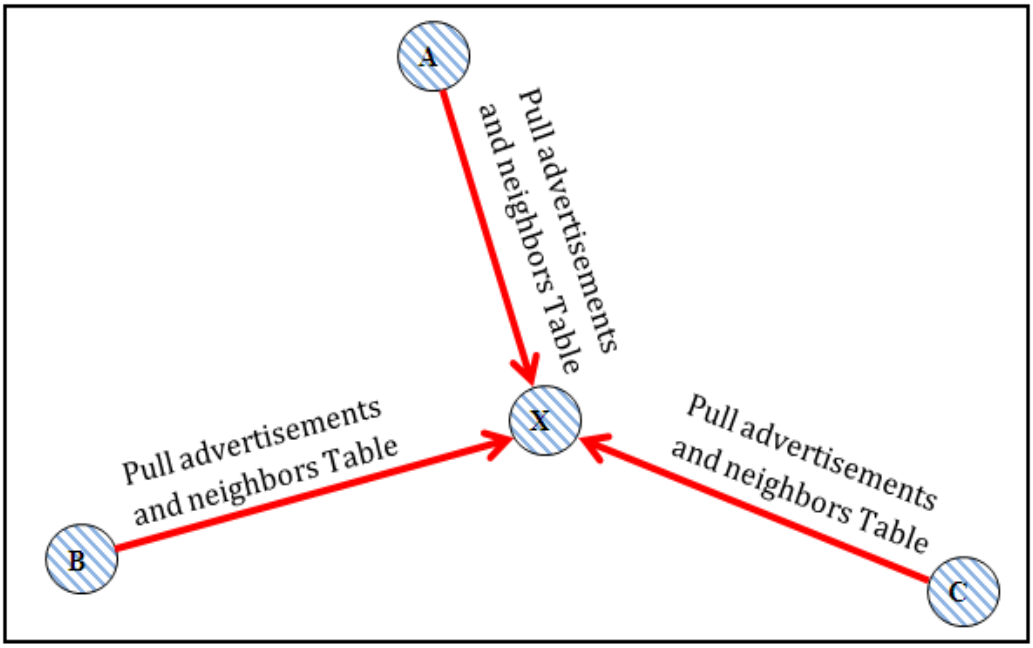

Figure (2): Pull Mode Mechanism.

4. METHODOLOGY

4.1 Distance Calculations

A distance metric is a concept of physical distance. The metric is a function that is acting according to the specific set of norms, It is a concrete way to describe what it means to some elements of the area of be "close to" or "far away from" each other. In most cases, "distance from (A) to (B)" Is alterable with "distance between (B) and (A)". Select the standard Euclidean distance can be square in order to progressively greater weight placed on the objects that are far away from each other. The following equation explains the previous.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Squared Euclidean Distance is not a metric, as it does not satisfy the triangle inequality. However, often it is used in optimized problems which distances only have to be compared. It is also referred to as quadric within the field of rational trigonometry. Euclidean Squared use of Euclidean square of the distance in the cases that will be used in ordinary Euclidean distance Jarvis Patrick or K-Means assembly [15], [24].

4.2 Choosing Neighbors

SDP_NBR metric represents the count of closest neighbor nodes. Each node selects closest three neighbor nodes to send service advertisements message, as shown in Figure (3). A node sends an advertisement to its neighbor, if the advertisement table of the selected neighbor has already three previous advertisements, then the source chooses the next closest neighbor, depending on the number of hop no more than five in such a case to are configured Hex-Cell. This prevents the construction of tri-cell network, as shown in Figure (3). But allows the construction of Hex-Cell Network [1] and can be built using hexagon unit cells, each of the six nodes, as shown in Figure (4).

Thus, a Hex Cell network will compose in that case. A node sends an advertisement to its neighbor, if the advertisement table of the selected neighbor has already three previous advertisements, then the source chooses the next closest neighbor. This prevents the construction of tri-cell network, as shown in Figure (3). Thus, the network represents an interesting network topology with features such as the capability of embedding several topological structures and extensibility at a minimal cost, which can utilize the capability of clustering, by splitting the advertising mechanism into two levels. The first: main node associated with the three other nodes (the nodes linked), which we call the node advertises. The second: Any external node you want to do the announcement (collected between nodes) must bind itself with a pool of the contract, and so you want to call it a node discover services request.One of the main advantages of the proposed advertising mechanism is that it does not need readdressing for nodes as the number of nodes increased. Moreover, it requires less computational power than the current service discovery.

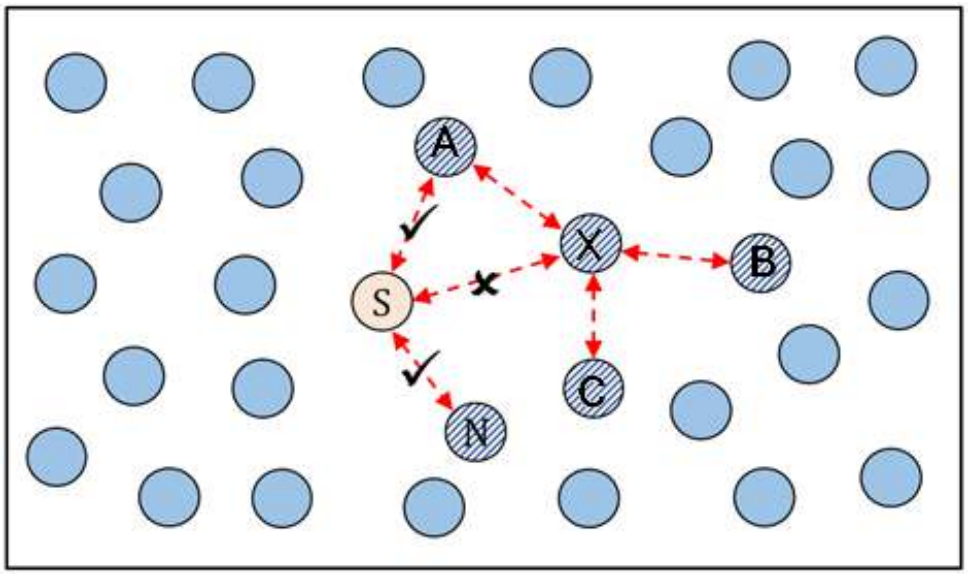

Figure (3): Three Closest Neighbor Nodes and Network Topology.

4.2.1 Choosing Neighbors Algorithm

Algorithm : Choosing Neighbors Algorithm (SDP_NBR)

```
1. for up to MAX_HOP_COUNT
2.     for each neighbor
3.             select closest neighbor node (N)
4.                 if node (N) is already added to SDP_NBR
                       select next closest node (N)
5.             else
6.                     add node (N) to SDP_NBR
7.             end if
8. select next neighbor
9. end for
```

For example in figure (3), node X chooses three neighbors by our algorithm. Node X will chose node A, node B and node C. Node X will check whether these nodes is included in its neighbor table or not. Also, by consulting the previously connected nodes, node X asks its neighbor to check connectivity with potential node before connecting to it to avoid duplication. When node S wants to be connected to node X, consulting with neighbors up to max hop count MAX_HOP_COUNT of 5 to check if it is already connected to a neighbor. Then next potential node is checked and so on until at least a hex cell diagonal is formed as shown in figure (4).

4.2.3 Operational Modes

The basic idea of Operational Modes is to adjust the contract dealt with each other about what is sent from or with what is being received from the busy network services quantity of messages. These modes are UPDATING, ADVERTISING, and IDLE Mode. Modes are not mutual exclusive. This means that a node can only work in only one particular mode at a time. There by requiring a set for that process to preserve the network resources.

1. **UPDATING Mode:**

   In this mode, the node chooses the three closest neighbors, whereupon calls update neighbor table and advertisement table of its neighbor nodes by pull mode. The yellow Led is blinking in as an indication that this node behaves in UPDATING mode.

2. **ADVERTISING Mode:**

   In this mode, a node sends and service advertisement to pre-selected three neighbors. The node in this mode acts proactively and sends the advertising messages for the services provided to its neighbor nodes by push mode. The red Led is blinking in as an indication of that this node behaves in ADVERTISING mode.

3. **IDLE Mode:**

   In this mode, a node stops sending or receiving advertisements or updates and does nothing. We consider this mode as a power saving mode. There is no led indication for this IDLE mode. When a service request is required, a service request is issued through the network or by node itself. Nodes

are prevented to handle service requests but in this mode. The green Led is blinking in as an indication of that this node behaves in IDLE mode.

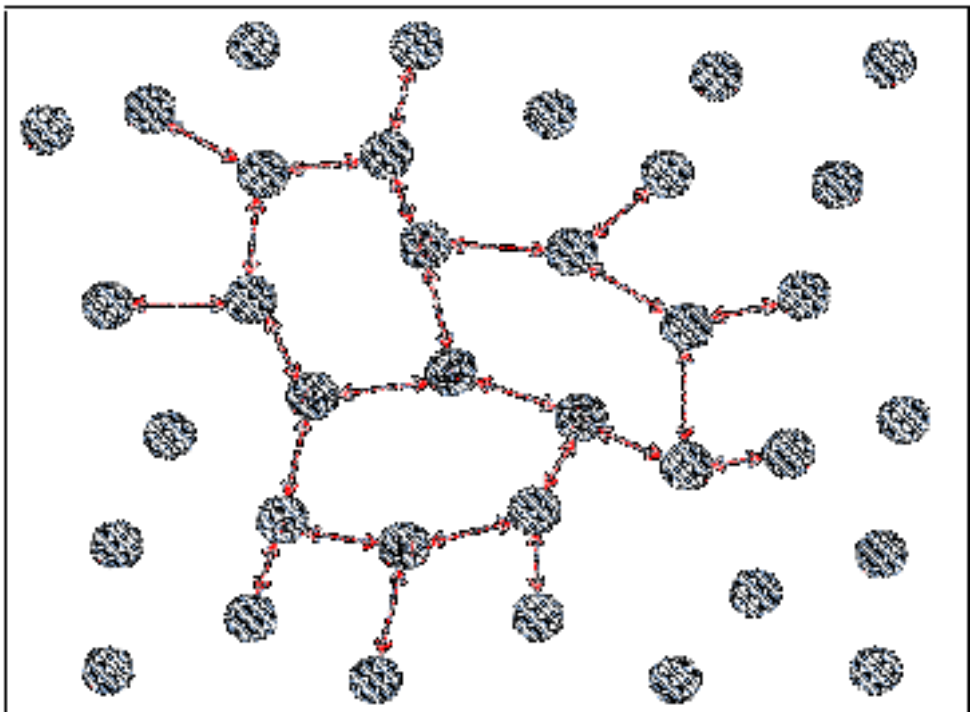

Figure (4): Hexagon-Cell of Network

4.2.4 Modes Switching Algorithm

When switch timer is triggered, mode switching algorithm executed. Three main subtasks are to be performed. *Choosing neighbors, update* and *advertise* are executed in consecutive order. In UPDATING Mode, node chooses the three closet neighbors are and stores them in SDP_NPR. This is considered a new approach. In the old approach, the node can send to all nodes. Unlike the old approach, we preserve the state of three neighbors. *Update* is also performed in UPDATING Mode. After *update* is finished, the node enters ADVERTISING Mode and perform *advertise* to only three neighbors form SDP_NPR. Then node enters IDLE Mode to send and receive services.

4.3 Service Discovery

We tried to reduce the advertisement and discovery messages to a minimum. In order to optimize sending and receiving messages through the network, the topology is preserved. Service request is directed only to three previously chosen neighbors. If the node cannot address the target node directly, then the request is forwarded to three neighbors in order to reach the node that provides the service or have service announcement. Consequently, it will lead to access to submitted in the node of the service is fast and the shortest path, which leads to reduced cost and time, and not flooding the network a tremendous amount of messages, as shown in Figure (5).

Algorithm : Modes Switching Algorithm

```
1.  For Switch Timer do
2.      If node not in UPDATING Mode
3.          Choosing Neighbors
            update
4.      If node not in ADVERTISING Mode
5.              Node Mode = ADVERTISING
6.              perform advertising routine
7.          EndIf
8.          If node not in IDLE Mode
9.              Node Mode = IDLE
10.             perform service Request / Receive
11.         EndIf
12.     EndIf
13. End timer Treset
```

4.4 Evaluation Metrics

To analyze the results, we will use metrics defined in [51]. For referring to the Total Number of Nodes (N), the number of clients that successfully locate the service as (Nsucc), the number of service advertisement messages as (Madv). The number of service request messages as (Mreq), The total number of messages sent as (Mtotal), each successful client C receives from 1 to m service reply messages of waiting time t_1, t_2,...,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:

$$AWT = \frac{\sum_{n=1}^{N} \min(t1, t2, \ldots, tm)}{N}$$

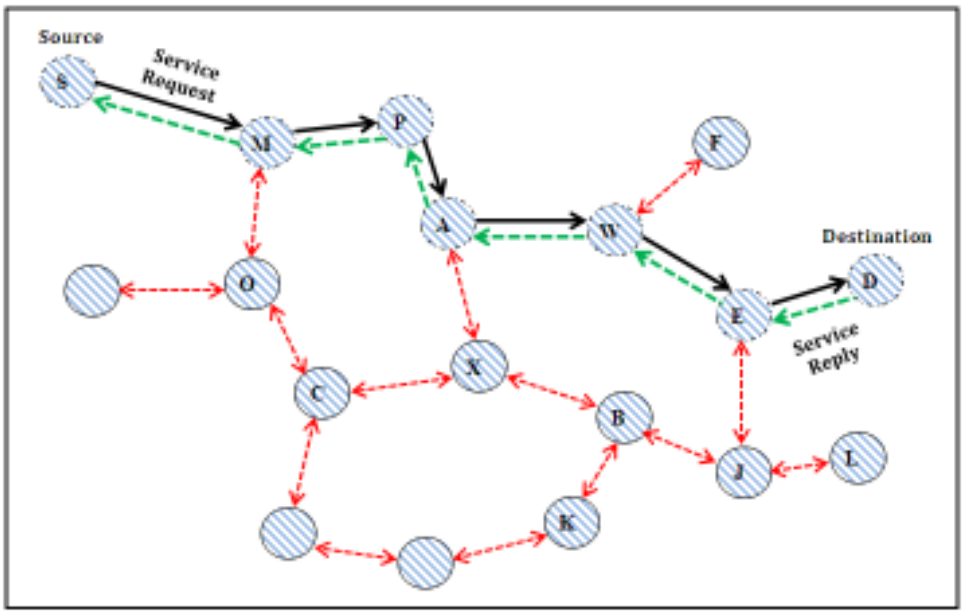

Figure (5): Services Discovery and Trajectory Packet in the Network.

5. SIMULATION AND DISCUSSION

In this section, we discuss the simulator used for this paper, the simulation environment and draw conclusions from the results. In this paper, we use TOSSIM [14], 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 in high fidelity while raising thousands of nodes. TOSSIM compiles directly from the code TinyOS, and is compiled automatically by default (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 Setup

All 50 simulated nodes are assumed to have the same resources in terms of memory and power. The experiment to study modified effect and to highlight on results. There are eleven types of services provided 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 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 Setup

Each one minutes of the real time of a node, the node starts executing MTinySDP. Switching between on operational mode to another according to metric calculations, advertisements are sent out when the network. Since we have 50 nodes in our network, a few of the service request packets get lost if all of the 50 nodes with MTinySDP in mobile WSN start sending out their service request together at network startup.

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. The Simulator runs for a time of 100 minutes to get the best reading of sending out request packets. The parameters are listed in Table (1).

5.3 Results Analysis and Discussion

In this section, we analyze and compare our results of the MTinySDP with Mobile WSN and TinySDP with Mobile WSN and TinySDP with static WSN and post-query protocol. We have used the results of the post-query protocol as specified in their paper [12]. A comparison of performance results shown in Table (3) with 50 nodes. The main result showed that MTinySDP having the best values in most performance measures, compared with results in Mobile WSN in terms of the Total Number of Transmitted Messages (Mtotal), and higher values in Average Waiting Time (AWT) for discovering service is a measure of the high performance metrics of MTinySDP.

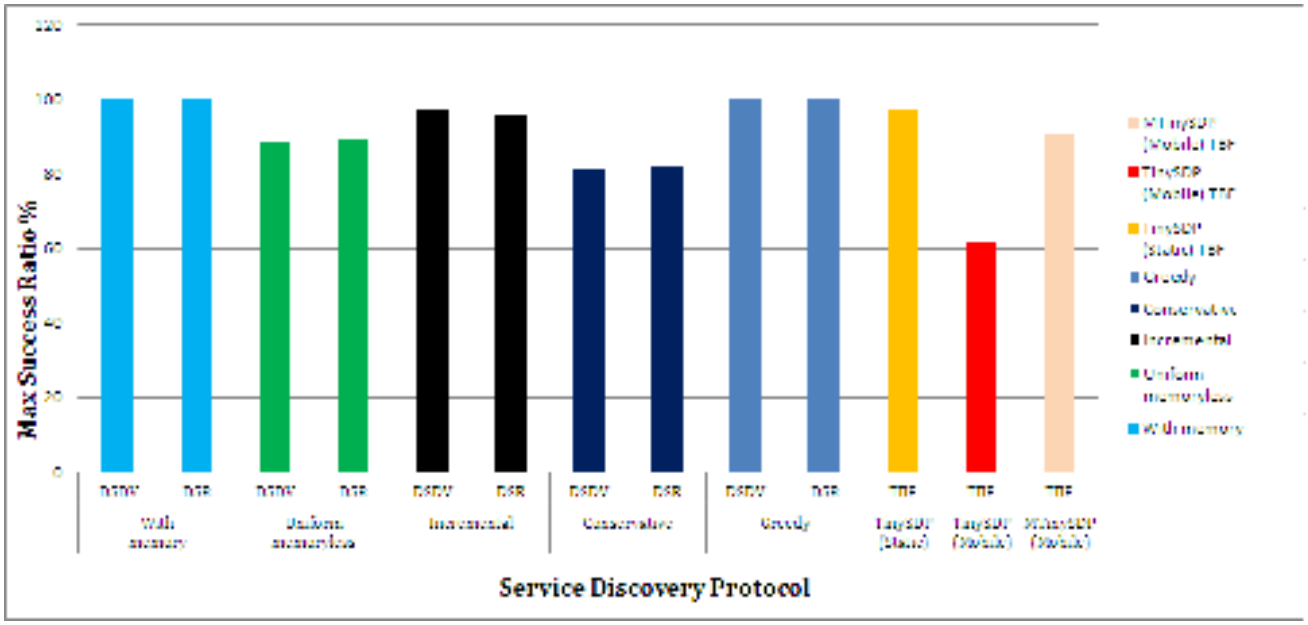

Figure (6) Success Ratio (SR) of MTinySDP and TinySDP with MWSN and Static WSN and post-query strategy.

The table also shows that the strategies Greedy and memory excelled in the values of the metric success rates on the rest of the other protocols. Results are shown in Table (3). From results we see that Mobile WSN have less success rates in the most results of the above-mentioned criteria. In addition, the metric obtained by MTinySDP embodied efficiency of the performance and success rates high compared with the rest of the other protocols. Finally, we argue that that the *Routing Protocol* (*RP*) used for TinySDP and MTinySDP *Trajectory Based Forwarding* (*TBF*) has less overhead in terms of memory and bandwidth requirements than either *Dynamic Source Routing* (*DSR*) [61] or *Destination Sequenced Distance Vector* (*DSDV*) [23]. The performance of the five Post-query strategies when combined with the *DSR* protocol and *DSDV* protocol.

Table (3): Performance Comparison.

Strategy	Routing Protocol	Max SR	M_{total}	AWT
MTinySDP (Mobile)	TBF	91%	506	0.3502
TinySDP (Mobile)	TBF	62%	611	0.3518
TinySDP (Static - Original)	TBF	97%	541	0.3510
Greedy	DSR	100%	16215	7.50
	DSDV	100%	16557	4.93
Conservative	DSR	82.25%	2909	36.55
	DSDV	81.5%	2122	37.62
Incremental	DSR	96%	2900	79.72
	DSDV	97%	2898	83.23
Uniform memoryless	DSR	89%	2602	54.55
	DSDV	88.25%	2596	59.27
With memory	DSR	100%	2845	46.96
	DSDV	100%	2816	51.89

We visualized results in three variant figures. Performance comparison of all metrics and then Success Ratio, Average Waiting Time and Total Number of Messages Sent between MTinySDP and TinySDP with Mobile WSN and post-query protocol. In Figure (6) illustrated that the greedy strategy and with memory strategy was better and significantly compared with MTinySDP and TinySDP with MWSN and with static WSN in overall comparison in results of Success Ratio (SR). The modification implemented on MTinySDP became able to approach multiple methods of behavioral processes.

Through the ability to change behavior in different situations in order to achieve the best results, and to reduce the backlog in communications, and to improve access to the required service in the most economical way to not waste network resources. The modified MTinySDP regulates the transmission of messages on the network and reduces the number of unwanted messages in the network. Node does not need to send hundreds of massages to discover or to advertise services. However, only send advertising will be restricted to a limited number of adjacent nodes, which may be registered in the neighboring nodes in the network directory. Therefore, to improve the efficiency of MTinySDP MTinySDP in a mobile network results superiority over other strategies in most performance metrics as in Table (3).

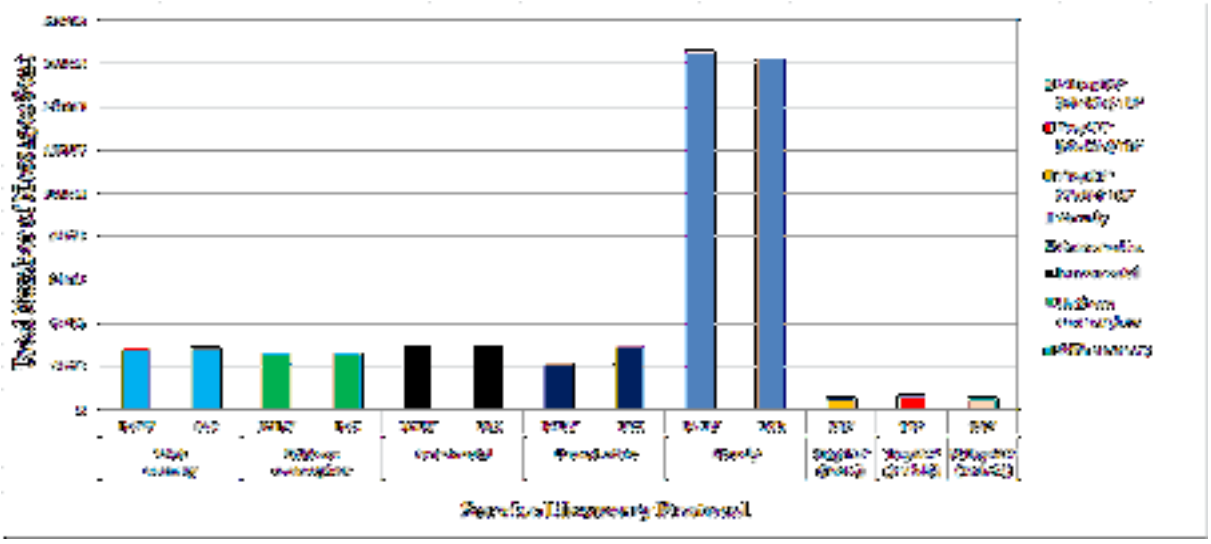

Figure (7): Performance Total Number of Messages Sent of MTinySDP and TinySDP with MWSN and Static WSN and post-query strategy.

In Figure (7), Total Number of Messages Sent (Mtotal) between MTinySDP and TinySDP with MWSN and static WSN and post-query protocol strategy has a clear gap. It is clear that significant progress in results of MTinySDP was better and significantly compared with other results. Figure (8) describes the relationships between the Average Waiting Time (AWT) from the message transmission service request and the number of nodes in MTinySDP and TinySDP with MWSN and TinySDP with static WSN and post-query strategy. As illustrated in Figure (8), the AWT of MTinySDP is much better than all another strategy. Therefore, MTinySDP with MWSN, have a direct relationship to the value of the Average Waiting Time AWT with the number of nodes in the network. The modification gave a speed in reply message to service requests.

6. CONCLUSION

Our objective is to do modifications on TinySDP to solve the actual problems when applying that protocol on mobile network. In mobile environment, protocol has to respond with continuous changing topology features. Hereafter, the application of TinySDP on mobile network had presented continuous changes of nodes position and progressive change of topology. Therefore, it had become an obligation to improve TinySDP to be MTinySDP as an outcome of that process when it is applied on MWSN.

Nevertheless, the measurement of MTinySDP performance within the three major metrics: Success Ratio, Number of Total Transmitted Messages, Average Waiting Time is considered by comparing between other strategies and using a network of 50 nodes. Referentially, by comparing the results of 50 nodes, greedy strategy and with memory strategy was better and significantly compared with MTinySDP and TinySDP with MWSN and with static WSN in overall comparison in results of Success Ratio (SR). Furthermore, using Number of Total Transmitted Messages as another metric to be compared, the results of MTinySDP had shown that the number of Mtotal is extremely less than TinySDP and post-query strategies.

The reflection that the new advertising mechanism has been credited with the ability to get a limited number of total messages sent through the network. Moreover, Average Waiting Time as another metric to be compared in investigating MTinySDP qualification, AWT had presented less records of its presentation in TinySDP and post-query strategies. Consequently, the compared results had shown a qualified performance of MTinySDP. In other words, the added qualifications of TinySDP had ultimate role in achieving that highlighted performance of MTinySDP.

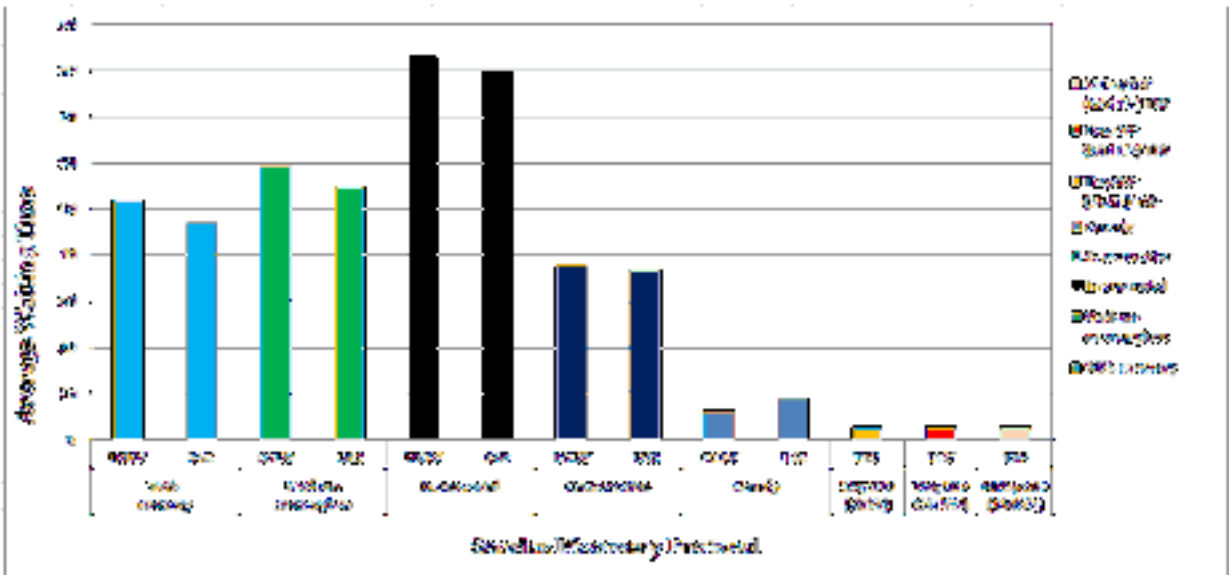

Figure (8): Describes the relationships between the (AWT) of MTinySDP and TinySDP with MWSN and Static WSN and post-query strategy.

REFERENCES

[1] A. Sharieh, M. Qatawneh, W. Almobaideen, and A. Sleit. "Hex-Cell: Modeling, Topological Properties and Routing Algorithm". European Journal of Scientific Research, ISBN: 1450-216X, Volume 22, Issue 2, Pages:457-468, 2009.

[2] 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.

[3] 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.

[4] S. Baek, E. Choi, and J. Huh. "Sensing Information Managing Mechanism for Context Aware Service in Ubiquitous Home Network". IEEE Consumer Electronics Journals, Pages: 1393-1400, November 2007.

[5] 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, Volume 50, Issue 17, Pages: 3264–3283, December 2006.

[6] C. Lee, S. Helal, and W. Lee. ”Gossip-Based Service Discovery in Mobile ad-hoc Networks”. IEICETransactions on Communications, September 2006.

[7] Madan, R. and S. Lall. "An energy-optimal algorithm for neighbor discovery inwireless sensor networks", Mob. Netw. Appl. 11 (3), 317–326, 2006.

[8] F. Sailhan, V. Issarny: Scalable Service Discovery for MANET, Proceedings of the 3rd IEEEInt'l Conf.on Pervasive Computing and Communications (2005).

[9] P. E. Engelstad, Y. Zheng: Evaluation of Service Discovery Architectures for Mobile AdHoc Networks, the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS’05) (2005).

[10] C. Frank, V. Handziski, and H. Karl. "Service Discovery in Wireless Sensor Networks". TKN Technical Reports Series, TKN-04-006, and March 2004.

[11] F. Bai, A. Helmy, “A Survey of Mobility Modeling and Analysis inWireless Adhoc Networks” in Wireless Ad Hoc and Sensor Networks,Kluwer Academic Publishers, 2004.

[12] H. Luo and M. Barbeau, “Performance evaluation of service discovery strategies in ad hoc networks”. In Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR 04), pages 61–68, 2004.

[13] Niculescu, D, B. Nath. "Localized positioning in ad hoc networks". SensorNetwork Protocols and Applications, April 2003.

[14] P. Levis, N. Lee, “TOSSIM: A Simulator for TinyOS Networks”, Computer Science Division, University of California Berkeley, California, September 2003.

[15] M. Mount, S. Netanyahu, and D. Piatko. "An Efficient k-Means Clustering Algorithm: Analysis and Implementation". IEEE Computer Society, Volume 24, Issue 7, Pages: 881-892, July 2002.

[16] Niculescu, D. and B. Nath, "Trajectory based forwarding and its applications". Technical Report DCS-TR-488, Department of Computer Science, Rutgers University, May 2002.

[17] Ratnasamy, S., B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A geographic hash table for data-centric storage in sensornets.Proceedings of the First ACM International Workshop on Wireless Sensor Networksand Applications (WSNA), September 2002.

[18] Jini(tm) Network Technology: "Devices, Desires, and Designs". IEEE Computer Society Washington, ISBN: 0-7695-1050-7, Pages: 3-687, July 2001.

[19] Niculescu, D. and B. Nath (2001). Ad hoc positioning system (APS). In GLOBECOM (1), pp. 2926–2931.

[20] Warneke, B., M. Last, B. Liebowitz, and K. S. J. Pister. "Smart dust: Communicating with a cubic-millimeter computer". Computer 34 (1), 44–51, 2001.

[21] J. Li, J. Jannotti, D. D. D. K. and R. Morris. "A scalable location servicefor geographic ad hoc routing". In Proceedings of ACM MobiCom (Aug. 2000).

[22] UPnP Forum. UPnP device architecture version 1.0, June 2000. Available online at http://www.upnp.org/.

[23] C. Perkins and P. Bhagwat. “Highly dynamic Destination-sequenced distance vector routing (DSDV) for mobile computers” ACM SIGCOMM '94 p234-244, 1994.

[24] A. Jarvis, and P. Edward. "Clustering Using a Similarity Measure Based on Shared Neighbors". IEEE Transactions on Vehicular Technology, Volume 22, Issue 11, Pages: 1025-1034, November 1973.