www.icgst.com

A Novel Service Providing Protocol with QoS Support over MANET's

V. Babaiyan, O. Bushehrian
Dept. of Computer Eng. and IT, Shiraz University of Technology
Shiraz, Iran
[v.babaiyan, bushehrian]@sutech.ac.ir
http://www.sutech.ac.ir

Abstract

Mobile ad-hoc networks (MANET's) consist of mobile nodes interconnected by multi-hop wireless paths. Due to the dynamicity of these networks, supporting Quality-of-Service (QoS) is a primary requirement of the applications deployed over MANET's. In this paper, a QoS aware service discovery and assignment protocol for MANET's is proposed which supports both reliability and performance. In order to maximize the reliability of services (measured by the number of hand-off's) and also serve the clients according to their performance needs, specified in the client Service Level Argument (SLA), we have used a load balancing method to select the service provider based on the current state of the network. To estimate the current state of the network a Hidden Markov Model (HMM) is applied.

Keywords: *MANET, Service Providing, HMM, Load Balancing, QoS.*

1. Introduction

A MANET is a self-organizing collection of wireless mobile nodes that form a temporary and dynamic wireless network without any infrastructure. MANETs are self-configuring; there is no central management system with configuration responsibilities. All the mobile nodes can communicate with each other directly, if they are in other's wireless radio range. MANETs allow ubiquitous service access without any fixed infrastructure and its applications such as audio/video conferencing, webcasting requires very stringent and inflexible QoS. The provision of QoS guarantees is much more challenging in MANETs than wired networks due to node mobility, limited power supply and a lack of centralized control [1].

Ad hoc network is faced with various limitations that need to be considered among design and implementation of any protocol for it. To have a fast service discovery in a network with minimum energy consumption, distributing of services information and their management play important roles [2].

Service discovery protocols in MANET make resource sharing among mobile nodes possible. Once a client requests for a specific service, this service might be provided by several providers with different qualities. Therefore, we need a mechanism to choose among the providers with some criteria. One of the important challenges in service discovery and assignment over dynamic networks such as MANET's is the successive disconnection and data management [3]. Considering the performance requirements of the clients during the service discovery and assignment procedures over MANET's is also another challenging issue. For instance, Most of the multimedia applications have stringent QoS requirements that must be satisfied during their executions.

In this paper a novel method for service discovery and assignment over MANET's is presented which supports both reliability and performance of the service delivery. The reliability is measured by the number of hand-off's(or disconnections) during the delivery of a service to a client perceived at the client side and the performance is measured in terms of the service response time required by the client. The main objective in the proposed protocol is to assign available services to the clients such that not only the amount of violations of the client response time requirements is minimal during the service delivery but also the number of hand-off's perceived at the client side becomes minimal. To achieve these goals, a load-balancing based method for service assignment is applied in this paper. Due to the dynamic nature of MANET's, to choose the best service provider upon a service request by a client it is very important to estimate the current status of the network. So we have applied a Hidden Markovian Model based method to estimate the current status of the environment from the network and service providing aspects.

The remainder of this paper is organized as follows. In Section (2), we discuss related works about providing service over MANET. In section (3), we explain the hidden Markov Model that used to estimates the system states. Purposed protocol for providing service with QoS on the MANET environment, is given is section (4). Section (5) explains a numerical example and finally we draw the conclusions and discuss our findings in the field of QoS providing in section (6).

2. Related works

In [4] Hoebeke et al. introduce two different service discovery schemes. In Directory-less and resource discovery mechanism, nodes reactively request services when needed and/or nodes proactively announce their services to others, which is an attractive approach for infrastructure less networks. The alternative scheme is the directory-based one and involves directory agents where services are registered and service requests are handled. Halonen and Ojala [5] present a novel cross-layer design for providing service oriented architecture (SOA) in a mobile ad hoc network. Here, the SOA service discovery is integrated into the ad hoc routing protocol underlying the MANET. They present a prototype implementation based on the optimized link state routing (OLSR) protocol. There is no central element in their approach and the service discovery is fully distributed on a mesh network. Since there is no central registry to collect the available service descriptions, there is no need to actually both publish and discover the services. These operations can actually be considered in parallel with ad hoc routing protocols, which are either proactive or on-demand.

Lund et al. [6] used the web services for implementing the service providing over MANET's. The main functional components of web services are the service provider, service consumer, and service registry. A service provider can publish the services it is willing to share with others in a service registry that announces their availability. A service consumer may browse the service registry to retrieve the relevant announcements that describe where and how the services may be invoked. In a service-oriented architecture, consumers and providers need not know each other's locations; they need only know where the service registry is.

Li [7], proposed a fully distributed and adaptive algorithm to provide statistical QoS guarantees with respect to accessibility of services in an ad-hoc network. In this algorithm, he has focused on the optimization of a new QoS parameter of interest, service efficiency, while keeping protocol overheads to the minimum. To achieve this goal, he first theoretically derived the lower and upper bounds of service efficiency based on a novel model for group mobility, followed by extensive simulation results to verify the effectiveness of his algorithm. Riva et al. [8] proposed a novel model of service provisioning in ad hoc networks based on the concept of context aware migratory services. Unlike a regular service that executes always on the same node, a migratory service can migrate to different nodes in the network in order to accomplish its task. The migration is triggered by changes of the operating context, and it occurs transparently to the client application. They designed and implemented a framework for developing migratory services.

Raychoudhury et al. [9] proposed a novel service discovery solution which employs service handoff to facilitate seamless service access for mobile users. In this model entire network is divided into the domains and in each domain K nodes with higher capability are elected as directory nodes and the best of them are appointed as an elector of the domain. All the service provider nodes then find the nearest directory in their domain and register their services with the directory. Service handoff is triggered if a disconnection takes place between the service provider and the user. Jayapal and Vembu [3] have proposed an adaptive service discovery protocol that enhances the performance of service discovery. Their main focus is to use an adaptive core node election mechanism that changes whenever the load increases and is also robust against network failures. This enhances the performance of discovery due to the reduction in frequent handoffs. They have used a distributed directory based service discovery mechanism that operates in a proactive mode with service advertisements to the core node and selects a provider based both on distance and service capability of the provider.

Table1. Comparison of previous studies

Study	Author(s)	QoS	Service advertising	Service discovery	Service selection
1	Hoebeke et al.	No	Hybrid	hybrid	Not mentioned
2	Halonen and Ojala	No	Directory - less	Proactive	Not mentioned
3	Lund et al.	No	Directory - based	Proactive	Not mentioned
4	Li	Yes	Directory - less	Reactive	Route specific
5	Riva et al.	Yes	Directory - based	Reactive	Route specific
6	Raychoudhury et al.	Yes	Directory - based	Reactive	hybrid
7	Jayapal and Vembu	Yes	Directory - based	Proactive	hybrid

In the directory less architecture, the mobile nodes do not distribute their service descriptions to the other nodes in the network. In contrast, in the directory based architecture, the service providers register their services with the directory nodes and the service information is provided to the requestors, through these directories. A requester can obtain service information using the reactive, proactive or hybrid service discovery modes. Service selection may either be route-specific [10] or service-specific [11].

3. Estimating the network state

In our proposed method, the service selection is performed depending on the current state of the environment. So it is very important to be able to estimate the current state of the MANET. Here we define the whole system state as a combination of the network state (infrastructure) and the service availability state (software). So the system state is defined as the pair: *(networkState, ServiceAvailiblity)*. The values of the first and second elements are taken from the set: *{'stable','dynamic'}*. Therefore at any given time, the whole system resides in one of the following four possible states: *(Stable, Stable), (Stable, Dynamic), (Dynamic, Stable), (Dynamic, Dynamic)*. To estimate the current state of the system, each node uses a HMM [12]. By using a HMM, a node is able to compute the most probable system state (out of the four states of the system) based on its local observations. Here the *'packet loss rate' and the 'service loss rate'* are observed periodically. Therefore, four independent observed states are possible as listed in Table 2.

Table2. List of possible observations

Observation Number	Packet Loss Rate	Service Loss Rate
1	Low	Low
2	Low	High
3	High	Low
4	High	High

The Hidden Markov Model is a variant of a finite state machine having a set of hidden states, Q, an output alphabet (observations), O, transition probabilities, T, output (emission) probabilities, B, and initial state probabilities, Π. The current state is not observable. Instead, each state produces an output with a certain probability (B). Usually the states, Q, and outputs, O, are understood, so an HMM is said to be a triple, (T, B, Π). Here we have used the forward-backward algorithm [13] for calculating the most probable state of the environment. To estimate the forward probabilities which provide, for all k∈ {1, 2, …, t}, the probability of ending up in any particular state given the first k observations in the sequence, i.e. $P(X_k \mid O_{1:k})$. Initial state probabilities are shown in matrix Π :

$$\Pi = [\Pi_1 \quad \Pi_2 \quad \Pi_3 \quad \Pi_4]$$

The transition probabilities $P(X_t \mid X_{t-1})$ of a given random variable X_t, representing all possible states in the hidden Markov model will be represented by the matrix T, where the row index, i represents the start state and the column index j represents the target state. T_{ij} is the probability of transforming from state i to state j. In a hidden Markov model the state is unknown, and we instead observe events associated with the possible states. In the event matrix B, element B_{ij} is the probability of observing event j given a particular state i. We can calculate $P(X_t = X_i \mid O_1, ..., O_t, \Pi)$ using the forward probabilities.

After using forward algorithm and estimating the current state of the system, a client decides which service provider to choose. List of possible actions that a client may take, upon a service request are as follows:

1. Select the fastest service provider (with the lowest response time).
2. Select the service provider with the lowest hop count.
3. Select a provider that minimizes a function of response-time and hop count
4. Wait a random amount of time and re-send the request.

When the system state is *(Dynamic, Dynamic)*, the action should be taken could be waiting for a random amount of time and re-send the request later. Because if the provider with the lowest response time is selected, it is possible that in the near future, the connection to the provider node is lost due to the unstable network condition. Furthermore, if the nearest provider is chosen, it is most likely to fail due to the unstable provider condition. When the system state is *(Dynamic, Stable)*, it is reasonable to select the service provider which is the nearest, because the network infrastructure is unstable and the network might become partitioned. Alternatively a provider which minimizes a function of response-time and hop count can be selected.

When we are in *(Stable, Dynamic)* state, it is better to choose the fastest provider from the list of available providers in which the recently selected providers are excluded. Therefore the recently failed providers are not chosen again. And finally when the system state is *(Stable, Stable)* we can select the provider that has the best performance and less distance to the client.

The preferred actions in each state are listed in Table3

Table3. Suitable action in each state

State	Action
(Dynamic, Dynamic)	4
(Dynamic, Stable)	2,3
(Stable, Dynamic)	1
(Stable, Stable)	3

4. The Proposed Protocol

In this section the proposed protocol for selecting a service provider is explained. To elect a service provider, any node in the network, called the requester, can initiate an election by sending a *"ServiceRequest"* message to its immediate neighbors (i.e., the nodes in its radio range). When a node receives this request for the first time, it designates the sender as its parent, and subsequently sends out a *"ServiceRequest"* message to all its immediate neighbors, except its parent. When a node receives a *"ServiceRequest"* message from a node other than its parent, it replies the sender with a NULL value in the *"ServiceReply"* message immediately.

Once node R has designated node Q as its parent, it forwards the *"ServiceRequest"* message to its immediate neighbors (excluding Q) and waits for their *"ServiceReply"* messages for a period of time. The *"ServiceReply"* messages contain a "capability" field that keeps information about the service provider mean response time and its resource capacities. This information will later allow a node participating in the protocol to compare its children capacities and selects the eligible node(s) as the service provider(s). Here, Q had sent a *"ServiceRequest"* message because its own parent P had done so as well. In turn, when Q eventually replies the *"ServiceRequest"* message previously sent by P, it will pass the most eligible node(s) to P as well. In this way, the requester will eventually get to know which node is best to be selected as the service provider, after that, it will send the "Request" messages to the selected node to obtain the service and the selected service provider starts serving the requester node with the *"Reply"* messages.

This process is illustrated in Figure 1. Nodes have been labeled *'a'* to *'j'*, along with their capacities. Node *'a'* broadcast *"ServiceRequest"* to its immediate neighbors. After that step, *"ServiceRequest"* messages are propagated to all nodes subsequently. When one intermediate node receives the "ServiceReply" from its children, it chooses the best provider(s) and takes it in *"ServiceReply"* message and sends it to its parent. In the end, the requester node will choose the best service and will request it to get service with *"Request"* message. Each requester node tags its *"ServiceRequest"* message with a unique identifier.

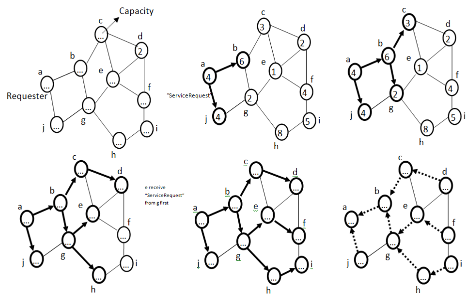

Figure1. Service Selection algorithm

The protocol is executed by a management agent installed on each node called Service Manager (SM). Each SM in the network is responsible to transmitting the following messages to other SM's or to its local client program:

"ServiceQuery": a client sends this message to its local SM to ask for a service.

"ServiceSuggestion": SM suggests the selected service provider to the local client after completing the protocol.

"ServiceRequest": Each SM transmits this kind of message to its counterparts during the execution of the protocol repeatedly.

"ServiceReply": Each SM responses its parent by this message to inform it about the selected providers.

"Request": A client program sends this message to the service provider to obtain the service iteratively.

"Reply": A service provider sends to its clients in response to every "Request" message.

The "ServiceRequest" message is defined as follows:

"ServiceRequest"*:*<PacketType, ServiceName, ServiceDescription, RequesterAddress, RequesterBroadcastID, LastAddress >

Where, *'Packet type'* keeps the type of messages, *'ServiceName'* is the requested service and *'ServiceDescription'* is information about the requested service. *'RequesterAddress'* field specifies the requester address. The *'RequesterAddress'* along with the *'RequesterBroadcastID'* field uniquely identifies each request in the system. *'LastAddress'* field is used in our protocol to determine the sender of a request in each stage. The "ServiceReply" message is defined as follows:

"ServiceReply": <PacketType, Capability, ProviderAddress, RequesterAddress, RequesterBroadcastID, DestinationAddress, HopCount >

Where, the *'Packet type'* defines the type of messages, *'Capability'* field determines performance information about the service provider, *'ProviderAddress'* field keeps the address of the service provider, *'DestinationAddress'* is the address of the next node which the response should be sent back to it, and the *'HopCount'* keeps the distance of the service provider node from the requesting node.

Each server has a certain computing capacity, which represents the largest number of requests that can serve. With the help of queuing theory [14], the mean response time of each request at each service provider can be computed. This value is kept in the *'Capability'* field in the *"ServiceReply"* message. When an intermediate node receives all the responses from its immediate children, during the expected deadline, it chooses service providers for which the response time is equal or lower than the requested response time of the requesting client. This list is then sent back to the parent successively.

The sequence of activities in the proposed protocol once a SM receives a message from its counterparts is shown in Figure 2.

Each node processes a received message as shown in Figure 3.

5. Case-Study

Here we give a simple numerical example of a hypothetical MANET to explain how our algorithm works in practice. Consider a mobile ad hoc network which is composed of 9 nodes. Assume a *"storage service"* which is provided by several providers. As mentioned in section (4), each service provider can calculate its response time. This network is depicted in Figure 4 and the response times of each node are shown on it.

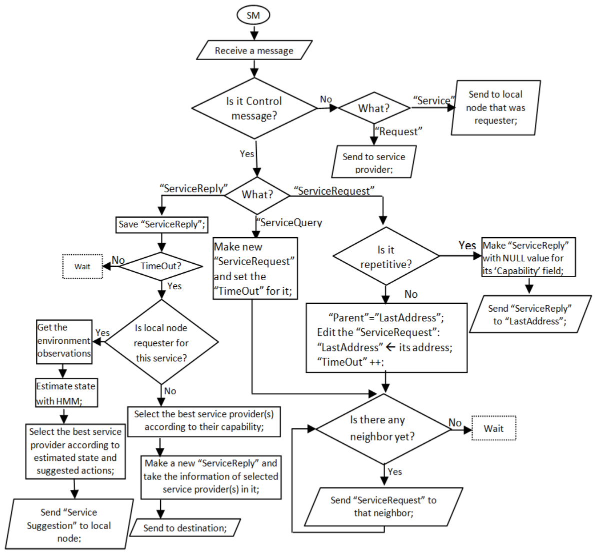

Figure2. Operational flowchart of SM

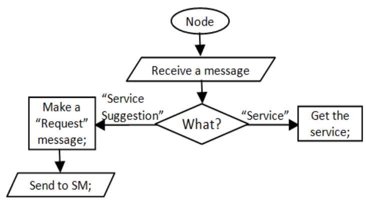

Figure3. Operational flowchart of node

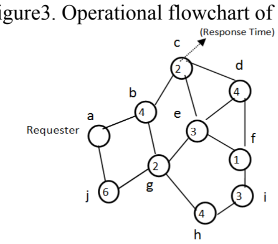

Figure4. Typical Mobile Ad-hoc Network

Assume that node *'a'* is a requester of the storage service. So it sends a "ServiceRequest" message to its immediate neighbours (*'b'* and *'j'*) as shown in Figure 5. The *"Service Request"* message fields are *<ServiceRequest,*

DataStorage, MaximumResponseTime=4, 'a', 1, 'a'>. These fields indicate that node *'a'* searches for a data storage service with maximum response time 4.

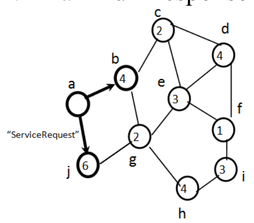

Figure5. Node *'a'* sends request to *'b'* and *'j'*

Nodes *'b'* and *'j'* receive the *"ServiceRequest"* for the first time, so they designate 'a' as their parent and edit the message by putting their address in the '*LastAddress'* field and send a *"ServiceRequest"* message to all their immediate neighbours, except their parent. This step is shown in figure 6.

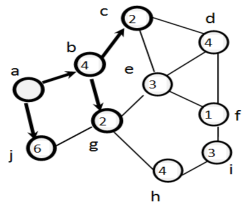

Figure6. Node '*b'* sends out the request to *'c*' and *'g'*

This process will continue as shown in Figure 7 and 8.

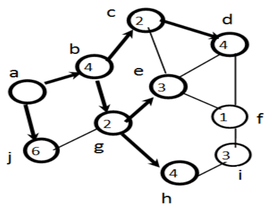

Figure7. Node *'c''* and *'g'* re-sends the request

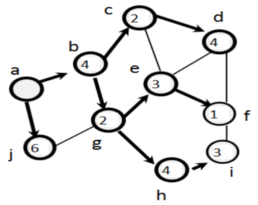

Figure8. Node *'e''* and *'h'* re-sends the request

Each node waits for certain duration of time to receive the *"ServiceReply"* from its children.

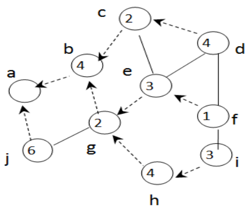

Figure9. Returning the "ServiceReply" messages to Requester

Finally the candidate servers are introduced to the requester. The necessary fields of the *"ServiceReply"* messages returned to the applicant are shown in the Table 4. Note that only those servers which their response time has been lower than the requester service demand are selected and introduced to the parent. Here*,* those servers with maximum response time lower or equal to 4 are selected.

Table4. "ServiceReply" messages that sent to requester node

sender	receiver	Maximum Response Time (<=4)	Provider	Hop Count
d	c	4	d	1
f	e	1	f	1
i	h	1	i	1
c	b	4 2	d c	2 1
e	g	3 1	e f	1 2
h	g	4 1	h i	1 2
g	b	4 3 2 1 1	h e g i f	2 2 1 3 3
b	a	4 4 4 3 2 2 1 1	d h b e c g f i	3 3 1 3 2 2 4 4

After the defined deadline, the requester node runs the Hidden Markov model algorithm and estimates the current state of the network. Then according to the estimated state and proposed actions mentioned in section (3), a provider among the candidates can be chosen.

As mentioned before, we have four possible states:

- State1: (Stable, Stable)
- State2: (Stable, Dynamic)
- State3: (Dynamic, Stable)
- State4: (Dynamic, Dynamic)

Suppose that the initial probabilities of being in each state are equal and are as follows:

$$\Pi = [0.25 \quad 0.25 \quad 0.25 \quad 0.25]$$

The transition probabilities between states are shown in matrix T:

$$T = \begin{pmatrix} 0.5 & 0.2 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.2 & 0.5 \end{pmatrix}$$

We assume the event matrix as follows:

$$B = \begin{pmatrix} 0.60 & 0.15 & 0.15 & 0.10 \\ 0.15 & 0.60 & 0.10 & 0.15 \\ 0.15 & 0.10 & 0.60 & 0.15 \\ 0.10 & 0.15 & 0.15 & 0.60 \end{pmatrix}$$

Four observations are assumed as listed in Table 5. For each observation the state probability vector is calculated.

Table5. Result of HMM

Observation: (PacketLossRate, ServiceLossRate)	**State probabilities**	**Most likely state**
(Low, Low)	[0.6, 0.15, 0.15, 0.10]	State1
(Low, High)	[0.15, 0.6, 0.10, 0.15]	State 2
(High, Low)	[0.15, 0.10, 0.6, 0.15]	State 3
(High, High)	[0.10, 0.15, 0.15, 0.6]	State 4

According to the proposed actions in Table 3 and the candidate *"ServiceReply"* messages received by the applicant, the best service provider can be selected.

If we are in *"State 1"*, action *(3)* should be taken. So, nodes *'c'* or *'g'* are better than others from both response time and the hop-count viewpoints. If we are in *"State 2"*, action *(1)* should be taken, hence nodes *'f'* or *'i'* are selected. If we are in *"State 3"* action *(2)* is taken. Hence, node 'b' is the best choice, due to its minimal distance to the requester. And finally if we are in *"State 4"* it is proposed to wait and send the request again which is action *(4)*.

6. Conclusion

In this paper, we have proposed a novel service providing protocol which applies the Hidden Markov Model to estimate the network state based on the local observations. We also have proposed the actions to take at each network state by the clients. Our protocol selects service providers at each network state such that the minimal amount of hand-off is resulted, so a reliable service is provided to the clients.

7. References

[1] G. Santhi, and A. Nachiappan, A SURVEY OF QOS ROUTING PROTOCOLS FOR MOBILE AD HOC NETWORKS, International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010

[2] A. Hosseini-Send, B. Pahlevanzadeh, T.C. Wan, R. Budiarto, and M.M. Kadhum, A DISTRIBUTED RESOURCE-MANAGEMENT APPROACH IN MANETS, Journal of Engineering Science and Technology Vol. 4, No. 2, pp. 142 – 153, 2009

[3] C. Jayapal, and S. Vembu, Adaptive Service Discovery Protocol for Mobile Ad Hoc Networks, European Journal of Scientific Research ISSN 1450-216X Vol.49 No.1, pp.6-17, 2011

[4] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester, An Overview of Mobile Ad Hoc Networks: Applications and Challenges, published in The Journal of The Communications Network, Vol. 3, Issue 3, 3th quarter 2004

[5] T. Halonen, and T. Ojala, Cross-layer design for providing service oriented architecture in a mobile Ad Hoc network, In Proc. Of MUM '06, vol. 193. ACM, New York, NY, 2006.

[6] K. Lund, A. Eggen, D. Hadzic, T. Hafsøe, and F. T. Johnsen, Using Web Services to Realize Service Oriented Architecture in Military Communication Networks, IEEE Communications Magazine, Vol. 45, No. 10, pp. 47-53, October 2007

[7] B. Li, QoS-aware Adaptive Services in Mobile Ad-hoc Networks, in Proceedings of the Nineth IEEE International Workshop on Quality of Service (IWQoS 01), Karlsruhe, Germany, pp. 251–268, June 2001

[8] O. Riva, T. Nadeem, C. Borcea, and L. Iftode," Context-Aware Migratory Services in Ad Hoc Networks" IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 6, NO. 12, December 2007

[9] V. Raychoudhury, J. Cao, W. Wu and C. Chen, Service Handoff for Reliable and Continuous Service Access in MANET, 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing, pg. 172-179, February 2011

[10] Alex Varshavsky, Bradley Reid and Eyal de Lara, 2005. A cross layer approach to service discovery and selection in MANETs, IEEE International Conference on Mobile Ad hoc and Sensor Systems, pp.466.

[11] Vincent Lenders, Martin May and Berhard Plattner, 2005. Service discovery in mobile ad hoc networks: A field theoretic approach, Elsevier Journal on Pervasive and Mobile Computing 1, pp. 343-370.

[12] Hidden Markov Models, Nikolai Shokhirev, available at http://www.shokhirev.com/nikolai/abc/alg/hmm/hmm.html

[13] Forward–backward algorithm From Wikipedia, the free encyclopedia, available at http://en.wikipedia.org/wiki/Forward%E2%80%93backward_algorithm

[14] Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modelling, by Raj Jain Wiley Computer Publishing, John Wiley & Sons, Inc. ISBN: 0471503363 Pub Date: 05/01/91

Biographies

Author One received her B.Sc. from Birjand university at Birjand, IRAN and Currently She is a M.Sc student at Shiraz University of Thecnology under the supervision of Dr Bushehrian. Her research interests include Mobile Ad-hoc Network and providing QOS-aware service in MANET. Email address: V.babaiyan@sutech.ac.ir

Author Two is currently Professor at Shiraz university of Technology, IRAN. He is a Member of the IEEE. His main research interest is Software Performance Engineering and Quality of Service. Email address: Bushehrian@sutech.ac.ir