

# Towards a Robust and Scalable Semantic Service Discovery Scheme for Mobile Ad hoc Network

Noman Islam<sup>1</sup> and Zubair A. Shaikh<sup>1</sup>

1. National University of Computer and Emerging Sciences, Shah Latif Town, Karachi

## Abstract

*Mobile Ad hoc Networks (MANET) are characterized as an infrastructure-less communication networks formed among a set of stranger nodes. Service Discovery is a challenging problem in such networks due to their non-deterministic and improvised nature. This manuscript addresses two main issues related to service discovery in MANET i.e. consistency management and knowledge representation. Consistency management i.e. the problem of maintaining a coherent view of the services in the network is a taxing job due to the sharp variations in the service availability information in MANET. Similarly, the lack of a standard mechanism for representation of data and resources on the network engenders syntactic and semantic interoperability issues during discovery of services. The use of a colossal schema to describe the services can resolve the issue of non-interoperability, but this approach is certainly not viable because of the limited capabilities of nodes. In view of these challenges, this paper presents a robust and scalable network layer semantic service discovery scheme. A network layer service discovery scheme is presented that finds out the requested service jointly with the corresponding route to the provider of service. For ensuring the robustness of the process, a network layer consistency management scheme is also proposed that maintains the valid state of the services by exploiting the vigilance of network layer. To solve the knowledge representation issue, we recommend a scalable multi-tiered approach based on a general purpose ontology called Software Ontology for Ad hoc and Vehicular Network Applications (SLAVE). The multi-tiered approach keeps a portion of global schema at individual nodes of the network and advocates progressive growth of schema information. The proposed scheme has been simulated in JIST/SWANS simulator. The simulation results assert the robustness and scalability of the consistency manager and knowledge representation scheme respectively.*

**Key Words:** Mobile Ad hoc Network, Semantic Service Discovery, Consistency Management, SLAVE, Network Layer Service Discovery

## 1. Introduction

The exponential growth of low-price computing gadgets and the appearance of small range connectivity mediums have paved the way for new paradigms of network computing. *Mobile Ad hoc Network (MANET)* is an emerging model of computing that has gained an immense amount of attention in research and academia recently. They are characterized as those types of communication network that are established among a group of mobile nodes in impromptu fashion. We can model MANET via a computer graph  $G(N,E)$ , where  $N = \{n_1, n_2, n_3, \dots\}$  is a set of mobile nodes in the network and  $E \subseteq N \times N$  is a set of links among the nodes of the MANET. At a particular instant of time  $t$ , any node  $n_i \in N$  travels with a velocity  $\vec{v}_i$  and is situated at a position  $(x,y)$ . Any link  $e_{ij} \in E$  has a bandwidth  $b_{ij} = [0,B]$ . In addition, the links in the network are not always bidirectional i.e.  $e_{ij}$  doesn't entail  $e_{ji}$ .

The spontaneous nature of MANET leads to a multitude of new research problems as mentioned in [1-3, 25]. One major research problem that arises in MANET due to its dynamic disposition is the discovery of appropriate services in the vicinity of a node. A *service* can be defined as "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description"[37]. A service can be qualified by its description  $D$ , the provider details  $P$  and QoS attributes  $Q$ . Examples of services are printing service, an executable program or gateway service etc. The *service discovery* can be defined as the process of locating a node offering a service i.e. *service provider* in the vicinity of supplicant node or *service consumer*, such that the requested service matches with the offered service. The *matching process* of service discovery scheme refers to comparing the description as well as the QoS attributes of the requested and offered services. The services are usually described by identifiers, attributes and trees etc. and are stored in a *service table*. A service table can be maintained at dedicated directory servers or by the individual nodes.

Suppose,  $S$  represents the set of services present in a network  $G$  and the set  $S_i = \{s_{i1}, s_{i2}, s_{i3}, \dots\}$  represents the services known by node  $n_i \in N$ , then  $S =$

$S_1 \cup S_2 \cup \dots \cup S_{|N|}$ . If  $s$  represents a service to discover, then the service discovery problem can be stated as follows:

$$\text{discovery}(s) = \{i \mid n_i \in N \text{ and} \\ \exists j: (s_{ij} \in S \text{ and } s.D \approx s_{ij}.D \\ \text{and } s.Q \leq s_{ij}.Q)\} \quad (1)$$

Although the topic of service discovery has been under research for last few years, yet service discovery in Mobile Ad hoc Network is not a straightforward job due to its exclusive nature. The primary challenges that arise during service discovery in MANET are due to the rapid topological changes, the heterogeneous character and limited capabilities of nodes. These attributes call for a cross layer and scalable semantic service discovery scheme for MANET.

### 1.1 The need for Cross Layer Service Discovery in MANET

In MANET, the mobility of nodes and the unsteady network links leads to rapid fluctuations in the service availability information. Unfortunately, traditional discovery algorithms were designed assuming a steady environment and have little information about the topological changes. Such algorithms fail to perform in MANET as they encounter difficulties in maintaining an updated state of the services available on the network. This stipulates the need for fresh discovery algorithms that can leverage information from the other layers of protocol stack to maintain the services information. The answer to this requirement is cross-layer protocol. The *cross-layer service discovery* can be defined as an approach that closely integrates the discovery layer with other layers (like routing) of protocol stack [17].

The benefits of cross-layer approach are multifold. The performance of discovery process gets improved due to reduced network overhead. As the service discovery process has the information about network happenings, it becomes quite easier to track and respond to network events in timely fashion. Finally, the service selection process can be improved when two or more service providers are available for same service. The service provider can be selected by exploiting the information from other

protocol stack (e.g. routing metrics, QoS parameters etc.).

## 1.2 The need for scalable semantic service discovery in MANET

Another challenge that arises during service discovery in MANET is due to the dearth of a definite mechanism to describe the services[36]. This heterogeneous outlook of MANET leads to interoperability issues. To confront this challenge, newer approaches are desirable that can describe the services and the corresponding domain information precisely.

Due to the improvised and unrestrained nature of MANET, the nodes in MANET (with diverse meta-data requirements) can join and leave the network any time. So, the required schema information (and their size) can't be determined apriori. This can be tackled by keeping a monolithic schema embodying all possible semantic information. However, this is certainly not desirable, because of the insufficient resources of the nodes [4].

To work out this issue, this paper recommends the maintenance of schema information in on-demand fashion via discovery of ontology from other nodes on the network.

## 1.3 Paper's Contribution & Organization

In view of the discussions above, this manuscript presents a cross layer scalable semantic service discovery scheme for MANET based on our earlier work reported in [7] and [27]. The proposed service discovery algorithm runs at the network layer to discover the provider of a particular service in parallel with the discovery of routing path from the supplicant to the provider. This improves the time duration required to access a particular service. Since, the scheme runs at the network layer, it is conscious of the dynamic changes occurring in the network. Thus the proposed scheme can take the actions accordingly in response to any event in the network. Drawing on this feature, a consistency management scheme has been proposed to maintain an up-to-date view of the services available on the network. In order to address the scalability and heterogeneity issues, a multi-tiered and scalable approach to

knowledge representation has been proposed. A general purpose ontology called *Software Ontology for Ad hoc and Vehicular Network Application (SLAVE)* has been presented that instead of maintaining a oversize schema, keeps fractional views of the domain concepts thus lowering the resource requirements.

Rest of the paper discusses the proposed scheme. Starting with the literature review, the details of the proposed scheme are presented in subsequent sections. This is followed with a look at the implementation details of the proposal and an analysis of the corresponding results. The manuscript finally concludes with a discussion on the future work.

## 2. Literature Review

There has been a plethora of research literature that talks about the service discovery in wireless sensor network (WSN) as well as mobile ad hoc networks. The service discovery in WSN is primarily qualified by their application specific nature (that lets the nodes works jointly to achieve a particular objective). Proposals relevant in the domain of WSN include Directed Diffusion, Tiny DB and Energy-Efficient Sensor Network (EYES) [4,5]. Directed Diffusion is an efficient data dissemination and aggregation scheme for WSN. TinyDB is a WSN based system featuring an acquisitional query processing that aspires to meliorate the performance based on querying location and frequency etc. The EYES proposal is an integrated network layer service discovery approach for WSN.

As MANETs are highly dynamic network with application agnostic properties, the discovery problem in MANET is distinctive from WSN. The initial approaches to service discovery in MANET mainly contemplate on the aspects related to architecture of service discovery schemes. These approaches include JINI[8], Salutation[9], Bluetooth Service Discovery protocol[10] and Service Location Protocol[11] etc. The *Sun Microsystem's JINI* is basically proposed for any distributed computing environment and is dependent on three fundamental protocols. The discovery protocol of JINI is invoked by a newly arrived to find out a lookup server on the network. Once the lookup server is discovered, the

newly arrived node registers its services' information to the lookup server by means of the join protocol. The third protocol used by JINI is the lookup protocol that is executed when a service consumer intends to discover a particular service. The service's interface is passed to the lookup protocol which searches the lookup server for matching services. Similar to JINI, the protocol *Salutation* relies on a dedicated set of hosts called salutation manager which is responsible for handling the service requests. The *Blue-tooth service discovery protocol* is a P2P protocol that uses a request-response mode of communication to execute the service discovery process. The scheme uses a set of attributes to describe the services and provides the facility to browse, search and invoke a particular service. The *Service Location Protocol (SLP)* by IETF is a hybrid scheme that can operate in directory-based and directory-less modes. In the directory-based mode, there are directory servers maintaining the information about the services and can be queried by consumers (user agents) to lookup the services. In the directory-less mode, every node in the network maintains the information about services locally in a service table. A service agent periodically advertises the services that can be cached at the consumers' nodes.

Unlike the conventional approaches, there is a second category of service discovery algorithms that emphasize on describing the services using rich representation languages[12] like Extensible Markup Language (XML), DARPA Agent Markup Language (DAML) and Web Ontology Language (OWL) etc. These approaches include Microsoft's Simple Service Discovery Protocol (SSDP)[13], DReggie[14], Group based Service Discovery (GSD) [15] and Konark[16] etc. *SSDP* is a peer-to-peer approach for discovery of services in small appliances. The protocol describes the services using XML and to invoke the services Simple Object Access Protocol (SOAP) is proposed. Another distinctive feature of SSDP is the auto configuration of node's address in the absence of any DHCP servers. *DReggie* is an enhancement of the Sun's JINI to describe the services semantically using DAML. The lookup server of JINI is extended by introducing a prolog based reasoner thus boosting the capability of matching process. *GSD* scheme exploits the inherent grouping traits present in the class and

subclass relationships of ontologies to propagate the discovery requests in selective fashion. By using this slanted approach to request propagation, the scheme claims to improve the network traffic. *Konark* is a P2P service discovery scheme based on periodic announcements of service information by nodes. The services are described in Web Service Description Language (WSDL) and can be discovered by nodes locally or using a pull-based mechanism.

Considering the dynamic temperament of MANET, it has been greatly emphasized that protocols for MANET must utilize the information available across the layers of TCP/IP [17]. Realizing the demand, a host of approaches based on *cross-layer interaction* has been proposed for service discovery in MANET. In [18], the authors integrate the Ad hoc On-Demand Distance Vector (AODV) routing protocol with the service discovery process by proposing a set of extension fields for AODV header; thus enabling the establishment of routes with the service discovery process. This lessens the time interval between issuing the request for a service and accessing the service. To curb the latency further, a proactive advertising component has been proposed in [7] as an extension to the service discovery scheme proposed in [18]. This proactive component pushes the service information with corresponding routes to its flanking nodes at fixed intervals; thus reducing the network traffic and latency. This proposal is further enhanced in [19] where a network layer service discovery scheme is proposed based on AODV and Dynamic Source Routing (DSR) protocols as the underlying routing protocol. The work also proposes to exploit inherent association among services and attaches responses of candidate future requests with the answer of the current request. This piggybacking of potential requests' answers with current response boosts the hit ratio of service consumers. Another relevant approach based on data mining has been advocated in [33]. Data mining has been employed to discover associations between user's context and the services and based on this mining oriented profiling approach, different levels of personalization have been achieved.

Similar to the integration of network layer with discovery process, [31] discusses the integration of Domain Name Service (DNS) with the discovery

process (via multicasting and distributed naming services etc.) and suggests several improvements for reducing traffic overhead and processing power.

In [20], a correspondence has drawn between an electrostatic field and the MANET. The services are modeled as positive point charge and the requests are considered negatively charged point. Every network entity calculates a potential value and services are discovered by routing the request packets towards the neighbor with highest potential. Based on this correspondence, a service discovery scheme has been proposed.

To curtail the network traffic generated during discovery of services, a bloom filter based approach has been proposed in [21] that aspires to minimize the flood storm problem in the network. The bloom filter data structure models the potential neighbors likely to contain a particular service. The discovery requests are routed towards prospective nodes expected to contain services. In [22], a service discovery scheme is presented based on specialized nodes called virtual backbones. These special nodes maintain the service information about various nodes in the network and are liable to entertain the requests of supplicant nodes in the network. Similar to GSD, the proposal *Allia*[23] is based on institution of alliances among a set of nodes. A service consumer floats its request originally to its alliance that can be propagated on the network depending upon the alliance's response. The protocol *IBM DeepSpace*[24] is a P2P protocol based on periodic broadcasting of services information. Every node disseminates information about services in clever fashion such that the advertisement by neighboring nodes doesn't coincide. In [26], *content-based routing* scheme is proposed where the messages are routed based on a predicate of key-value pairs. [27] presents the notion of semantic routing that takes into account details like sender, receiver and metadata etc. for propagating a service discovery request towards the service provider.

In this paper, we present a network layer service discovery scheme based on [7] and [27] that aspires to improve the robustness and scalability of the discovery process. The approach differs from current literature in two ways. First, it exploits routing

protocols for discovery as well as consistency management. In addition, a multi level schema management is proposed to discover and maintain ontology in on-demand fashion.

### 3. Proposed Service Discovery Scheme

After providing a gentle introduction to the topic and a review of the literature, this section talks about the details of proposed semantic service discovery scheme. Figure 1 shows the block diagram depicting the major components of proposed service discovery scheme. The scheme comprises of the hybrid service discovery component responsible for actual discovery of services, a query processing component that is liable for query analysis and optimization, a context manager for acquisition and provision of contextual information and a service catalog (with the multi level schema) for maintaining the service information and corresponding metadata. The specifications of these components are presented in next section.

#### 3.1 Service Discovery Component

The *service discovery component* is responsible for finding a requested service in the surroundings of the supplicant. It comprises of three sub-components i.e. a proactive discovery component, a reactive discovery component and a consistency management component. The *proactive component* is in charge of periodic dissemination of services and relevant routes to nearby nodes simultaneously. The *reactive component* is accountable for entertaining a request on-demand by fetching the details about a provider along with the corresponding routes. The reactive component coexists with routing protocol by utilizing the delivery mechanism of underlying protocol. The final component i.e. *consistency manger* is responsible for ensuring the consistency of the services based on the information from the lower layers.

##### 3.1.1 Proactive Component

The *proactive component* is based on our earlier work presented in [7,19]. It periodically publicizes its services and hears for advertisement of other nodes using an advertisement message UST. *UST* includes the service name, functional description about the service, provider details, Quality of Service (QoS)

details and corresponding routing details to access the provider of the service.

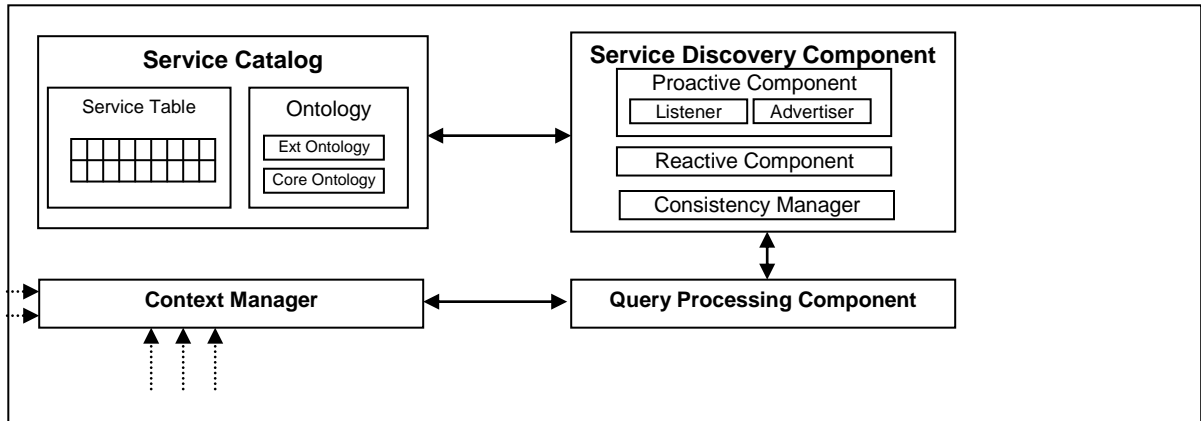
The proactive component comprises of two sub-components: an advertiser and a listener component. The *advertiser* is responsible for periodically advertising the services and corresponding information about the services to the vicinity nodes as shown in Figure 2a. The advertiser employs a sliding window mechanism to broadcast part of its service table information to adjacent nodes. Any node that receives any advertisement from neighboring nodes will accumulate this information in its local service table using the *listener component* as shown in Figure 2b.

If  $A$  is any advertisement from neighboring node and  $\oplus$  represents the operation to accumulate the

advertisement information by any node  $j$ , then we can write the accumulation process as:

$$S_j = S_j \oplus \bigcup_{a \in A} a \quad (2)$$

The proactive advertisements will lead to increased hit ratio of the service consumers. However, this comes with the cost of additional traffic overhead due that can cause the broadcast storm problem. To circumvent this problem, a number of techniques have been employed. The dissemination of packets to neighbors is done in controlled fashion at appropriate intervals. Secondly, the nodes utilize sliding window mechanism to broadcast a portion of its local catalog instead of the complete service catalog. Finally, the listener module accumulates the local service table with a received



**Fig. 1:** Block diagram illustrating the proposed scheme

```

[Advertiser]

While(System.Running()) {
    Message m;
    For i=0; i < windowSize; i++ ) {
        if( window + i > ServiceTable.Size ) {
            window = 0;
            i = 0;
            break;
        }
        Service s = ServiceTable(window + i);
        m.AddAdvertisement(s.Service(),RoutingTable(s));
    }
    window += i;
    SendToAdjacentNodes(m);
    Wait(interval);
}

```

a) Advertiser Component

```

[Listener]
While(System.Running()) {
    Message m;
    m = Listen (MessageType.UST);
    Advertisement a = m.getAdvertisement();
    Service s = new Service(a.GetService,a.GetRoute());
    If(ServiceTable(s) == Null) {
        ServiceTable.place(s);
    }
}

```

b) Listener Component

**Fig. 2:** Proactive Component of Proposed Scheme

advertisement if it doesn't have the desired services already listed in its local catalog. The final section of this paper also mentions various means that can be adopted for improving the service advertisement process. It is also to be noted that the additional overhead due to the advertisement will also be compensated by the increased hit ratio of the nodes. A significant number of requests will be satisfied locally and lead to less request propagation on the network.

The proactive advertisements will lead to increased hit ratio of the service consumers. However, this comes with the cost of additional traffic overhead due that can cause the broadcast storm problem. To circumvent this problem, a number of techniques have been employed. The dissemination of packets to neighbors is done in controlled fashion at appropriate intervals. Secondly,

the nodes utilize sliding window mechanism to broadcast a portion of its local catalog instead of the complete service catalog. Finally, the listener module accumulates the local service table with a received advertisement if it doesn't have the desired services already listed in its local catalog. The final section of this paper also mentions various means that can be adopted for improving the service advertisement process. It is also to be noted that the additional overhead due to the advertisement will also be compensated by the increased hit ratio of the nodes. A significant number of requests will be satisfied locally and lead to less request propagation on the network.

### 3.1.2 Reactive Component

The reactive service discovery component which is based on [7], works by disseminating the discovery requests using an underlying routing

protocol. Whenever a node  $i$  requests for any service  $s$ , it is first looked up in the local service table. If no information is found then the reactive service discovery component is set in motion to locate any service on the network. The reactive component prepares a service requests SREQ and propagates the request on the network by sending this request to adjacent nodes  $\eta$ . The *SREQ* comprises of routing headers (similar to RREQ), the functional and QoS details of the requested service, the contextual attributes and the specifics of ontology required to understand the service. Any node  $j$  that receives SREQ checks in its local service table for the requested service  $s$ . If any matching service is found i.e.  $\forall \square_{x \in S_j} (x.D \approx s.D \text{ or } x.QoS < s.QoS)$ , a service reply SREP is generated, otherwise this request is disseminated to the neighboring nodes. While disseminating a request SREQ, the node can append the information about hops it has traversed with the packet (as in DSR) or it can make a temporary reverse route (as in AODV), such that a route can be established between a service consumer and service provider. The circulation of the SREQ continues until the request reaches to the desired provider of the service or the TTL value of the packet perishes. The *SREP* embraces the essential headers for routing (as of SREQ), the information about the requested service, the corresponding provider details and the details of the ontology required to understand the

response. The dissemination of SREP depends upon the underlying routing protocol [7,29]. In case of DSR[28], the SREP is dispatched using the traversed hops information enclosed in the SREQ. In case of AODV[6], the temporary reverse route information maintained by intermediate node is utilized to create a forward entry. The working of the algorithm is based on our initial work in [7, 19]. Figure 3 and Figure 4 show an abstract view of the proposed scheme in pseudo code format.

### 3.1.3 Consistency Manager

To maintain an up-to-date status of the services, a *network layer consistency manager* component is proposed based on the underlying routing protocol. Since, the consistency manger is executing at the network layer, it is conscious of the network proceedings. Upon any type of failure, the consistency manager component is triggered to perform local maintenance. The component tries to discover an alternate route to the provider or a substitute provider to the same service. If the all of indigenous up keeping efforts becomes futile, the corresponding nodes on the network are notified such that these nodes can invalidate their corresponding routing and service table entries. Figure 5 shows the pseudo code describing the working of proposed scheme. A new message SERR is introduced based on RERR message of the AODV protocol. The *SERR*

```

While(true) {
    Message m = Listen();
    If(m is SREQ) {
        If(m is already processed) {
            Continue;
        }
        Else If(ServiceTable(m.GetService()) <> Null) {
            Send(GenerateSREP(m));
        }
        Else {
            m.UpdateHopCount();
            m.HopsTraversed();
            Broadcast(m);
        }
    }
    If(m is SREP) {
        If(m.DestinationIPAddress = LocalAddress) {
            ServiceTable.Add(m.GetService());
            RoutingTable.Add(m.Destination,m.Source);
        }
        Else {
            RoutingTable.Add(m.Destination,m.Source);
            Send(m);
        }
    }
}

```



**Fig. 3:** AODV-based On-Demand Service Discovery Component[7,19]

```
While(true) {  
    Message m = Listen();  
    If(m is SREQ) {  
        If(m is already processed) {  
            Continue;  
        }  
        Else If(ServiceTable(m.GetService()) <> Null) {  
            Send(GenerateSREP(m));  
        }  
        Else {  
            m.UpdateHopCount();  
            Broadcast(m);  
            RoutingTable.Add(m.Originator,m.Source);  
        }  
    }  
    If(m is SREP) {  
        If(m.DestinationIPAddress = LocalAddress) {  
            ServiceTable.Add(m.GetService());  
            RoutingTable.Add(m.Destination,m.Source);  
        }  
        Else {  
            RoutingTable.Add(m.Destination,m.Source);  
            Send(m);  
        }  
    }  
}
```

**Fig. 4:** DSR-based On-Demand Service Discovery Component[7,19]

```

While(true) {
    If(System.FailureEvent())
        List AffectedServices = Get services whose provider is accessible through unreachable node
        List AffectedNodes = Get nodes that becomes unreachable because of failure
        If(PerformLocalRouteMaintenance() = "Success") {
            Return;
        }
        Message SERR = GenerateServiceError(AffectedServices, AffectedNodes);
        Int Status = InitiateSREQ();    [perform local Service maintenance]
        If(Status = "Success") {
            ServiceTable.Update (AffectedServices);
            SERR.SetFlag(Flag.N);
        }Else {
            ServiceTable.Mark (AffectedServices, "Unavailable");
        }
        Send(SERR);
    }
    If(m is SERR) {
        If(PrecursorsList <> Null) {    [intermediate nodes]
            Retransmit(m);
        }
        If(m.Flag("N") = True) {
            ServiceTable.Update(m.GetService());
            RoutingTable.Update(m.GetRoute());
        }Else{
            RoutingTable.Mark(m.AffectedNodes, "Invalid");
            ServiceTable.Mark(m.AffectedServices, "Unavailable");
        }
    }
}

```

**Fig. 5:** Pseudo code describing the operation of Consistency Manager

message extends RERR message with an additional field *Affected Services* that holds the list of services affected due to the link failure. The algorithm first attempts the local repair of the route and if it results in success, the necessary operations are performed similar to [6] i.e. by issuing a RERR message notification. In case of failure to find an alternate route, the algorithm tries to discover an alternate service provider by launching a SREQ. If this results in success, the service tables are updated and an SERR message is sent with the N flag set to true. In case of no alternate service provider, service table entries are invalidated and SERR message is floated to the upstream nodes. Whenever, a node receives SERR message, it attempts to retransmit the message to precursor nodes, as in the case of RERR handling in [6]. Depending upon whether the N flag is true or false, the corresponding service and routing tables are updated.

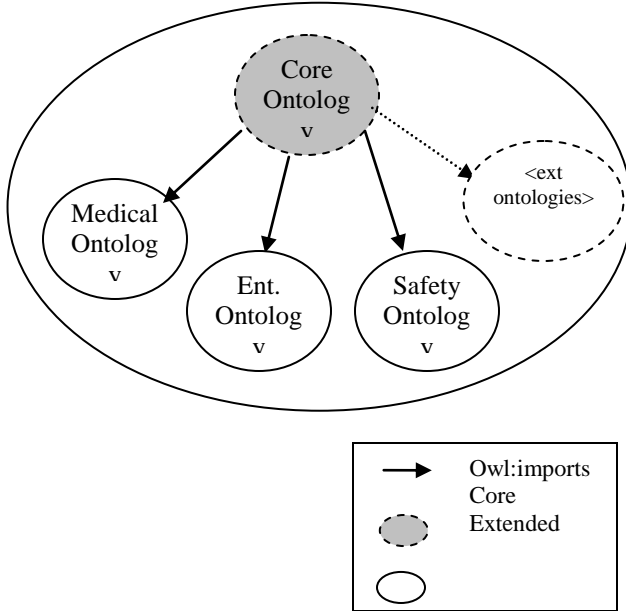
### 3.1.4 Service Catalog

The service catalog maintains a repository of services present on the network as well as the meta-data required to comprehend the services. There is a

service table that holds the information about the services hosted by the local node or by other nodes in the network. The ontology component maintains the schema information for understanding the meaning of the resources, data and services. Because of the ajar nature of MANET, it is very difficult to maintain a monolithic ontology document describing the concepts altogether. However, it is vital to maintain the meta information to address the interoperability issue. Therefore, a scalable multi-tier scheme based on ontology is presented that maintains partial views of the global schema at individual hosts of the network.

We present in this paper SLAVE as a sample ontology to describe the multi-level schema management approach. SLAVE describes the vocabulary related to ad hoc and vehicular network applications. The objective is to maintain the general concepts in a basic ontology document called *SALVE Core* while for coping with the concepts varying across different applications, an escalating knowledgebase (*SLAVE Ext*) is maintained that is evolved in cold start fashion. Figure 6 shows the

general layout of SLAVE ontology. The SLAVE *core ontology* describes the general concepts and the *ext ontology* documents (medical, safety and entertainment etc.) import the core ontology to further describe the concepts for specialized domains.



Fig, 6: SLAVE Ontology

#### 3.1.4.1 SLAVE Core Ontology

Figure 7 illustrates a partial view of core ontology. The *Profile* is a generalized concept that is extended by specialized concepts i.e. Computational Profile, Driving Profile and Spatial Profile etc. The *Computational Profile* possesses attributes pertaining to any computational gadget like memory, processing cycles and battery power etc. The *Driving Profile* characterizes the driving behavior of any driver. The *Spatial Profile* keeps the spatial attributes of any object like the cartesian coordinates.

The concept *Ad hoc Node* enfoldes the details about any host in the network like its computational and spatial profile and the services hosted by the node. The *Service* concept describes any service hosted on the network including the corresponding routing information to reach to the service. The *Person* concept has the information about the person owning the node i.e. name, age, gender etc. The *Person* class is extended by the *Driver* class to describe details about the driver of VANET Node.

The concept *Place* describes any place by attributes like its coordinates etc.

#### 3.1.4.2 SLAVE Ext Ontology

The Ext Ontology comprises of a collection of documents modeling the vocabulary for a range of applications like safety, medical and entertainment applications. Additional ontology documents can also be defined based on SALVE Core Ontology.

##### a) Medical Ontology:

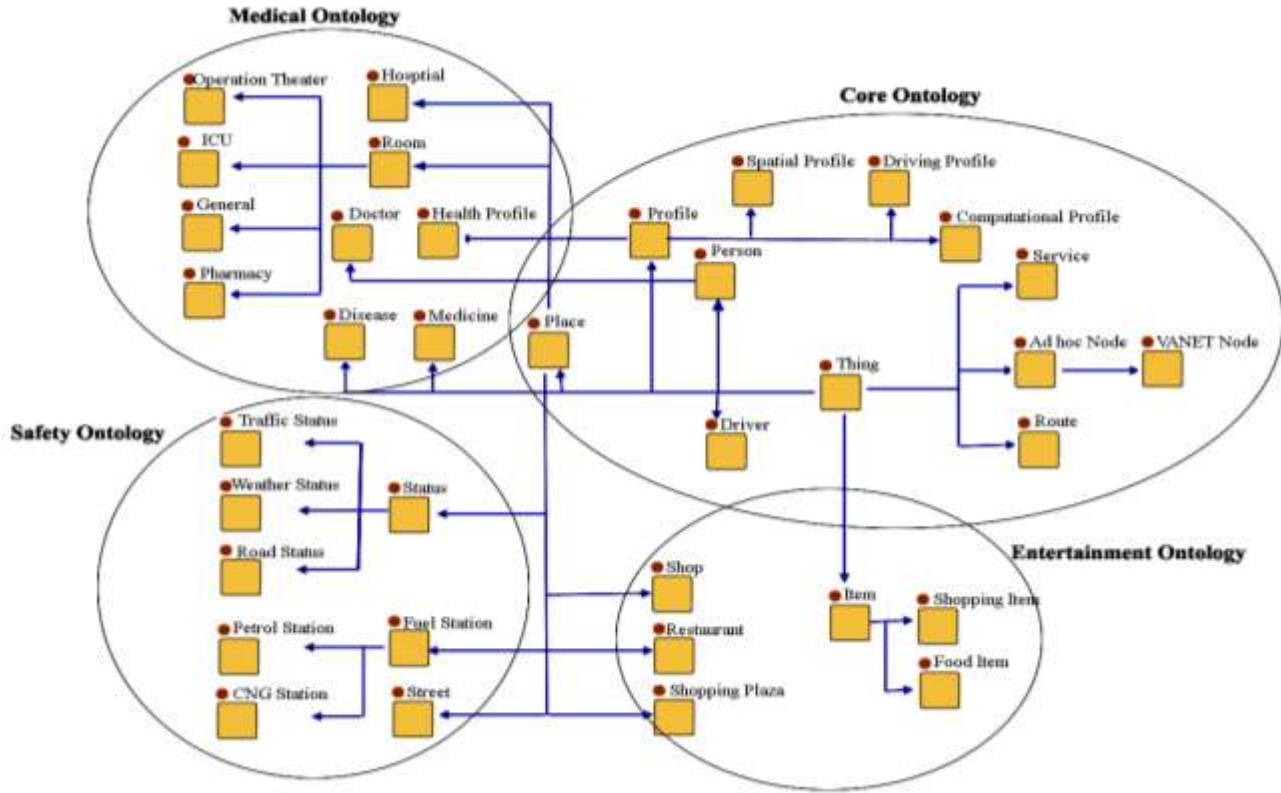
This ontology is useful for applications in the domain of health care. The *Doctor* is a sub-concept of *Person* and includes attributes like qualification of doctor and the *Disease* he can treat etc. *Hospital* concept maintains information like the doctors working in the hospital and different types of *Room* (*General*, *Operation Theater*, *Pharmacy* and *ICU*) etc. The *Driver* concept has been overridden to include a new relationship with the concept of *Health Profile*. The *Health Profile* maintains information about health of an individual. The *Medicine* concept maintains details about a particular medicine.

##### b) Safety Ontology:

A safety application (in the domain of vehicular network) ensures risk free driving experience of passengers. Figure 7 outlines the proposed safety ontology based on the core ontology. It introduces several new concepts like *Street*, *FuelStation* and *Status*. The *Street* class extends the *Place* class to describe the street in which a particular host is travelling. The *FuelStation* extends the *Place* concept and describes two sub-concepts *PetrolStation* and *CNGStation*. The *Status* class maintains different types of statuses relevant for safety applications like *WeatherStatus*, *TrafficStatus* and *RoadStatus*.

##### c) Entertainment Ontology:

This ontology can be useful for different types of entertainment applications like shopping etc. It introduces several new concepts. The *Place* concept is extended by *Restaurant*, *Shopping Plaza* and *Shop*. The *Restaurant* concept describes any particular restaurant that prepares and sells food. It is linked with the *Food Item* via the relation *prepares Food*. The *Shop* concept is also linked with *Shopping Item* via the relation *provide Item*.



**Fig. 7:** A Software Ontology for Ad hoc and Vehicular Network Applications (SLAVE)

### 3.1.5 Query Processing Component

The *Query Processing Component* is in charge of analysis, planning, construction and the optimization of a context-sensitive query corresponding to the requested service by the supplicant. A *query* comprises of the requested service and its description, desired QoS, the contextual attributes and the name of the ontologies crucial to understand the query. Since, it is not viable to maintain a unified schema embracing entire concepts at a particular host; it is likely that the quoted ontology is not available at any node. Hence, the corresponding ontologies are often loaded from the other hosts of network using the reactive service discovery component. To discover a particular ontology on the network, the reactive service discovery component is prompted with the ontology's name as the desired service to be sought. The requested ontology is discovered using the reactive component and then stored locally in ext ontology database. Figure 8 shows the pseudo code describing the process. The matching component of the discovery process first looks in the catalog for the

ontology required for understanding the request. If the ontology is not available, a new discovery request is launched to discover the required ontology. Once the ontology is discovered and put aside in catalog, further semantic matching is performed to satisfy the original request.

The query processing besides holding the responsibility of query composition and execution can also perform a range of other tasks. This includes query optimization to reduce the processing time and end-to-end query execution latency etc.

### 3.1.6 Context Manager

A *Context Manager* is obliged of obtaining the contextual data and endowing this information on-request to any other component. A *context* is a set of attributes essential to characterize the current situation. Examples of contextual information include current time, current date and location of any node etc. The contextual information is primarily used by querying component for the composition of a context-sensitive query.

```

Public boolean Match (Service service) {
    String ontology = service.requiredOntology
    If (!Catalog.Contains(ontology) {
        Catalog.Add(ReactiveComponent.Discover(ontology));
    }

    //construct model and reasoner
    Model m;
    ...
    //initialize reasoner
    Reasoner reasoner;
    ...

    //create corresponding query
    String query;

    //Execute query
    QueryExecution qexec = QueryExecutionFactory.create(query,model) ;

    //now extract relevant information
    ...
}

```

**Fig. 8:** Pseudo code describing how ontologies are discovered and maintain in multi-tier configuration

#### 4. Simulation Details and Results

This section talks about the specification and analysis of the various experiments conducted to scrutinize the proposed scheme. The proposed scheme has been simulated in Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator (JIST/SWANS) [30]. The SLAVE ontology has been coded in the OWL language [31] using the ontology editor Tool for Ontology Development and Editing (TODE) [32]. For various inferencing and reasoning tasks, the JENA toolkit [33] has been used. A suite of experiments have been performed (with the parameters shown in table 1) to assay the robustness and scalability of the scheme. The results of the simulation have been analyzed through various statistical techniques e.g. graphs, descriptive statistics and t-testing as discussed in following sub-sections.

**Table 1:** Simulation Parameters

Parameters	Value
Simulation Area	500 × 500
Node Placement	Random
Cache Replacement	FIFO
Simulation Time	5000
Mobily Model	Waypoint

##### 4.1 Analyzing the performance of Semantic Service Discovery Component

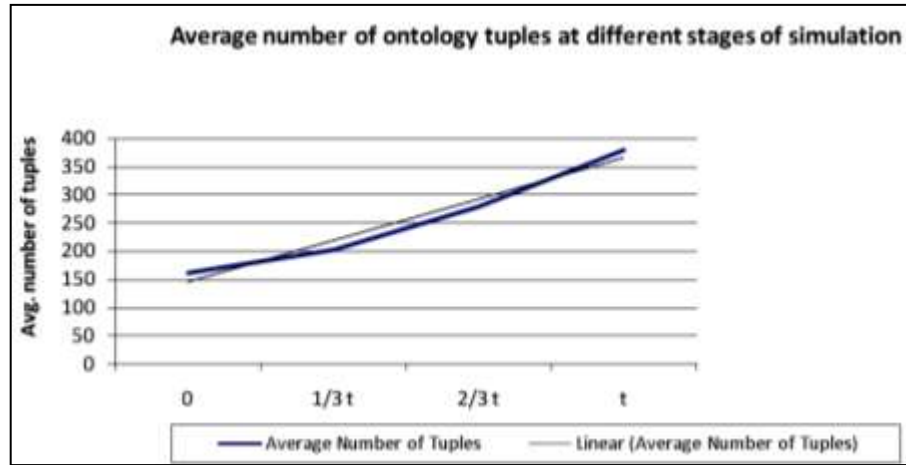
Figure 9 analyzes the *degree of schematic information* maintained by nodes at various stages of simulation for an experiment performed with 50 nodes and 25 services running AODV protocol.

Figure 9a shows the average number of ontology tuples maintained by network hosts during simulation along with a linear trend line drawn based on regression analysis. The average number of ontology tuples maintained at a node rises smoothly with the rise in time. Figure 9b displays the sub-divided bar diagram for the total number of ontology tuples present at individual nodes during simulation. It can be construed from these graphs that the nodes' requirement for schematic information mounts up gradually with time.

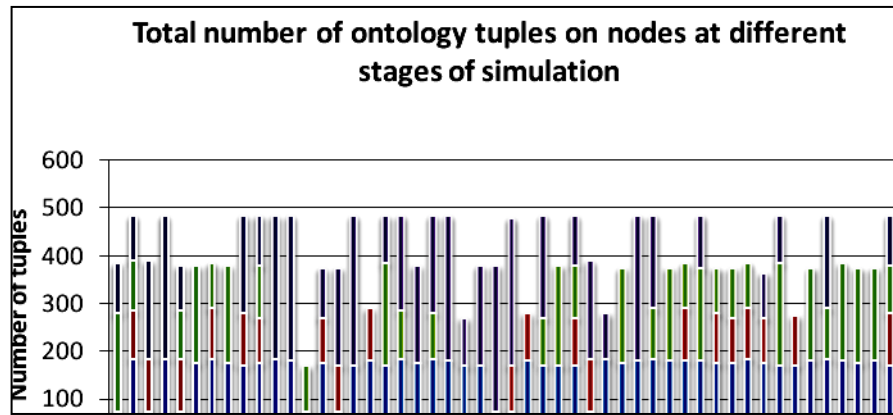
To assert this interpretation further, an unpaired t-test assuming unequal variance is performed on two datasets  $D_1$  and  $D_2$ . The first dataset  $D_1$  contains the average number of tuples maintained by nodes (as obtained from the experiment). The second dataset  $D_2$  is an optimal dataset constructed by allotting the entire set of ontology tuples  $O$  evenly across the time line of simulation  $t$ . The construction of dataset  $D_2$  can be expressed mathematically as follows:

$$D_2(i) = i * (|O| / t) \quad (3)$$

The t-test analyzes the two datasets to establish *if the trend of  $D_1$  is analogous to  $D_2$*  (the null hypothesis  $H_0$ ) or *if  $D_1$  and  $D_2$  are two different data sets* (the alternate hypothesis  $H_a$ ). Figure 9c shows the specification of t-test and the results obtained from the test. The one-tail  $p$  and two-tail  $p$  values are 0.144 and 0.282 respectively as shown in Figure 9c. Since  $p$  value is greater than 0.005, the null hypothesis has been established.



a) Average number of ontology tuples maintained by nodes and the trend line based on regression analysis



b) Bar Graph illustrating total number of ontology tuples maintained by nodes

Hypotheses			
$H_o$	The datasets $D_1$ and $D_2$ have similar trend		
$H_a$	The datasets $D_1$ and $D_2$ have non-similar trend		
Results of Tests			
$Parameter$	$Value$	$Parameter$	$Value$
alpha level	0.050	t stat	1.220
t critical one tail	2.132	P one tail	0.144
t critical two tail	2.776	P two tail	0.282

c) Output of t-test comparing dataset D1 and D2

#### Simulation Parameters

Simulation Time:	5000s	No. of Nodes:	50
Routing Protocol:	AODV	No. of Services:	25

**Fig. 9:** An analysis of the scalability of the proposed scheme

Figure 10 analyzes the impact of multilevel schema management on latency with an experiment

conducted on 100 nodes and 25 services running AODV protocol. During earlier stages of simulation, the latency incurred with multilevel schema is comparatively larger, as it requires discovery of

ontology documents that are not maintained locally. However, afterwards, the latency of the multilevel schema scheme is better than the monolithic schema. To be precise, the reduction in latency with multilevel schema management is ( $M=2.2$ ,  $SD=0.97$ ) with a 95% confidence interval (0.3,4.1).

Although the proposed semantic approach addresses the syntactic and semantic interoperability issues in scalable manner however, this compatibility comes with the slight degradation in performance. To analyze the impact of semantics on latency, a simulation was performed with 100 nodes and 25 services on the network with DSR routing protocol. Figure 11 shows the results of the simulation in graphical format. The various parameter of the simulation are also mentioned underneath the figure. The graph clearly points out that the semantic service discovery give rise to latency slightly above the non-semantic case. The average rise in latency ( $M=10.53$ ,  $SD=1$ ) with a 95% confidence interval about mean difference is (8.88, 12.15). The reason behind the rise in latency is the increased computational overhead by to lookup the requested service based on semantics.

## 4.2 Analyzing the performance of Proactive Broadcasting Module

To understand the effect of proactive service advertisement, a simulation experiment is conducted with 50 nodes and 25 services. The latency of the services is analyzed when proactive advertisement of services is performed (broadcasting case) against the case when no advertisements are floated to the flanking nodes (non-broadcasting case). In the later case, the services are discovered entirely in on-demand fashion.

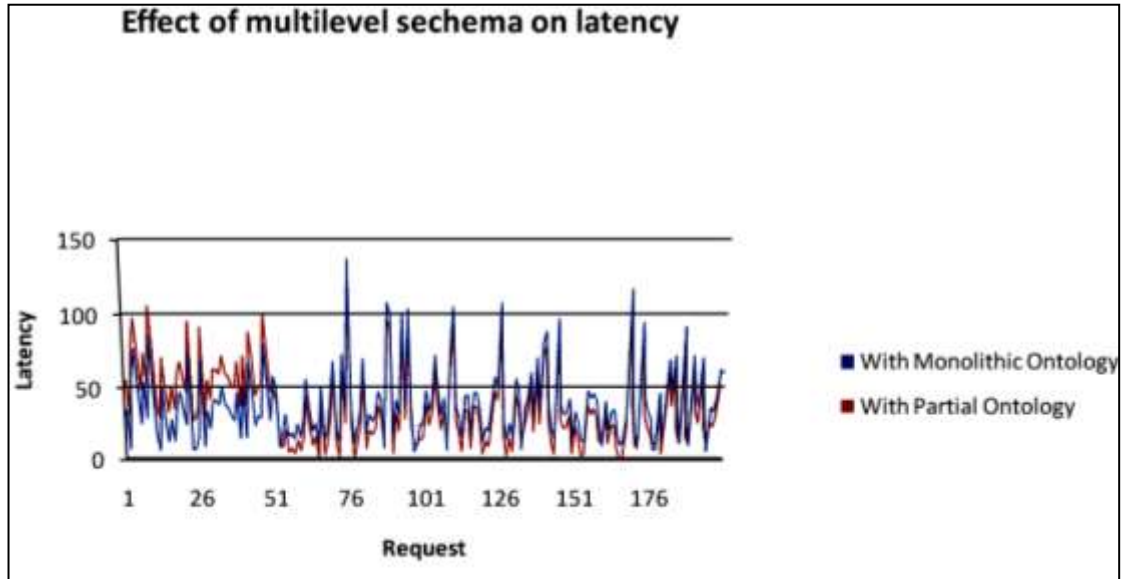
Figure 12 compares the *latency* incurred in looking for and accessing the services for the two cases. The various parameters of the experiment are listed at the bottom of the graph. With the exception of few cases, the latency of the non-broadcasting is greater than the broadcasting case. The average latency of the broadcasting scheme (i.e. 21.4s) is better than the non-broadcasting scheme (i.e. 26.4s). This evidently suggests that latency is improved due

to the attachment of a proactive advertising module with the on-demand service discovery component. The minor instances where the results are contrary are due to reasons like slamming of an advertisement packet with other packets (request or response packets), issuance of request for a packet before it is actually advertised on the network and network failures exclusively in the broadcasting case as compared to the non-broadcasting case etc.

To affirm the argument that the floating of services in advance leads to improvement in latency, an unpaired t-test assuming unequal variance is performed on the data acquired from the experiment. The t-test illustrates that the average reduction ( $M=8.06$ ,  $SD=38.25$ ) in latency is quite significant with one-tail  $p=0.003$  and two-tail  $p=0.007$ . Since  $p$ -value is less than 0.005, we can reject the null hypothesis. Furthermore, a 95% confidence interval about mean difference is (2.72 sec, 13.39 sec). Hence, we can conclude that proactive broadcasting leads to improved latency in the range (2.72 sec, 13.39 sec).

## 4.3 Analyzing the robustness of Consistency Manager

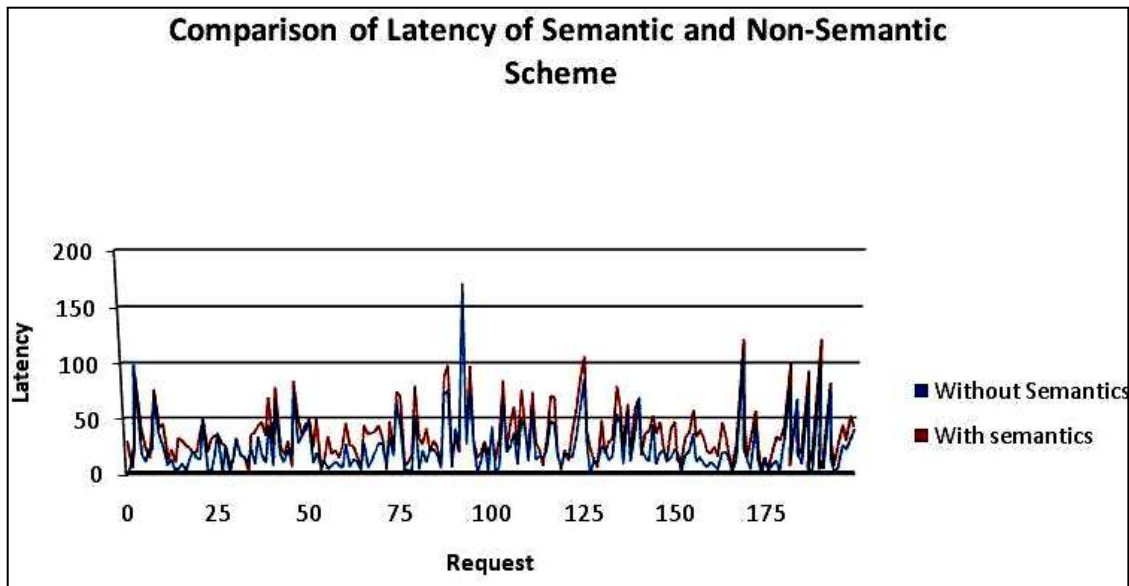
To improve the robustness of service discovery process, a consistency manager is proposed that functions at network layer and is perceptive to network events. This section reflects on the performance of the consistency component. A robust service discovery component is vigilant against the network incidents and always maintains up-to-date view of the services available on the network. This leads to reduced network latency and high hit ratio of service consumers. Figure 13a compares the *latency* incurred in accessing the services when the hosts in the network are running the proposed consistency manager against the case where no consistency manger is available. The results of the t-tests are shown in Figure 13b. By employing a consistency manager, the latency gets reduced as the most of the hosts in the network maintain an updated status of the services available on the network. This avoids the overhead that could incur when an unavailable service with false state in service table is sought out (as is done without a consistency manager).



#### Simulation Parameters

Avg. Latency (monolithic schema):	38.466 s	Avg. Latency multilevel schema:	36.16291 s
Simulation Time:	5000s	No. of Nodes:	100
No of requests:	200	No. of Services:	25
Routing Protocol:	AODV		

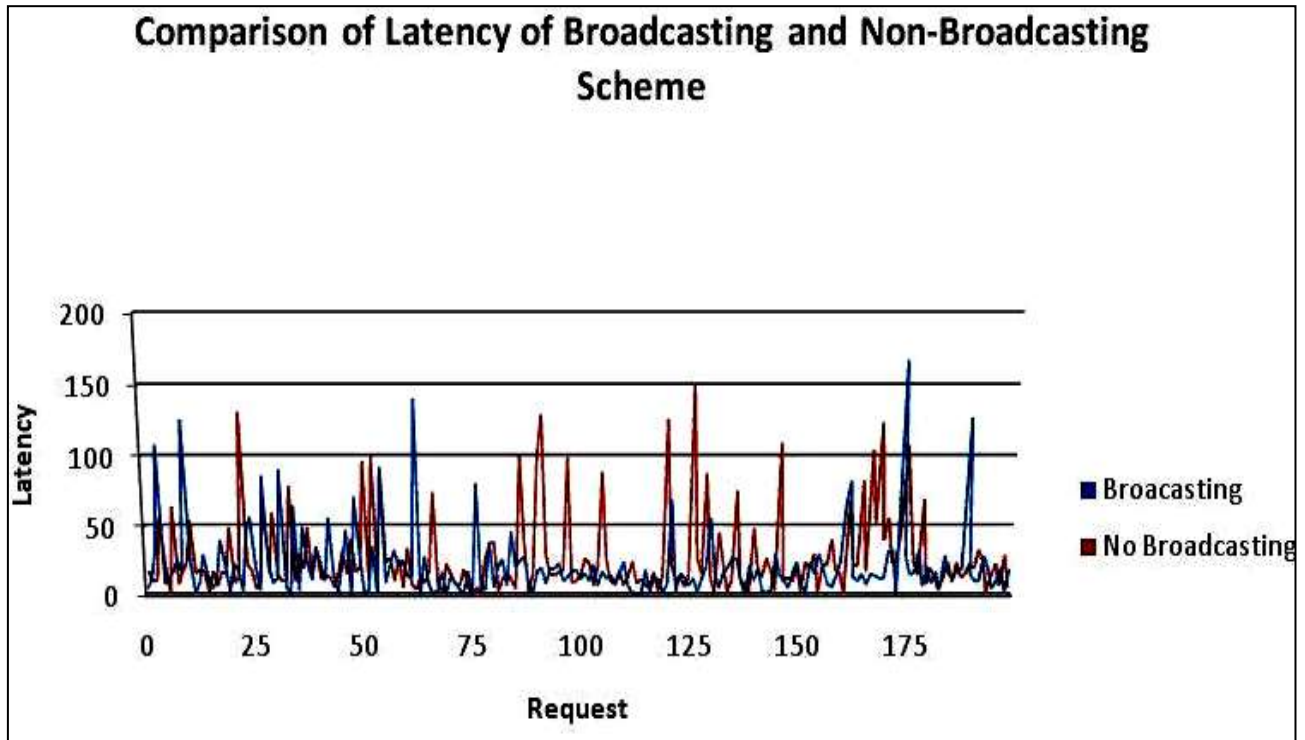
**Fig. 10:** Line graph analyzing the impact of multilevel schema on latency



#### Simulation Parameters

Avg. Latency with semantics:	34.86 s	Avg. Latency without semantics:	23.86 s
Simulation Time:	5000s	No. of Nodes:	100
No of requests:	200	No. of Services:	25
Routing Protocol:	DSR		



**Fig. 11:** Line graph comparing the latency of semantic and non-semantic service discovery schemes

a) Line Graph comparing the latency of broadcasting and non-broadcasting schemes

Hypotheses	
$H_0$	latency reduction due to proactive advertisement is zero
$H_a$	latency reduction due to proactive advertisement is not-zero

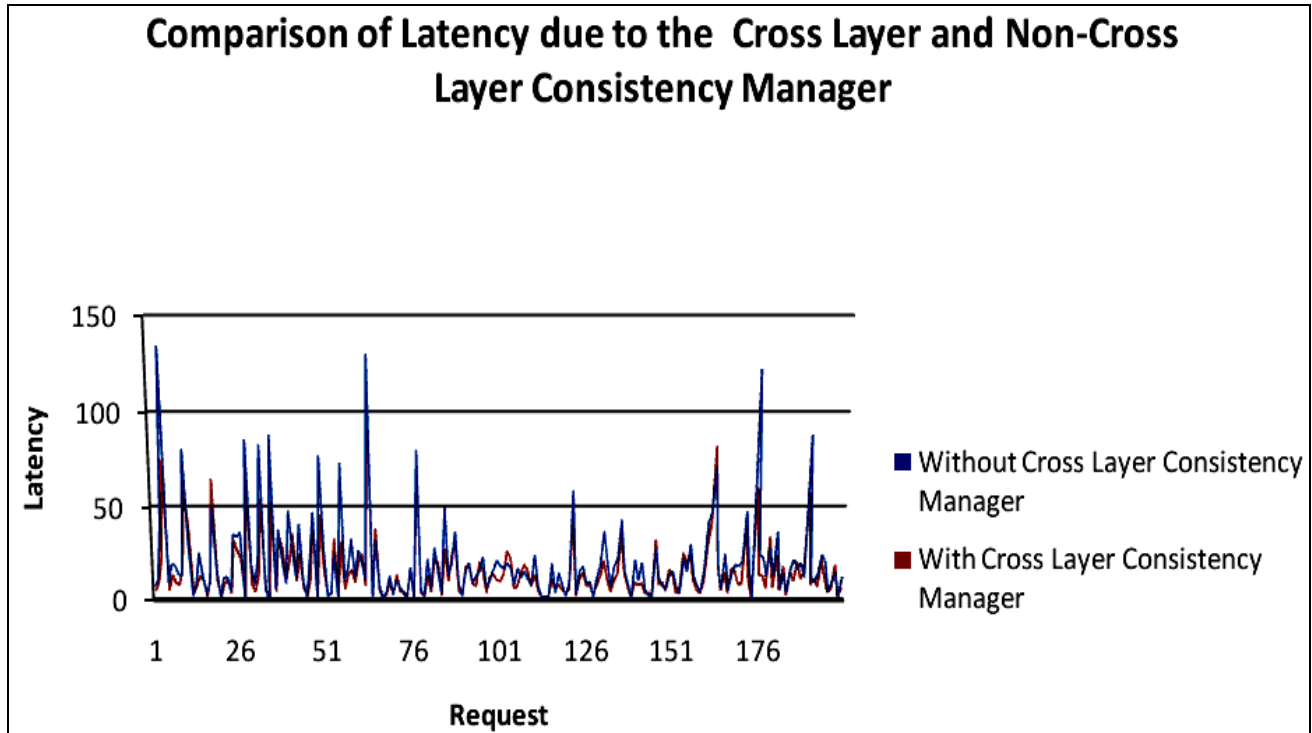
Results of Tests			
Parameter	Value	Parameter	Value
alpha level	0.05	t stat	-2.73
t critical one tail	1.65	P one tail	0.003
t critical two tail	1.97	P two tail	0.007

a) Output of t-test, comparing latency of broadcasting and non-broadcasting schemes

**Simulation Parameters**

Avg. Latency with broadcast:	21.4 s	Avg. Latency without broadcast:	26.4 s
Simulation Time:	5000s	No. of Nodes:	50
No of requests:	200	No. of Services:	25
Routing Protocol:	AODV	Advertisement Interval:	6 s

**Fig.12:** Comparison of latency for the broadcasting and non-broadcasting schemes



a) Line Graph comparing the latency of two schemes

Hypotheses			
$H_o$	latency reduction due to consistency manager is zero		
$H_a$	latency reduction due to consistency manager is not-zero		
Results of Tests			
$Parameter$	$Value$	$Parameter$	$Value$
alpha level	0.05	t stat	2.49
t critical one tail	1.65	P one tail	0.006
t critical two tail	1.97	P two tail	0.001

b) Output of t-test, comparing the two schemes

#### Simulation Parameters

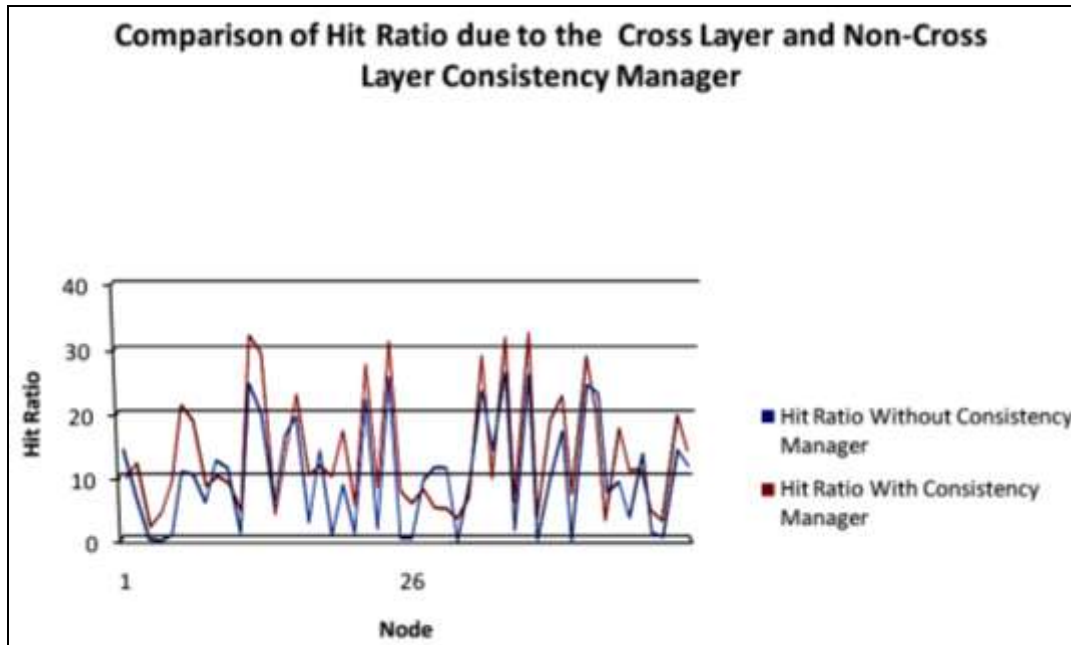
Avg. Latency without Consistency Manager: 20.08 s    Avg. Latency with Consistency Manager: 16.41 s  
 Simulation Time: 2000s    No. of Nodes: 50  
 No of requests: 200    No. of Services: 10  
 Routing Protocol: AODV

**Fig. 13:** Comparison of latency with and without the Consistency Manager

Figure 14a analyzes the hit ratio incurs due to the two schemes. The average hit ratio by employing a consistency manager is better than without a consistency manager. A t-test was also performed on the results obtained from the experiment and its results are showing in Figure 14b. The t-test validates the efficacy of the consistency manager. Based on the graph results and t-tests, it can be concluded that the proposed consistency manager discovers the services robustly with an improvement in latency about (3.033 sec, 5.877 sec) and enhancement in hit ratio about (3.389 %, 5.645%).

## 5. Conclusion

The outset of emerging computing paradigms demands a fresh look at traditional issues of the network (e.g. routing, service discovery, security etc.). This paper presents an innovative model for service discovery based on the inherent characteristics of Mobile Ad hoc Network. The scheme proposes the integration of service discovery and routing layer by providing extensions on earlier work [7,18]. An evaluation of these extensions (proactive module) was performed against [18] and these evaluations suggest improvement in latency of the service discovery process.



a) Line Graph comparing the hit ratio due to two schemes

Hypotheses			
$H_o$	hit ratio improvement due to consistency manager is 0		
$H_a$	hit ratio improvement due to consistency manager is not 0		
Results of Tests			
<i>Parameter</i>	<i>Value</i>	<i>Parameter</i>	<i>Value</i>
alpha level	0.05	t stat	-1.86
t critical one tail	1.66	P one tail	0.003
t critical two tail	1.98	P two tail	0.003

a) Output of t-test, comparing the two schemes

### Simulation Parameters

Avg. hit ratio without Consistency Manager: 10.6%      Avg. hit ratio with Consistency Manager: 14.4%  
 Simulation Time: 2000s      No. of Nodes: 50  
 No. of Services: 10      Routing Protocol: AODV

**Fig. 14:** Comparison of Hit Ratio with and without the Consistency Manager

Realizing the open nature of MANET, a multi-tier approach to describe the services is proposed. This tiered approach not only addresses the syntactic and semantic heterogeneity issues but also the scalability challenges associated with MANET. Finally, a network layer consistency manager is also proposed considering the swift nature of MANET. The proposed consistency manager leads to improvement in latency and hit ratio, as has been verified by experimental results and t-testing. A thorough comparison of the proposed consistency manager with existing schemes is left as a future work. However, based on these results, it has been substantiated that the proposed network layer semantic service discovery scheme instigates robustness and scalability in the discovery process.

The proposed service discovery scheme serves as a key enabler for pervasive environments. This allows the seamless discovery and access of heterogeneous services dispersed on unreliable, dynamic and resource constrained mobile network in a robustness fashion. A number of potential applications can be envisioned. This includes smart spaces, disaster management and intelligent transportation systems etc.

There are a number of areas where further research work can be carried out. To precisely figure out the actual gain in performance, the proposed approach requires the implementation in some real world environment with the results analyzed after large experimental runs. This manuscript discusses the process of integration of service discovery and network layer for two routing protocols. Can the same approach be mapped to other reactive routing protocols (e.g. OLSR, TORA and ABR etc.)? The overhead of cross layering also demands investigation as soft timers are usually used to update entries across the protocol stack. The matching process of reactive component at individual hosts can be made more robust by reduction of the search space using a range of techniques like longest prefix matching and lexical ordering etc. [38]. Another possible extension is the exploitation of cross layering for service provider selection, when there are multiple providers available for the same service.

The proactive advertising module shares its own knowledge with flanking nodes at regular interval. Studies are required in determining the optimal length of the interval under different network settings[34]. The total memory cost of proactive component is bounded by the service catalog size. When the total number of services available on the network falls above the catalog size, FIFO module is used. Evaluations are required for other techniques of cache management. Finally, the proposed scheme doesn't talk about security issues associated with discovery of services in hostile ad hoc network. Can the security designs like [35] be plugged in the proposed service discovery solution?

## 6 References

- [1] A. Singh, M. Kumar, R. Rishi, and D. K. Madan, "A Relative Study of MANET and VANET: Its Applications, Broadcasting Approaches and Challenging Issues," in *Advances in Networks and Communications*, Springer, (2011) 627-632,
- [2] Y. Wang and F. Li, "Vehicular Ad hoc Networks," in *Guide to Wireless Ad Hoc Networks*, Springer, (2009) 1-23.
- [3] C.K. Toh, P. Mahonen, and M. Uusitalo; Standardization Efforts & Future Research Issues for Wireless Sensors & Mobile Ad hoc Networks, *IEICE Transactions on Communications*, 9(2005) 3500-3507,
- [4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, 52(2008) 2292-2330,
- [5] V. Handziski, C. Frank, and H. Karl, "Service Discovery in Wireless Sensor Networks," Technical Report TKN-04-006, Telecommunication Networks Group, Technische Universitat Berlin (2004).
- [6] C. Perkins, E. Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing", IETF RFC 3561 (2003).
- [7] N. Islam and Z.A. Shaikh, "A Novel Approach to Service Discovery in Mobile Adhoc Network," in proceedings of *International*

- Networking and Communication Conference*, Lahore, Pakistan, (2008) 58-62.
- [8] "JINI, Technology Core Platform Specification Version 2.0.", Sun Microsystems, Inc. (2003).
- [9] B.A. Miller and R.A. Pascoe, "Salutation Service discovery in Pervasive Computing Environments", IBM Whitepaper (2000).
- [10] E.A. Gryazin, "Service Discovery in Bluetooth," Group for Robotics and Virtual Reality, Department of Computer Science. Helsinki University of Technology, Helsinki, Finland (2006).
- [11] E. Guttman, C. Perkins, J. Veizades and M. Day, "Service Location Protocol", version 2. IETF RFC 2608, 1999
- [12] L. Ding, P. Kolari, Z. Ding, and S. Avancha, "Using Ontologies in the Semantic Web: A Survey," in *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, Springer, 2006 (79-113)
- [13] "Understanding Universal Plug and Play: a White Paper", Microsoft Corporation (2000).
- [14] D. Chakraborty, F. Perich, S. Avancha, and A. Joshi, "DReggie: Semantic Service Discovery for m-commerce Applications," in proceedings of *Workshop on Reliable and Secure Applications in Mobile Environment* (2001).
- [15] D. Chakraborty, A. Joshi, Y. Yesha, and T. Finin, "GSD: A Novel Group-based Service Discovery protocol for MANETS," in proceedings of *4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002)*, Stockholm, Sweden (2002) 140-144.
- [16] N.D. S. Helal, V. Verma, and C. Lee, "Konark - A Service Discovery and Delivery Protocol for Ad hoc Networks," in proceedings of *3rd IEEE Conference on Wireless Communication Networks (WCNC)*, Dept. of Comput. & Inf. Sci. & Eng., Florida Univ., Gainesville, FL, USA , 3(2003) 2107-2113.
- [17] A. Varshavsky, B. Reid, and E.d. Lara, "The Need for Cross-layer Service Discovery in MANETs," in *Mobile Adhoc and Sensor Systems Conference*, Technical Report, CSRG-492, University of Toronto (2005).
- [18] D. Arias-Torres and J.A. García-Macias, "Service Discovery in Mobile Ad-hoc Networks by Extending the AODV Protocol," in proceedings of *2nd Mobile Computing Workshop*, Colima, Mexico (2004).
- [19] N. Islam, N.A. Shaikh, G. Ali, Z.A. Shaikh and Aqeel-ur-Rehman, "A Network Layer Service Discovery Approach for Mobile Ad hoc Network Using Association Rules Mining, *Australian Journal of Basic and Applied Sciences*, 6(2010) 1305-1315.
- [20] V. Lenders, M. May, and B. Plattner, "Service discovery in Mobile Ad hoc Networks: A Field Theoretic Approach," *Pervasive and Mobile Computing*, 3(2005) 343-370.
- [21] Z. Gao, X. Li, L. Wang, J. Zhao, Y. Zhao, and H. Shi, "BFGSDP: Bloom Filter Guided Service Discovery Protocol for MANETs," *Managing Traffic Performance in Converged Networks*, (2007) 446-457.
- [22] R.A. Mallah and A. Quintero, "A Light-Weight Service Discovery Protocol for Ad Hoc Networks," *Journal of Computer Science*, Science Publications 4(2009) 330-337.
- [23] O. Ratsimor, D. Chakraborty, A. Joshi, and T. Finin, "Allia: Alliance-based Service Discovery for Ad-hoc Environments," in proceedings of *2nd International Workshop on Mobile Commerce*, Atlanta, Georgia (2002) 1-9.
- [24] M. Nidd, "Service Discovery in DEAPspace", *IEEE Personal Communications*, 4(2001) 39-45.
- [25] E.M. Royer and C.K. Toh, "A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks," *IEEE personal communications*, 2(1999) 46-55.

- [26] A. Carzaniga, M.J. Rutherford, and A.L. Wolf, "A Routing Scheme for Content-based Networking," in *IEEE Infocomm*, (2004) 918-928.
- [27] A.J. Ford, R. Manor, "A Semantic Routing Framework," in proceedings of *Second Annual Conference of the International Technology Alliance*, London UK (2008).
- [28] D. Johnson, D. Maltz, and Y.C. Hu, "*The Dynamic Source routing protocol for Mobile Ad hoc Networks (DSR)*", IETF RFC (2003).
- [29] N. Islam and Z.A. Shaikh, "Service Discovery in Mobile Ad hoc Networks Using Association Rules Mining," in *13th IEEE International Multitopic Conference (INMIC-2009)*, Islamabad, Pakistan (2009) 1-5.
- [30] E. Weingartner, H. Vom Lehn, and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in proceedings of *IEEE International Conference on Communications (ICC '09)*, Dresden (2009) 1-5.
- [31] C. Campo and C. García-Rubio, "DNS-based Service Discovery in Ad hoc Networks: Evaluation and Improvements," in *Personal Wireless Communications*, Springer, (2006) 111-122.
- [32] N. Islam, M.S. Siddiqui, and Z.A. Shaikh, "TODE: A Dot Net based tool for Ontology Development and Editing," in *2nd International Conference on Computer Engineering and Technology (ICCET)*, China, 6(2010) 229-233.
- [33] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, and M. Marchetti, "CAS-Mine: providing personalized services in context-aware applications by means of generalized rules," *Knowledge & Information Systems*, 2010 (1-28).
- [34] A.S. Malik. and S.A. Qureshi, "Analyzing the Factors Affecting Network Lifetime for Cluster-based Wireless Sensor Networks," *Pak. J. Engg. & Appl. Sci.* 6(2010) 9-16.
- [35] N. Mazhar, "Energy Efficient Security in MANETs: A Comparison of Cryptographic and Artificial Immune Systems," *Pak. J. Engg. & Appl. Sci.* 7(2010) 71-94.
- [36] F. Perich, A. Joshi, and R. Chirkova, "Data Management for Mobile Ad-Hoc Networks," in *Enabling Technologies for Wireless E-Business*, (2006)132-176.
- [37] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, "OASIS Reference Model for Service Oriented Architecture 1.0," Committee Specification, <http://www.oasis-open.org>, (2006).
- [38] A. Karim, S. I. Jami, I. Ahmad, M. Sarwar, and Z. Uzmi, "Clustering IP Addresses Using Longest Prefix Matching and Nearest Neighbor Algorithms," in proceedings of *Pacific Rim International Conference on AI (PRICAI 2004)*, Auckland, New Zealand, (2004).