Performance evaluation and comparison study of adaptive MANET service location and discovery protocols for highly dynamic environments

Ghassen Ben Brahim[1], Nazeeruddin Mohammad[1,2], Wassim El-Hajj[3*] , Gerard Parr[4] and Bryan Scotney[5]

*Correspondence:
we07@aub.edu.lb
[3] Computer Science
Department, American
University of Beirut, Beirut,
Lebanon
Full list of author information
is available at the end of the
article

Abstract

A critical requirement in Mobile Ad Hoc Networks (MANETs) is its ability to automatically discover existing services as well as their locations. Several solutions have been proposed in various communication domains which could be classified into two categories: (1) directory based, and (2) directory-less. The former is efficient but suffers from the amount of control messages being exchanged to maintain all directories in an agile environment. However, the latter approach attempts to reduce the amount of control messages to update directories, by simply sending broadcast messages to discover services; which is also a non-desirable approach in MANETs. This research work builds on top of our prior work (Nazeeruddin et al. in IFIP/IEEE international conference on management of multimedia networks and services, Springer, Berlin, 2006)) where we introduced a new efficient protocol for service discovery in MANETs (MSLD); a lightweight, robust, scalable, and flexible protocol which supports node heterogeneity and dynamically adapts to network changes while not flooding the network with extra protocol messages—a major challenge in today's network environments, such as Internet of Things (IoT). Extensive simulations study was conducted on MSLD to: (1) initially evaluate its performance in terms of latency, service availability, and overhead messages, then (2) compare its performance to Dir-Based, Dir-less, and PDP protocols under various network conditions. For most performance metrics, simulation results show that MSLD outperforms Dir-Based, Dir-less, and PDP by either matching or achieving high service availability, low service discovery latency, and considerably less communication overhead.

Keywords: Service discovery protocol, Distributed directory-based approach, Mobile Ad hoc networks, Auto-configuration

1 Introduction

Ad hoc networks continue to draw a lot of attraction to the research community due to the exponential increment in the number of portable handheld devices such as tablets and smartphones, which are equipped with wireless technology such as Bluetooth, UWB, WiMax, WiFi, Zigbee, 5G, and LTE making it possible for these devices

to communicate[16]. When mobility is considered, the communication is referred to Mobile Ad Hoc Networks (MANET). MANET is also considered as the key enabler to an efficient and reliable Internet of Things due to its ability to establish networks and therefore allow mobile nodes to wirelessly communicate without any pre-planning phase [15]. Effectively utilizing the services of network elements is a major design goal for MANETs and was the impetus toward developing reliable protocols for network service discovery (SD). These protocols will be utilized for advertising, selecting, locating, accessing, and picking-up the desired services [25].

Services on a device include accessible software components, hardware components, storage, processing, and data which other devices may need. For instance, in military communications, data from several heterogeneous devices carried by warfighters should be combined and analyzed to make accurate real-time decisions. These resource constrained devices dynamically look for available mobile cloudlets [2, 3] in MANETs to delegate computationally intensive tasks.

Given the several inherent limitations of mobile devices and wireless networks, enabling service discovery is a challenging task.

In general, most of the SD protocols for MANETs attempt to address the following five design goals: (1) reduced control message exchange in a resource constrained environment; (2) ability to scale for large networks with a variety of large number of applications as well as devices; (3) high level of robustness in terms of the ability to cope with the loss of connectivity due to the high mobility or failure of nodes; (4) increased level of versatility in terms of the ability for SD protocols to properly operate in the presence of a large diversity of devices; (5) highly interoperable in the scenarios where MANETs are connected to fixed networks via gateways and the MANET SD protocol is supposed to transparently discover services on the Internet and allow Internet devices to discover services on the MANET.

This paper builds on top of our previous work where we proposed a new service discovery protocol for MANET environment (MSLD)[1], which is integrated with DHAPM—a stateful auto-configuration protocol [5]. It provides an extensive performance evaluation of MSLD under various network conditions as well as a comparison study with three different SD approaches: Dir-Based, Dir-less, and PDP. Although for demonstration purposes we have integrated with DHAPM, MSLD can be tuned to work with any directory based auto-configuration protocol.

This paper has the following main contributions:

- It reviews the existing Service Discovery protocols in three different contexts: infrastructurebased networks, MANET, and IoT.
- It highlights the main MSLD design features.
- It compares and analyses the performance of MSLD in terms of Latency, Service availability, and overhead messages with that of Dir-Based, Dir-less, and Pervasive Discovery Protocol (PDP). This performance evaluation study was conducted under various network conditions including: network density, message discovery requests rates, network servers' redundancy, and cache life time.

Table 1 Typical service directory

Service name	Service URL (attributes and providers)
Printer	host2@color = false;duplex=true, host1@color = true;duplex = false
Camera	host3@fps = 10, host7@fps = 20;zoom = true, host15@fps = 15
Gateway	host10@type=satellite;bandwidth = 2 Mbps
.....	.....
Storage	host5@size = 100 GB, host9@size = 1000 GB

The rest of the paper is organized as follows. Section presents a summary of the major components and features of MSLD protocol. Section discusses the complexity of MSLD in terms of control messages. Section covers the simulation setup, performance evaluation, and comparison with Dir-based, Dir-less, and PDP. Section gives a brief overview about the literature related to Service Discovery Protocols in three different contexts: infrastructure based network, MANET, and IoT. Section concludes the paper and sets directions for future work.

2 Method—MSLD protocol design

MSLD was designed to achieve: (1) robustness against directory failures by considering distributed directories and (2) low communication overhead, through its integration with a stateful autoconf protocol. The rest of this section describes the details of MSLD design approach in terms of its integration with one of the existing stateful autoconf protocol (DHAPM) and the major MSLD building blocks.
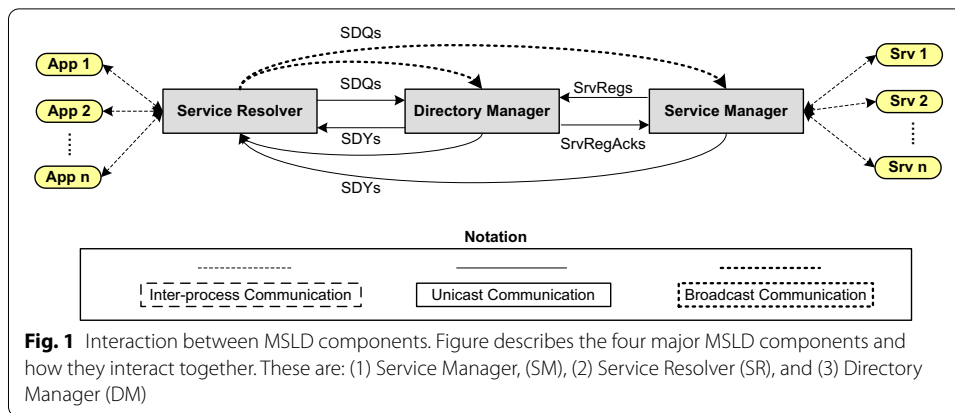
2.1 Integration with stateful auto-configuration protocols

MSLD was designed in such away that it reuses some of the existing stateful auto-configuration protocols such as autoconf. Such an approach aims at optimizing the process of service discovery in terms of the number of control messages being exchanged.

Important auto-configuration functions of the autoconf protocol are delivered through the use of elected/selected hosts. These hosts could be one or multiple and are referred to as Address Agents—AAs. Every AA keeps track of various types of information such as: node specification, IP address, ALT, etc. AAs ensure that local autoconf database (db) are disseminated to other AAs.

Since Autoconf protocol is involved in maintaining, initiating and updating AAs, it would have been more efficient to widen the functionality of AAs by having them perform more than one task including autoconf tasks. For instance, a directory based service directory protocol requires service directory and directory managers (DMs). These can be assured by the db and AAs, respectively. In order to accommodate the service directory and DMs under stateful autoconf protocol, the following changes need to be incorporated.

- Service directory with a full list of accessible services along with information about these services such as attributes and service providers. Table 1 illustrates the structure of such a service directory.

Fig. 1 Interaction between MSLD components. Figure describes the four major MSLD components and how they interact together. These are: (1) Service Manager, (SM), (2) Service Resolver (SR), and (3) Directory Manager (DM)

- A module for querying and administrating the service directory database, which is being added to the autoconf protocol to maintain the services-related information residing in the database. We refer to this module as MANET Service Location and Discovery (MSLD).

For evaluation purposes, SD protocol was integrated with a stateful auto-configuration protocol, DHAPM, which is based on AAs' being selected dynamically. DHAPM is chosen because it has a lower communication overhead than the other stateful approaches [5] and is highly robust against node failures.

2.2 MSLD components

MSLD protocol includes three main components: (1) Service Resolver (SR), (1) Service Manager (SM), and (3) Directory Manager (DM). Figure 1 describes the level of interaction between these components.

2.2.1 Service resolver (SR)

The SR is the module responsible for acquiring and retrieving detailed information about available services. Based on the DMs accessibility, the retrieval process could occur in either of the following two modes: Unicast mode (U_C mode), or Broadcast mode (B_C mode). The former is the default mode and is being activated when the information about DMs is available. With this mode, DM responds directly with unicast messages to incoming unicast service discovery requests (SDQs). The latter generates more overhead compared to the (U_C mode) due to its message broadcasting nature and is activated upon the failure to connect to any of the DMs. With this mode, SDQs messages are broadcasted[1] to all neighboring devices and a unicast message response is expected from any node having the requested service. The response comes in the form of service discovery reply (SDY) with the requested service information details. This mode gives the opportunity to DMs to respond to SDQs with their availability as well as any matching records from their directory, and would cause the SR within the service request initiator to revert to the U_C mode.

---

[1] SR uses multicasts if the underlying MANET supports multicasts.

SR was designed to support two types of queries: single-response and multi-response queries [6]. Based on the user (or application) requirement either of these two modes is selected. A single-response query is selected when only a service is needed, not the device offering this service. However, a multi-response query is selected in case the user is interested in knowing about all devices providing a desired service.

The collected information about service details is cached for a certain time called CLT (Cache Life Time). This makes the response to SR queries more efficient in case the information is readily available in the cache. The cache entries that exceeded their CLT will be removed from the cache at the time of inserting new entries. Cache size is decided based on the type of the device and the number of different service queries generated from that device. It is recommended that minimum cache size should be sufficient to hold at least one entry for each different service request generated from the device.

2.2.2 Service manager (SM)

The SM major two tasks are the following: (1) serves as a liaison between the DM and all existing service modules. This is in contrast to having all service modules interact directly with DMs; instead, every service module needs to register with the local SM by providing their attributes and configuration parameters, which optimizes the number of exchanged messages. (2) handles incoming SDQ messages broadcasted by neighboring devices.

2.2.3 Directory manager (DM)

The MSLD DM is responsible for the following two tasks: (1) collecting then maintaining all available services in MANET, which constitutes the service directory; (2) resolving all SDYs being originated by local or distant SR running on the same or different hosts, respectively.

2.3 MSLD operations

MSLD service directory management process includes the following steps: services naming conventions, services registration, service lookup, and services deregistration. These are described next.

2.3.1 Service naming conventions

MSLD uses compact service Uniform Resource Locator (URL) adjusted for MANETs to encode services. We propose using similar semantics and syntax to the encoding scheme as that outlined in RFC 2609 [34].

2.3.2 Service registration

The service registration process consists of having the service agent on a particular device register itself with the local SM, which coordinates the registration with a neighboring DM.

2.3.3 Service lookup

The service lookup process starts when a user application issues a service discovery request message (SDQ). The SR checks its local MSLD cache for a matching entry. If a valid match was found, then the matching entry is sent to the application along with a flag showing its source (cache or DM). The application may request more up-to-date information by commanding the SR to ignore the cache and query a DM. When querying the DM, the SR unicasts an SDQ message to its DM and initiates a Timer. Upon the reception of as SDY from the DM, the SR processes the message then forward the response to the requesting application. If the timer expires before a SDY messages is received, the SR repeats the same process until the number of attempts reaches a certain threshold $r_{\min}$. If no valid response is received, the SR switches to the broadcast mode (B_C). Under this mode, the SR broadcasts the old SDQ eliciting for service responses. If no positive response is received, the SR signals the termination of the lookup process by returning an error message to the requesting application. The SR reverts back to the unicast mode (U_C) as soon as a reply is received from a DM.

2.3.4 Service deregistration

This phase is executed whenever a node leaves the network. An address release message (*AddrRelReq*) is sent out to its corresponding AA, which marks the entire record of the leaving node as invalid. The same update happens when the ALT time of services expires, and records need to be automatically deleted.

3 MSLD protocol complexity

In this section, the communication overhead of different alternative service discovery approaches is analytically compared with MSLD protocol. For the sake of fairness and clarity, we presume all approaches using the same routing protocol and directory based auto-configuration protocol. This means the overhead arising from routing and auto-configuration is the same for all compared approaches.

3.1 Directory-based complexity in terms of message overhead

To assess the performance of the directory based approach, we assume that all network devices register their available services by adding an entry in the directory. Overhead messages include announcement messages which are periodically sent out by the directory, SDQs issued by client devices (as unicast messages), and the SDY responses from directory to clients' requests. In a given period T_{ad}, the message overhead is given by:

$$O_{B_c} + 2 * N_\chi * O_{U_c} + 2 * N_{\mathrm{reg}} * O_{U_c} + 2 * N_{\mathrm{cng}} * O_{U_c}, \tag{1}$$

where N_χ: SDQs average count, N_{reg}: Service registration average count, N_{cng}: Service updates average count during every announcement period, O_{B_c}: Broadcast message overhead, O_{U_c}: Unicast message overhead.

3.1.1 Directory-less complexity in terms of message overhead

The message overhead of the directory-less approach is different for both modes (push/pull based). For the push-based mode, where services are announced periodically by service agents, and for a given period T_{ad}, the message overhead is given by:

$$N_{\mathrm{sp}} * O_{B_{\mathrm{c}}}, \tag{2}$$

where N_{sp} is the service providers average count. For the pull-based mode, where SDQ messages are flooded by clients, then Unicast SDY replies are sent out by service agents, and for a given period T_{ad}, the message overhead is given by:

$$N_{\chi}(O_{B_{\mathrm{c}}} + \alpha * O_{U_{\mathrm{c}}}), \tag{3}$$

where α is the average number of service agents.

In the case of combined pull/push approaches, clients cache service announcements made by service agents. The pull-based approach is invoked whenever the desired service does not exist in the cache. With this combined approach, we gain in terms of time required to discover services, however more overhead is incurred. It is important to mention that traditional combined pull/push based approaches experience more overhead compared to pull-based ones [31].

3.1.2 MSLD complexity in terms of overhead messages

The overhead messages in MSLD include those messages needed to maintain and synchronize the distributed directory with the goal of achieving resiliency in dealing with node failure. Considering the fact that MSLD is integrated with DHAPM, the directory maintenance is performed alongside with the address agents (AAs) synchronization process. With this scheme, the AA synchronization messages are augmented with extra *service* data. The same scheme is used for the first service registration which is performed as part of the address request step. Subsequent service registration/updates are triggered independently. For a given period T_{ad}, the message overhead of MSLD is given by:

$$\zeta + 2 * N_{\chi} * O_{U_{\mathrm{c}}} + 2 * N_{\mathrm{reg}}^{-} * O_{U_{\mathrm{c}}} + 2 * N_{\mathrm{cng}} * O_{U_{\mathrm{c}}}, \tag{4}$$

where ζ: generated overhead for directory synchronization, N_{reg}^{-}: average late service registration counts during (T_{ad}).

Analyzing both Eqs. 1 and 4 in the case where $N_{\mathrm{reg}}^{-} \leq N_{\mathrm{reg}}$ and $\zeta \ll O_{B_{\mathrm{c}}}$, it is clear that the MSLD achieves better performance in terms of overhead messages compared to that of directory based.

Also comparing Eqs. 3 and 2 with Eq. 4, we conclude that MSLD outperforms the directory-less approach in both modes (pull/push based) in terms of the overhead messages.

4 Experimental results

In this section, we describe the simulation setup, the set of performance metrics being considered to evaluate MSLD, and the performance results under different case scenarios.

4.1 Simulation setup

The performance of MSLD was evaluated using the NS-3 simulation platform under various network configuration conditions. The communication stack consists of DSDV and IEEE 802.11 standards as the main protocols running at the networking layer and at the

MAC layer, respectively. The size of the network consists of 50 nodes randomly distributed with 250 m communication range. The mobility model used is the random way point model with 0 pause time, 2 m/s as a minimum node speed and 4 m/s as a maximum speed. During all experiments, the number of service discovery requests is the same for all nodes and the requests are uniformly distributed over a period of 150 s. Also, we set SR SDQ timeout to 1 s, r_{min} to 2 trials, and the Address Lease Time (ALT) to 100 s.

The graphs and performance results in Sect. were reported by depicting the mean values collected from 1000 trial runs of corresponding appropriately designed experimental scenarios.

4.2 Performance metrics description

The performance of the MSLD protocol is evaluated while considering the following three metrics:

1 *Service discovery latency or time* This metric tracks the time elapsed between when the service request was originated and when a successful service binding response was received.
2 *Service availability ratio* This metric tracks the ratio of positive service bindings (positive binding resolution) to the request messages (to contact the servers) being originated.
3 *Message overhead* This metric counts the total number of control messages during the service discovery process.

We used the aforementioned metrics under different case scenarios to evaluate the performance of MSLD. We also compared the MSLD performance to that of the following three protocols under the same conditions:

- *Dir-based* A distributed directory based protocol is simulated because it has a higher SA than the single directory based approaches. Also, the directory based protocols are known to have a low communication overhead for service discovery when the directory maintenance overhead is ignored. During our simulation we assume that all directories maintain service information and synchronize among themselves, therefore nodes can query any one of the directories for service discovery.
- *Dir-less* This protocol was considered due to the fact that it possesses a large number of SAs when SDQ messages are sent across the entire network.
- *PDP* This protocol showed its effectiveness when compared to many existing IETF protocols [6]. Therefore we considered this protocol for comparison purposes. However, PDP has been configured to use retries in the implementation being considered in this study. This has been done mainly to enable a fair comparison since all other protocols considered in this study use retries which improve service availability.

4.3 Results and discussion

Results were collected while considering different scenarios (static versus dynamic) and under various network configuration parameters.

Table 2 Hops versus latency

Hops	μ_{sdy} (σ_{sdy})	μ_{bind} (σ_{bind})	μ_{total} (σ_{total})
(a) Directory-based: Unicast SDQ and SDY			
1	3.26 (0.25)	3.43 (0.22)	6.69 (0.34)
2	6.73 (0.37)	6.87 (0.35)	13.61 (0.52)
3	10.21 (0.44)	10.43 (0.45)	20.64 (0.63)
4	13.73 (0.54)	13.95 (0.52)	27.68 (0.73)
5	17.27 (0.58)	17.44 (0.60)	34.71 (0.83)
6	20.74 (0.63)	20.96 (0.62)	41.70 (0.87)
(b) Directory-less: Broadcast SDQ and Unicast SDY			
1	8.24 (3.50)	4.54 (2.25)	12.78 (4.31)
2	14.97 (4.22)	7.73 (0.79)	22.71 (4.10)
3	22.74 (5.14)	11.40 (1.30)	34.13 (5.33)
4	30.34 (5.93)	14.33 (1.82)	44.67 (6.41)
5	37.72 (6.34)	17.49 (0.63)	55.22 (6.42)
6	45.38 (7.25)	22.24 (2.69)	67.62 (8.30)
(c) PDP: Broadcast SDQ and SDY			
1	12.04 (4.29)	4.20 (1.92)	16.24 (4.85)
2	23.57 (5.78)	7.17 (0.96)	30.73 (5.77)
3	35.35 (7.22)	11.29 (1.61)	46.64 (7.52)
4	46.97 (7.98)	14.43 (1.74)	61.40 (8.33)
5	58.93 (9.09)	17.96 (1.64)	76.90 (9.49)
6	69.94 (10.23)	22.70 (3.39)	92.64 (10.78)

4.3.1 Comparison in static scenarios

We considered the following three ad hoc network static case scenarios where we assume that there are no message losses and there is a full service availability: (1) scenario1: both SDY and SDQ messages are being unicasted, (2) scenario2: SDY messages are unicasted and SDQ are broadcasted, (3) scenario3: both SDY and SDQ messages are being broadcasted. In all experiments, we considered a maximum of six hops between the server node and the DM.

In every hop, we calculate the standard deviation (σ), and the mean (μ) of each of these service times: resolution time (μ), service provider contact time (T_{bind}), and total time (T_{total}). The results presented in Table 2 show that the latency increases as the number of hops increase. However, the case scenario PDP suffers from the highest latency and unicast forwarding outperformed the broadcast case. Also, we noticed that despite the six hops distance between the server and the DM, we achieved less than 95 ms latency.

In the idealistic case scenario execution, and for each of the above cases, the control overhead messages is $(3 * O_{U_c} + O_{B_c}) * N_\chi$ for the Dir-less case, $4 * N_\chi O_{U_c}$ for the Dir-based case, and $2 * N_\chi * (O_{U_c} + O_{B_c})$ for the PDP case.

From these observations, since the Dir-based showed better latency and overhead in static cases, we conclude that MSLD should operate in U_C mode in case of the existence of DM.

Table 3 Comparison of SD Protocol Latency (s)

Approach	μ_{sdq} (ci_{Tsdq})	μ_{bind} (ci_{bind})	μ_{total} (ci_{total})
Dir-based	0.122 (0.016)	0.051 (0.008)	0.166 (0.018)
Dir-less	0.203 (0.019)	0.080 (0.008)	0.258 (0.021)
PDP	0.060 (0.009)	0.063 (0.009)	0.123 (0.013)
MSLD	0.138 (0.017)	0.057 (0.008)	0.187 (0.020)

4.3.2 Comparison in dynamic scenarios

During this set of experiments, we simulated a 50 node MANET network with four different coverage areas: (1) 0.250×0.250 km^2, (2) 0.5×0.5 km^2, (3) 0.75×0.75 km^2, and (4) 1×1 km^2. Several sets of experiments were executed for each of these topologies consisting of varying the rate of generation of SDQ messages (3–27 SDQ messages/second). In this dynamic scenario, and because of the node mobility, we assumed a Random loss of messages. Also for simplicity, we made the following two assumptions: (1) every server in the network offers a unique service, and (2) SDQ messages are of type single response requests.

(i) *Latency*: Table 3 captures the mean and its corresponding 95% confidence interval performance of the three metrics (the discovery time, the binding time, and the total time) of the four approaches (Dir-based, Dir-less, PDP, and MSLD). The following observations and analysis can be made:

- The latency performance behavior for Dir-based, Dir-less, and MSLD is comparable to that of the static case scenario. The Dir-based showed the lowest latency which is attributed to the fact that request messages are directly unicasted to DM. On the other hand, the Dir-less technique showed the highest latency which is attributed to the extra delay induced by the broadcast jitter.
- MSLD performance was between that of Dir-based and Dir-less approaches. This was expected because MSLD relies on both approaches.
- Because of the broadcast of SDQ and SDY, It was expected that PDP achieves the highest latency, however lower latency was witnessed. This is attributed to the fact that the cache hit ratio is higher due to SDY messages being cached and acting as announcements for available services, and the computed latency does not account for the extra wait time (PDP$_{wait}$) that the responder node waits while expecting a different node with better energy specifications to respond. Based on [6], the wait time can be configured and usually varies between 0.5–3 s. In the event of accounting for PDP$_{wait}$, the latency is around 0.6 s which is way higher than that achieved in the other approaches.
- For all approaches, the difference in the latency is considered small (in the order of milliseconds) since a delay between 1 and 2 s is acceptable by the user in this context (service initialization). Therefore, a different metric needs to be considered in order to differentiate between these protocols.

(ii) *Service Availability (SrvA)*: Table 4 captures the performance of all four Service Discovery approaches in terms of service discovery failure (F_{res}), service binding

Table 4 Service availability

Approach	F_{res}	F_{binds}	SA
Dir-based	34	215	0.9667
Dir-less	90	131	0.9705
PDP	0	243	0.9675
MSLD	10	211	0.9704

Table 5 Message overhead

Approach	U_C	B_C	M_{total}	$\overline{M}_{total}$
Dir-based	63,703	0	63,703	8.493[a]
Dir-less	451,213	8524	459,737	61.29
PDP	380,526	7556	388,082	51.74
MSLD	66,491	64	66,556	8.874

[a] 11.21, when the directory maintenance cost included

failure (F_{binds}), and the total SrvA when a total number of 7500 SDQ messages are being originated. The following observations and analysis can be made:

- The Dir-less approach witnessed a higher failure probability of service requests compared to that of the Dir-based technique. This is attributed to the fact that with the former approach the failure/success depends on the accessibility or availability of the server offering such services. However with the latter case, the failure/success of service requests mainly depends on the availability of DMs. On the other hand, the MSLD protocol witnessed fewer service request failures when compared to Dir-less and Dir-based approaches. This is attributed to the fact that MSLD borrows from both protocols and hence a failure occurs only in case of the non-availability of DM and non-accessibility of servers offering the requested services. It is also important to note that the PDP witnessed a 100% success rate which is attributed to the fact that nodes local caches are being advertised along the SDY message broadcasts making PDP look like a distributed Dir-based protocol where all nodes are playing the role of directories leading to low F_{res}.
- The binding failure performance was better in the case of Dir-less protocol and was comparable to the rest of the protocols (MSLD, Dir-based, and PDP). This is attributed to the fact that in the case of the Dir-less approach service queries are resolved by the server, making it less likely for the same server to be inaccessible, and hence there are fewer service binding failures.
- The overall results about SrvA showed that MSLD protocol outperformed the PDP and Dir-based approaches. However, similar performance was achieved when compared to Dir-less.

(iii) *Overhead*: We collected measurements for the following four metrics: (1) amount of unicast messages (U_C), (2) Broadcast messages being sent (B_C), (3), total number of control message being exchanged during the entire experiment (M_{total}), and (4)

average SDQ overhead messages ($\overline{M}_{\text{total}}$) given by ($M_{\text{total}} / \sum \text{SDQs}$). These results are captured in Table 5.

As expected, results are showing poor overhead of PDP and Dir-less protocols which is due to the broadcast nature of these protocols. In case of Dir-less protocol, this performance could be improved by limiting the maximum number of hops that broadcast messages could travel. However, this is expected to reduce the availability. Also, results are showing that PDP slightly outperforms the Dir-less approach because this protocol resolves SDQs by using the existing information in the cache, which minimizes the need for broadcasting the control messages. On the other hand, MSLD message overhead was improved with a rate between 14 and 17% compared to that of PDP and Dir-less, respectively.

Table 5 is also showing that MSLD and Dir-based protocols are experiencing almost similar performance in terms of overhead messages.

To conclude, MSLD outperformed the other approaches in terms of communication overhead while maintaining a similar SrvA as the Dir-less approach. In practical cases where SrvA is of major importance, MSLD variant could be considered (MSLD-hsa). It is expected that MSLD-hsa achieves higher SrvA with considerably less overhead.

4.3.3 Performance evaluation under different network setup

In this section, we study the impact of varying some of the network setup parameters on the performance of the SD protocols under study. These parameters are the following: Network area, SDQ rates, network servers' redundancy, and Cache Life Time. We also simulated MSLD-hsa to study the tradeoff between message overhead and SA.
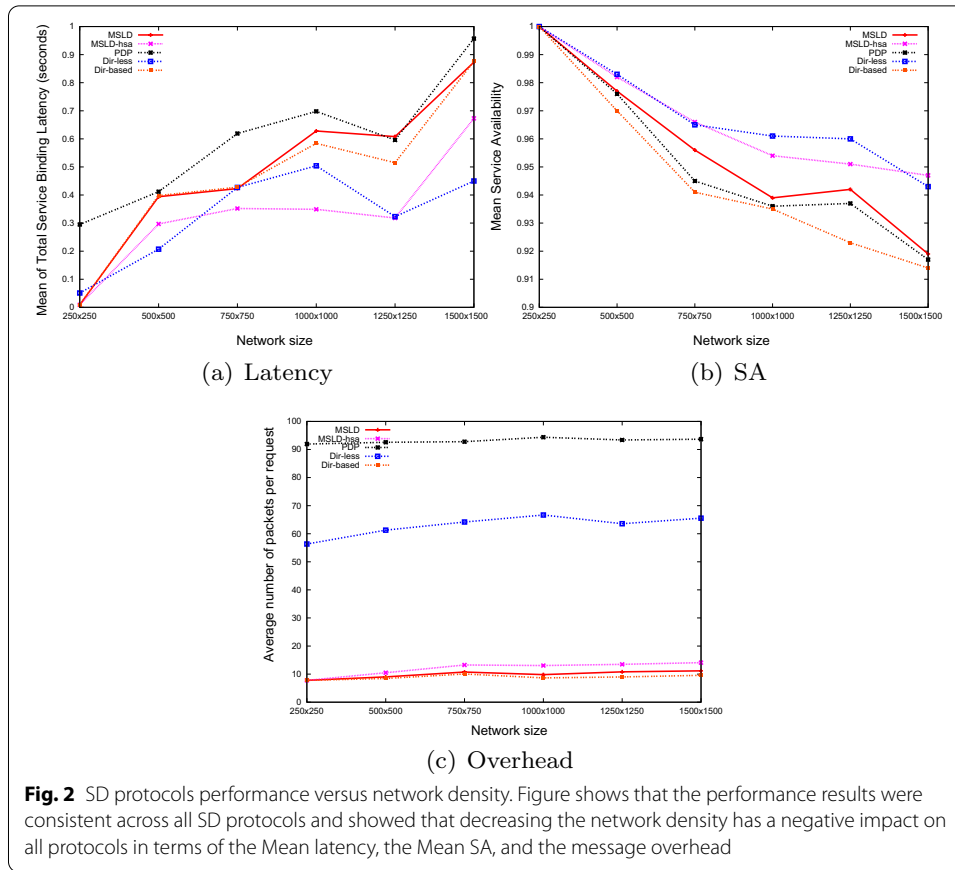
4.3.4 (i) Impact of network area

In this experiment, we propose varying the MANET service coverage range from 0.25 to 1.5 km^2 while keeping the number of nodes the same. In other words, we propose varying the node density within the MANET network. Performance results are captured in Fig. 2.

Performance results were consistent across all SD protocols and showed that decreasing the network density has a negative impact on all protocols in terms of the Mean latency, the Mean SA, and the message overhead. Further, MSLD and its variant (MSLD-hsa) achieved lower overhead and comparable SA performance compared to other protocols.

4.3.5 (ii) Impact of SDQ rates

The goal of this experiment is to study the effect of increasing the SDQ message rates on the different SD protocols under study. We propose varying the SDQ rate from 0.8 to 27 SDQ/s. We collected performance results for different network areas (ranging from 0.25 to 1 km^2). Results are captured in Fig. 3.

As the number of SDQ messages in the network increases, it is expected to see no impact (in terms of service availability as well as $\overline{M}_{\text{total}}$) on the Dir-based and MSLD protocol running in Unicast mode. In the case of Dir-less protocol, the behavior is the same as the previous two cases, whenever the SDQ rate is low. In the case of high SDQ rates, more latency is experienced because the protocol relies on excessive broadcast

Fig. 2 SD protocols performance versus network density. Figure shows that the performance results were consistent across all SD protocols and showed that decreasing the network density has a negative impact on all protocols in terms of the Mean latency, the Mean SA, and the message overhead

messages. In addition, for the Dir-less protocol more retries is needed due to the high F_{res} (see Table 4) contributing to the number of broadcast overhead messages leading to low SA (compared to other protocols) and high latency.

On the other hand, results show that PDP performance in terms of latency and overhead improves as the SDQ rates increase. This is due to the fact that the probability of the same requested service to re-occur becomes higher as the number of SDQ messages increase (Fig. 3). We are also noticing, as expected, that the SA is the same for different SDQ rates. This is due to the fact that SA is dependent on the service binding which does not depend on how SDQs get resolved.

In conclusion, and for this set of experiments, the performance of MSLD-hsa depends on the rate of SDQ messages and achieved lower overhead and the best SA.

4.3.6 (iii) Impact of increasing the network servers redundancy (N_{srv})

Figure 4 shows the message overhead ($\overline{M}_{total}$) of all four protocols under study. These results were collected when increasing the percentage of servers offering the same service from 5 to 50 for low and high SDQ rates (these are captured in Fig. 4a, b, respectively.

For low rates of SDQ messages, a variation of N_{srv} does not seem to have a major impact on the MSLD protocol running in a unicast mode and the Dir-based protocol. In both protocols, the number of control messages remains the same because only DM
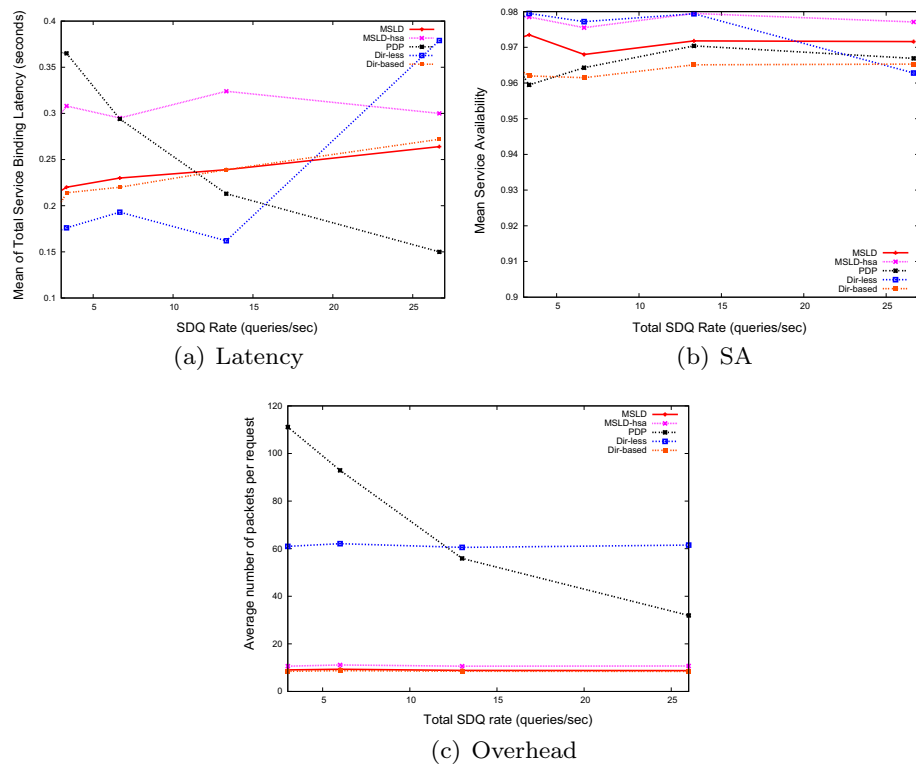
Fig. 3 SD protocols performance versus SDQ rates. Figure shows that As the number of SDQ messages in the network increases, no impact (in terms of service availability) was noticed on the Dir-based and MSLD protocol running in Unicast mode. In case of high SDQ rates, more latency is experienced. Also, results show that PDP performance in terms of latency and overhead improves as the SDQ rates increase
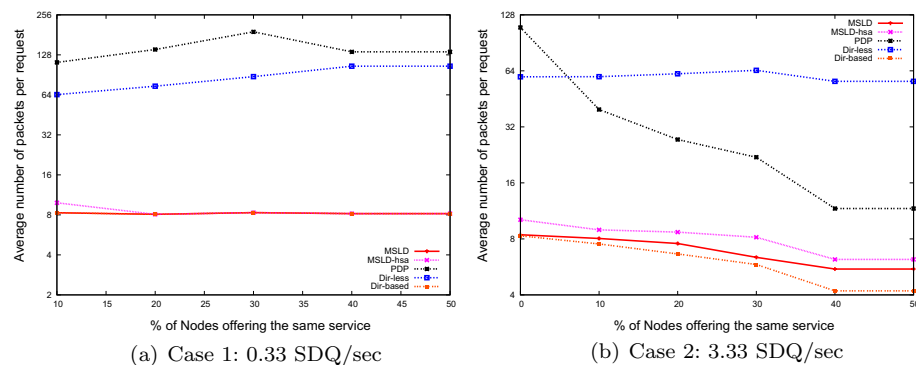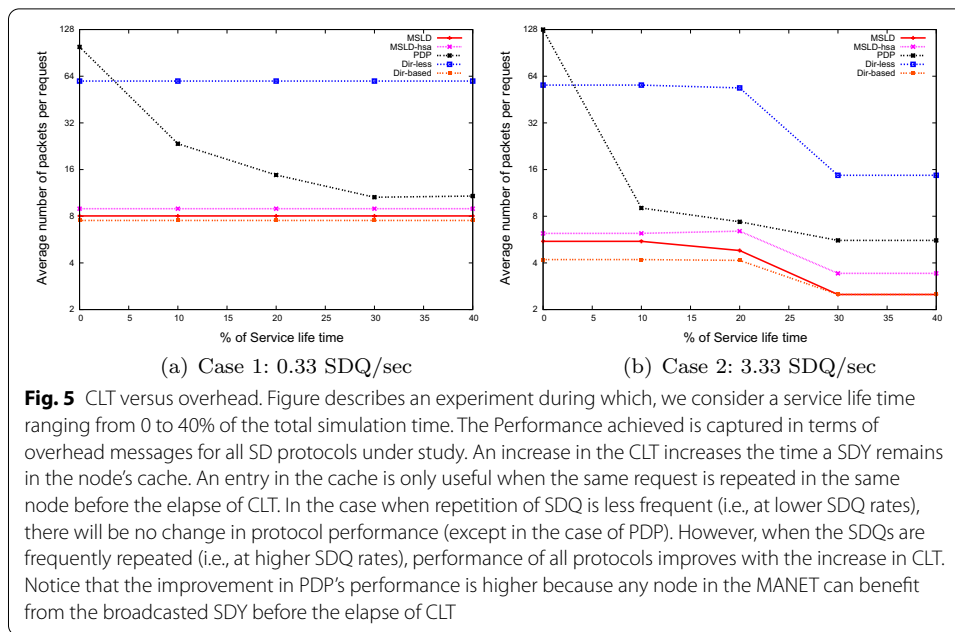

Fig. 4 Services reliability versus Overhead. Figure shows the message overhead of all four protocols under study. These results were collected when increasing the percentage of servers offering the same service from 5 to 50 for low and high SDQ rates. For low rates of SDQ messages, a variation of N_{srv} does not seem to have a major impact on the MSLD protocol running in a unicast mode and the Dir-based protocol. On the other hand, the N_{srv} seems to have an impact on the number of control messages since all servers having the requested service are expected to respond with an SDY messages for every service request being originated leading to more overhead as N_{srv} increases. Same behavior is witnessed for PDP except the case where the cache has information about the requested service and only devices with higher energy may respond on behalf of low-energy nodes. Conversely, at higher SDQ rates, when the number of servers offering a similar service increases, there will be a higher chance of re-requesting the same service, and hence message overhead decreases because of higher cache hits. PDP has maximum performance gain because of SDY broadcasts, which well-populate node cache

(a) Case 1: 0.33 SDQ/sec   (b) Case 2: 3.33 SDQ/sec

Fig. 5 CLT versus overhead. Figure describes an experiment during which, we consider a service life time ranging from 0 to 40% of the total simulation time. The Performance achieved is captured in terms of overhead messages for all SD protocols under study. An increase in the CLT increases the time a SDY remains in the node's cache. An entry in the cache is only useful when the same request is repeated in the same node before the elapse of CLT. In the case when repetition of SDQ is less frequent (i.e., at lower SDQ rates), there will be no change in protocol performance (except in the case of PDP). However, when the SDQs are frequently repeated (i.e., at higher SDQ rates), performance of all protocols improves with the increase in CLT. Notice that the improvement in PDP's performance is higher because any node in the MANET can benefit from the broadcasted SDY before the elapse of CLT

nodes reply to SDYs. On the other hand, the N_{srv} seems to have an impact on the number of control messages since all servers having the requested service will respond with an SDY message for every service request being originated leading to more overhead as N_{srv} increases. The same behavior is witnessed for PDP except for the case where the cache has information about the requested service and only devices with higher energy will respond on behalf of low-energy nodes.

Conversely, at higher SDQ rates, when the number of servers offering a similar service increases, there will be a higher chance of re-requesting the same service, and hence message overhead decreases because of higher cache hits. Thus, as shown in Fig. 4(b), the performance of all protocols improves with N_{srv}. PDP has maximum performance gain because of SDY broadcasts, which well-populate node cache.

4.3.7 (iv) Impact of cache life time (CLT)

The CLT corresponds to the lifetime of services and is set by the server. During this experiment, we consider a service lifetime ranging from 0 to 40% of the total simulation time. Fig. 5 captures the performance achieved in terms of overhead messages for all SD protocols under study.

An increase in the CLT increases the time an SDY remains in the node's cache. An entry in the cache is only useful when the same request is repeated in the same node before the elapse of CLT. In the case when repetition of SDQ is less frequent (i.e., at lower SDQ rates), there will be no change in protocol performance (except in the case of PDP) as shown in Fig. 5a. However, when the SDQs are frequently repeated (i.e., at higher SDQ rates), the performance of all protocols improves with the increase in CLT. Notice that the improvement in PDP's performance is higher because any node in the MANET can benefit from the broadcasted SDY before the elapse of CLT, as explained (ii).

5 Related work

Service discovery (SD) approaches can be classified into either directory based (centralized) or directory-less (distributed) [4].

The former approach consists of having a single or group of hosts maintaining the directory of all existing services in the entire network. Under this setup, servers or service agents need to update the centralized directory with the corresponding service information details, which will eventually be queried by client devices. On the other side, the directory-less based approach avoids having a centralized management directory through the distribution of service messages across all network nodes. Depending on the distributed message type, the service discovery can be done in two modes: (i) push mode: servers broadcast (or multicast) frequent service advertisements to all hosts, and clients passively cache the services and select the service which is of interest; (ii) pull mode: clients broadcast service requests, and servers that offer the requested service reply to the request.

Service Discovery (SD) solutions have also been conceived for different domains or environments such that traditional infrastructure based networks, infrastructure-less based networks (aka MANET), and Internet of Things (IoT) platforms. It is important here to mention that "Traditional infrastructure based network" includes networks with physical network connectivity allowing devices to efficiently communicate and exchange information. In the next subsections, we overview existing solutions in each of these platforms.

5.1 SD in infrastructure based network context

Many Service Discovery (SD) protocols were proposed for traditional networks. For instance, the Service Location Protocol (SLP) [35], is among the first protocols that was introduced and designed by the Internet Engineering Task Force (IETF) working group. The SLP protocol is built around three main agents: (1) Service agents, (2) Directory agents (DAs), and (3) User agents (UAs). Service agents are responsible for advertising information about available services such as types, attributes, access etc. The DAs are responsible for storing service information already announced for users. The UAs act on behalf of end-user applications to query DAs about available services. Under this scheme, the SLP protocol can operate in two modes: centralized (if DAs are present), or pull-based (if DAs are absent). Jini, Salutation and UDDI [26] are other standard approaches that use centralized salutation managers, lookup servers, and registry respectively for retrieving, publishing, and managing services information.

Contrarily to SLP, the Simple Service Discovery Protocol (SSDP) [26] was proposed by Microsoft [30] as part of the Universal Plug and Play initiative. SSDP falls under the category of directory-less protocols and operates in both modes: (1) pull and (2) push modes.

Though these protocols are very efficient in traditional networks, they cannot be considered as efficient in resources constrained environments such as MANETs for the following reasons: (1) they fail to account for packet loss and transmission latency, (2) they do not take into consideration various inherent MANET constraints. These refer to the "nature" and "characteristics" of MANET networks, where no centralized control

exist, no association between communicating nodes, relatively fast mobility, limited bandwidth, shared broadcast communication channel, limited availability of resources, and heterogeneous types of communicating nodes. In fact, there has been a consensus among researchers that the traditional SD protocols are not a good option for MANETs [27].

5.2 SD in MANET context

Several research proposals have tried to address the SD problem in MANETs from different perspectives.

DEAPspace [8] falls under the category of push based and decentralized protocols. DEAPspace operates by periodically synchronizing among neighboring devices a database list of all known services called *world view*. Compared to other existing push model based approaches, DEAPSpace is known to be a fast protocol in terms of discovering services while not over-consuming available network bandwidth.

Konark [9] is another protocol that was designed specifically for service discovery and delivery in devices operating in ad hoc networks. This protocol alternates between pull and push modes then uses multicast messages for service discovery and advertisement.

Bonjour (known as Rendezvous) is another type of SD protocol that uses a DNS based protocol for service discovery (DNS-SD [32]). The service discovery requires sending two queries: one to acquire the list of available services, and a second query to acquire all information pertaining to a particular service.

PDP [6] is yet a different SD protocol for MANETs. This protocol uses features of push and pull based approaches. Under this scheme, devices keep track of an exhaustive list of all advertised services, which are used to respond to service queries.

Unfortunately, all these directory-less based approaches do not scale well for larger networks due to the excess of control messages being sent out during multicasting or broadcasting [33].

Several techniques were considered to cope with the excess of control overhead messages. Authors in [10, 11] have discussed the concept of using DSR and TORA routing protocols for network service discovery. While others considered Proactive and hybrid routing protocols have also been extended to support SD such as OLSR and ZRP [12–14].

In [7], the authors proposed location aware service discovery protocol and tried to achieve efficiency by restricting service discovery broadcasts to limited area.

Using the distributed service directories is another technique that improves the number of control messages being exchanged [27].

5.3 SD in IoT context

Recently, SD in IoT started to emerge. Most of the proposed solutions attempt to address the following challenges: availability, scalability, robustness, security, interoperability, management, and self-configuration. Proposed solutions for SD in IoT have been surveyed and were classified into nine different categories [22]. Unfortunately most of the proposed approaches fail to effectively address the aforementioned requirements which are imposed by IoT environment. In [21] for instance, the authors introduced the concept of "IoT Gateway", which is a special smart node with caching capability and is part of the peer-to-peer

network used to store and retrieve information about resources. The IoT Gateway interacts with IoT nodes at the application level.

Authors in [23] proposed an SD solution for the IoT environment which is based on mDNS/DNS-SD, a widely used protocol with good scalability. They introduced the concept of "Service Context" to cope with the protocol scalability of large networks.

In [24], a new framework for SD in IoT was proposed which is based on three layers: (1) Proxy layer aiming at discovering physical things regardless of the protocols and communication technologies being used, (2) Discovery Layer, which consists of a database where resource parameters can be stored.

Authors in [17] have studied and presented an extended survey of existing SD mechanisms being deployed in the IoT context along with their weaknesses. They grouped these techniques into 4 categories namely: (1) context-aware techniques, (2) Energy-aware techniques, (3) QoS-aware techniques, and (4) semantic-aware techniques.

Along the same line, authors in [18] presented a review of SD techniques in IoT environment. The authors have presented extended classification of such techniques compared to [17]. This includes: (1) Content-based, (2) Context-based, (3) Location-based, (4) User-based, (5) Semantic-based, (6) Resource-based. They also summarized the main challenges, which are inherent to the nature of large scale mobile networks that categorize IoT environment. These challenges are the following: Asynchronous notifications, Device replacement, standardization, latency, flexibility and support to communication protocols, service selection optimization, and energy constraints.

In [19], the authors have proposed a different classification of existing SD in IoT context techniques. The proposed classification is based on their architecture, search technique, their dependency on Internet protocols, and their operation mode.

Authors in [20] have focused on service search in IoT environment. Contrary to existing service discovery solutions which rely on trivial service information revealed by service providers, the authors have proposed a flexible service search method that uses an extended set of information and data generated by IoT devices to support service retrieval. Such approach was demonstrated to overcome the service availability problem.

Unfortunately, current service discovery protocols proposed for infrastructure based communication networks, MANET, and IoT networks failed to address all inherent MANET constraints and challenges in terms of the non-existence of centralized control, non-existence of association between communicating nodes, relatively fast mobility, limited bandwidth, shared broadcast communication channel, limited availability of resources, and heterogeneous types of communicating nodes. MSLD, which we have proposed in prior work [1], attempts to overcome most of these challenges. It is a robust general purpose protocol in the sense that it works in all environment. MSLD is integrated with stateful autoconf protocol with the goal of using its address agents (AAs) functionalities. Consequently, it is guaranteed that the selection and maintenance of directories occur with no additional overhead, while taking advantage of the distributed directory-based approaches in terms of protocol robustness.

6 Conclusion

This paper proposes an extensive performance evaluation and comparative study of a service discovery protocol for MANETs (MSLD) that we proposed in our prior work. MSLD was introduced with several key features such as limited protocol control messages overhead, high availability, and generality. MSLD was evaluated based on the following three metrics: service discovery latency, service availability features, and control message overhead. The same metrics were also considered to compare the performance of MSLD with Dir-Based, Dir-less, and PDP protocols under various network simulation setups. In a dynamic scenario, while all protocols achieved a comparable low latency, MSLD outperformed all in terms of communication overhead, which is crucial in a resource constrained environment such as MANET. Similar performance results were also achieved under various network configuration setup in terms of varying the network area, service discovery request, network servers discovery, and the services cache lifetime.

6.1 Future work

In our future work, we propose exploring the following directions: we intend to develop an integrated architecture based on MSLD that provides dynamic auto-configuration, name resolution and service discovery for tactical cloud environments [2]. Furthermore, we intend to add a lightweight security module with the goal of providing defensive mechanisms for MSLD when operating in hostile environments. We intend to further reduce the protocol overhead by using space efficient probabilistic data structures such as bloom filters [36].

Abbreviations
AA: Address agent; ALT: Address lease time; BC: Broadcast; CLT: Cache life time; DA: Directory agent; DHAPM: Dynamic host auto-configuration protocol for MANETs; DM: Directory manager; DNS: Domain name system; DNS-SD: DNS service discovery; DSR: Dynamic source routing; LTE: Long-term evolution; IETF: Internet engineering task force; IoT: Internet of things; MANET: Mobile ad hoc networks; MSLD: MANET service location discovery; OLSR: Optimized link state routing; PDP: Pervasive discovery protocol; RFC: Request for comments; SD: Service discovery; SDQ: Service discovery request; SDY: Service Discovery Reply; SLP: Service location protocol; SM: Service manager; SR: Service resolver; SSDP: Simple service discovery protocol; TORA: Temporally ordered routing algorithm; UA: User agent; UC: Unicast; UDDI: Universal description discovery and integration; UWB: Ultra wideband; ZRP: Zone routing protocol.

Authors' contributions
Authors NM,GP, and BS designed the protocol and the simulation experiments. NM, GBB, and WEH contributed in performance results discussions and writing the first draft of the paper. All authors read and approved the final manuscript.

Data availability
Data sharing not applicable to this article as no datasets were generated or analyzed during the current study

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, Al Khobar, Kingdom of Saudi Arabia. [2]Cybersecurity Center, Prince Mohammad Bin Fahd University, Al Khobar, Kingdom of Saudi Arabia. [3]Computer Science Department, American University of Beirut, Beirut, Lebanon. [4]School of Computing Sciences, University of East Anglia, Norwich, UK. [5]School of Computer and Information Engineering, University of Ulster, Coleraine, UK.

References

1. M. Nazeeruddin, G. Parr, B. Scotney, An efficient and robust service discovery protocol for dynamic MANETs. In *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, pp. 49-60 (Springer, Berlin, 2006)
2. P. Teller, M. McGarry, D. Shires, S. Park, L. Nguyen, J. Deroba, Enabling battlefield decision-making in the tactical cloud. Army High Performance Computing Research Center (AHPCRC), research project (2014)
3. H. Wu, K. Wolter, A. Grazioli, Cloudlet-based mobile offloading systems: a performance analysis. IFIP WG **7**, 24–26 (2013)
4. C.N. Ververidis, G.C. Polyzos, Service discovery for mobile ad hoc networks: a survey of issues and techniques. IEEE Commun. Surv. Tutor. **10**(3), 30–45 (2008)
5. M. Nazeeruddin, G. Parr, B. Scotney, DHAPM: a new host auto-configuration protocol for highly dynamic MANETs. J. Netw. Syst. Manag. **14**(3), 441–475 (2006)
6. C. Campo, C. García-Rubio, A. Marín López, F. Almenárez, PDP: a lightweight discovery protocol for local-scope interactions in wireless ad hoc networks. Comput. Netw. 50(17), 3264-3283 (2006)
7. J. Kniess, O. Loques, C.V.N. Albuquerque, Green service discovery protocol in mobile ad hoc networks. In *ENERGY 2013 The Third International Conference on Smart Grids, Green Communications and IT Energy-Aware Technologies*, pp. 143–149 (2013)
8. R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, A. Schade, DEAPspace—transient ad hoc networking of pervasive devices. Comput. Netw. **35**(4), 411–428 (2001)
9. C. Lee, A. Helal, N. Desai, V. Verma, B. Arslan, Konark: a system and protocols for device independent, peer-to-peer discovery and delivery of mobile services. IEEE Trans. Syst. Man Cybern. A Syst. Hum. **33**(6), 682–696 (2003)
10. G. Bhumika, M. Zaveri, H. Rath, Trust based service discovery in mobile ad-hoc networks. Lect. Notes Softw. Eng. **3**(4), 308 (2015)
11. N. Islam, Z.A. Shaikh, Towards a robust and scalable semantic service discovery scheme for mobile ad hoc network. Pak. J. Eng. Appl. Sci. **10**, 68–88 (2012)
12. A. Ergawy, B. Magnusson, Supporting distance vector routing over device discovery flows in the pervasive middleware PalCom. In *Proceedings of the 6th International Conference on Ambient Systems, Networks and Technologies, ANT-2015, London, UK. Procedia Computer Science*, vol. 52, pp. 153–160 (2015)
13. M.I. Vara, C. Celeste, Cross-layer service discovery mechanism for OLSRv2 mobile ad hoc networks. Sens. J. **15**, 17621–17648 (2015)
14. A.G. Walikar, R.C. Biradar, A survey on hybrid routing mechanisms in mobile ad hoc networks. J. Netw. Comput. Appl. **77**, 48–63 (2017)
15. A. Deshpande, MANETs as propellant in the growth of the Internet of Things. J. Comput. Eng. **18**(5), 1–7 (2016)
16. D.G. Reina, S.L. Toral, F. Barrero, N. Bessis, E. Asimakopoulou, *The Role of Ad Hoc Networks in the Internet of Things: A Case Scenario for Smart Environments, Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence*, vol. 460 (Springer, Berlin, 2013), pp. 89–113
17. B. Pourghebleh, V. Hayyolalam, A. Aghaei Anvigh, Service discovery in the Internet of Things: review of current trends and research challenges. Wirel. Netw. **26**, 5371–5391 (2020)
18. K. Khalil, K. Elgazzar, M. Seliem, M. Bayoumi, resource discovery techniques in the internet of things: a review. Internet of Things **12**, 100293–1 (2020)
19. S. Abdellatif, O. Tibermacine, A. Bachir, Service discovery in the internet of things: a survey. In *International Symposium on Modelling and Implementation of Complex Systems*, pp. 60-74 (Springer, Cham, 2018)
20. S. Sim, H. Choi, A study on the service discovery support method in the IoT environments. Int. J. Electr. Eng. Educ. **57**(1), 85–96 (2020)
21. S. Ciran, L. Davoli et al., A scalable and self-configuring architecture for service discovery in the Internet of Things. Internet of Things J. **1**(5), 508–524 (2014)
22. M. Asiez, S. Benharzallah, H. Bennoui, Service discovery for the Internet of Things: comparison study of the approaches. In *Control, Decision and Information Technologies Conference (CoDIT'17), Spain*, April 5–7 (2017)
23. M. Stolikj, R. Verhoeven, P. Cuijpers, J. Lukkien, Proxy support for service discovery using mDNS/DNS-SD in low power networks. In *IEEE 15th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (Australia, Sydney, 2014)
24. S. Datta, R. Costa, C. Bonnet, Resource discovery in Internet of Things: current trends and future standardization aspects. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum*, Milan, Italy, December, pp. 14–15 (2015)
25. M. Zakarya, I. Rahman, A short overview of service discovery protocols for MANETS. VAWKUM Trans. Comput. Sci. **1**, 1–6 (2013)
26. S. Kaushik, R.C. Poonia, S. Kumar Khatri, Comparative study of various protocols of DDS. J. Stat. Manag. Syst. **20**, 647–658 (2017)
27. B. Djamaa, M. Richardson, N. Aouf, Towards efficient distributed service discovery in low-power and lossy networks. Wirel. Netw. **20**(8), 2437–2453 (2014)
28. K. Cho, G. Park, W. Cho, J. Seo, K. Han, Performance analysis of device discovery of bluetooth low energy (BLE) networks. Comput. Commun. **81**, 72–85 (2016)
29. W. Zhang, L. Chen, X. Liu, Q. Lu, P. Zhang, S. Yang, An OSGi-based flexible and adaptive pervasive cloud infrastructure. Sci. China Inf. Sci. **57**(3), 1–11 (2014)
30. Universal Plug-and-Play (UPnP) Forum, Microsoft Corporation. http://www.upnp.org
31. P.E. Engelstad, Y. Zheng, Evaluation of service discovery architectures for mobile ad hoc networks. In *The Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05)* (2005)

Ben Brahim *et al. J Wireless Com Network*   (2022) 2022:6

Page 21 of 21

32. K. Lynn, S. Cheshire, M. Blanchet, D. Migault, Requirements for scalable DNS-based service discovery (DNS-SD)/multicast DNS (mDNS) extensions. RFC 7558 (2015)
33. M. Buvana, M. Suganthi, An efficient cluster based service discovery model for mobile ad hoc network. KSII Trans. Internet Inf. Syst. (TIIS) **9**, 680–699 (2015)
34. E. Guttman, C. Perkins, J. Kempf, Service templates and service: schemes. RFC 2609 (1999)
35. E. Guttman, C. Perkins, J. Veizades, M. Day, Service location protocol. RFC 2608 (1999)
36. K. Warren, S. Weber, *Below Cross-Layer: An Alternative Approach to Service Discovery for MANETs. Ad Hoc Networks* (Springer, Berlin, 2013), pp. 212–225

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.