Mobility Management Using Virtual Multi-parent Tree in Infrastructure Incorporated Mobile Ad Hoc Networks*

Trung-Dinh Han and Hoon Oh**

School of Computer Engineering and Information Technology, University of Ulsan, (680-749) San 29, Mugeo 2-Dong, Nam-gu, Ulsan, South Korea
trungdinhvn@yahoo.com, hoonoh@ulsan.ac.kr

Abstract. Mobile nodes in the integrated network of infrastructure networks and mobile ad hoc networks need an efficient registration mechanism since they register with Internet gateway via wireless multiple hops. Tree-based mobility management is highly dependable in that mobile nodes register with Internet gateway along tree paths from itself up to a gateway without using flooding. In this paper, we introduce a new method using a virtual multi-parent tree in which a node maintains multiple parents whenever possible. A mobile node can still retain its upstream path to a gateway even though it loses connectivity to one of its parents. The new scheme is evaluated against the tree-based method and the other traditional methods - proactive, reactive, and hybrid by resorting to simulation. The result shows that the new method outperforms the other ones and also is very robust against the change of topology.

Keywords: Mobile IP, Mobility Management, Multiple Wireless Hops, Multi-Parent Tree, Registration.

1 Introduction

Wireless mobile nodes can communicate with each other via wireless multiple hops by building a mobile ad hoc network (MANET) without the help of infrastructure base stations. However, they often need to have an access to the Internet or communicate with mobile nodes (MN) in other MANETS. In this case, a mobile node has to register with Internet gateway and remain registered so that other nodes can find easily where it is located; this registration activity causes a lot of control overhead since it is done via wireless multiple hops unlike the original Mobile IP or Cellular IP that only manages the mobility of nodes with a wireless one hop [1]. Therefore, the Mobile IP that deals with mobile nodes in MANETs requires an efficient registration or mobility management scheme.

Recently, a lot of researches have been conducted regarding mobility management; however, the existing schemes suffer from a high mobility management

* This work was supported by 2008 Research Fund of University of Ulsan.
** Corresponding author.

C.S. Hong et al. (Eds.): APNOMS 2009, LNCS 5787, pp. 82–91, 2009.

overhead due to control messages flooded over the networks, curtailing effective bandwidth. Broch et al. [8] assume that IG is equipped with two network interface cards, where one that executes the standard IP mechanism acts as a bridge to Internet nodes while another does as a bridge to mobile nodes; this protocol uses DSR protocol to discover IG that rely on flooding. Jonsson et al. [4] focus on when to register with a foreign agent using MIPMANET Cell Switching algorithm in order to reduce control overhead. In [12], Sun et al. discuss how Mobile IP and AODV can cooperate to discover multi-hop paths between mobile nodes and IGs. Prashant et al. [11] propose a hybrid mobility management scheme that combines the techniques such as agent advertisements, TTL scoping and caching of agent advertisements, eavesdropping, and agent solicitation. In [5], Ammari et al. propose a three-layer approach in which a mobile gateway is used as an interface between mobile nodes and IG that executes Mobile IP and DSDV [3]. Some other studies on gateway discovery were conducted based on AODV [6, 10]; however, they still use flooding.

The traditional approaches for node registration are in general categorized into three types: Proactive scheme [6], reactive scheme [6, 8, 11], and hybrid scheme [6, 11]. In the proactive scheme (Proactive), an IG floods advertisement message periodically to inform its presence to the mobile nodes that are visiting its coverage area. Upon receiving the advertisement message, the mobile nodes set up the forward paths to IG and can register with the IG that has issued the advertisement message along the paths. Since this method allows all mobile nodes to register with a gateway regardless of whether they participate in active communication or not, remote nodes can make connection request to the registered mobile nodes all the time. Nevertheless, the repeated broadcasting of IG advertisements and solicitations can have a negative impact on the MANET due to excessive flooding overhead. In the reactive scheme (Reactive), only the mobile node that wants to participate in communication floods IG solicitation message to explore an IG, setting up a reverse path to IG. This approach also suffers from the flooding of IG solicitation message. The hybrid approach (Hybrid) was proposed to complement the shortcomings of two approaches, where connectivity is maintained in such a way that IG floods advertisements, but only within a limited number of hops, say k-hop, from the IG while the mobile nodes residing outside k-hop range flood IG solicitations only if they want to communicate with any mobile nodes. This approach still uses flooding for registration and solicitation.

Meanwhile, the tree-based approach (Tree-Based) does not use an inefficient flooding for registration, and instead, trees are formed and maintained such that every node in a tree keeps track of information of all its descendent nodes. However, it has to find a new parent if it loses connection to its parent, requiring another join process. The new approach proposed in this paper is basically based on the tree-based approach [7] where every node registers with the gateway along tree paths; however, it is different in that a node maintains multiple parents. Thus, each node in a tree maintains multiple paths virtually to gateway. That is, a mobile node can still retain its upstream path to a gateway even though it

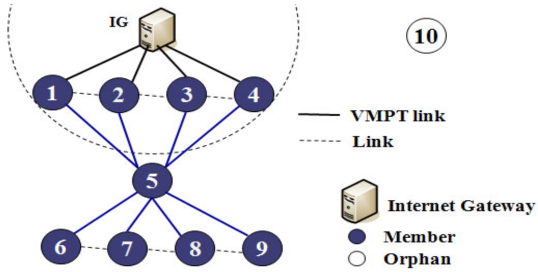

Fig. 1. Network Model

loses connectivity to one of its parents without going through a tree maintenance process. The proposed method is evaluated against the tree-based approach and the traditional mobility management schemes for different network scenarios in terms of control overhead, registration ratio, registration latency and registration jitter. The simulation results show that the new approach greatly outperform the previous approaches and is highly robust against the variations of network scenarios.

The remainder of the paper is organized as follows. Section 2 describes network model with useful definitions. The formal description of a new mobility management method is given in Section 3. In Section 4, we evaluate different methods by resorting to simulation and finally, we make concluding remarks in Section 5.

2 Preliminary

2.1 Network Model

Mobile nodes (MN) form the multiple trees, each of which stems from an Internet gateway (IG) being a root as in Fig. 1 in which the solid lines indicate Virtual Multi-Parent Tree (VMPT) links and the dashed lines do links that do not belong to the trees. A node is said to be a *member* if it belongs to any tree, and is said to be an *orphan* if it does not have any parent. We define a *primary parent* as a node that an orphan node joins for the first time. An IG acts as both home agent (HA) and foreign agent (FA). We assume that a node can overhear packets that are transferred among the other nodes [9]. We further assume that all nodes have to be ready to receive communication request from other distant nodes regardless of the schemes, proactive, reactive or hybrid.

2.2 Message Definitions

We define some control messages used to build VMPT. We denote the type of control message as CT.

- IG-Hello = (CT), IG broadcasts this message periodically.
- MN-Hello = (CT, HopToIG), MN sends this message to its primary parent periodically.
- J-REQ = (CT), MN sends this message to its primary parent to join.
- CR-REQ = (CT), an orphan responds with this message to its child which sends MN-Hello.

2.3 VMPT Information Structure (VTIS)

2.3.1 VTIS Definition

We define some notations for convenience with respect to node i as follows. *i.IG* indicates IG to which node i belongs. *i.NS* indicates a set of neighbors of node i. *i.P* and *i.C* denotes a set of parents and children of node i, respectively. *d(i, j)* stands for the distance in hops from i to j; if j is IG, it is simply referred to as *HopToIG*.

Definition 1: VTIS(i) = (i.P, i.C, d(i, i.IG)) where $i.P = \{x |\ x \in i.NS,\ d(i, i.IG) = d(x, x.IG) + 1\}$, $i.C = \{x |\ x \in i.NS,\ d(i, i.IG) = d(x, x.IG) - 1\}$.

2.3.2 VTIS Maintenance

A node can get a child when it either receives J-REQ or overhears MN-Hello from a node which has HopToIG greater than its own one by one. A node can get a parent when it either receives ACK to J-REQ or overhears MN-Hello from a node which has HopToIG less than its own one by one.

Every member in a VMPT sends MN-Hello periodically to its primary parent. The parents update their VTIS whenever they receive or overhear MN-Hello from their children. If a node does not receive or overhear MN-Hello from its child for the specified interval, it deletes the child. If a node fails to send MN-Hello to its primary parent, it selects another parent to send MN-Hello and if it succeeds, it changed its primary parent. In case that there does not exist a parent to select, it becomes an orphan node. Thus, every node in a VMPT can maintain its VTIS as indicated in Definition 1.

3 VMPT Mobility Management

3.1 VMPT Establishment

3.1.1 MN-IG Join Process

Let us consider the procedure that MN joins IG. If a MN i receives IG-Hello from an IG and $d(i, i.IG) > 1$, it sends J-REQ to join. If it receives ACK from the IG, it becomes a child of the IG. If an IG receives J-REQ, it takes the sender as its child.

3.1.2 MN-MN Join Process

Consider the procedure that a MN joins a MN being a member of a VMPT. When an orphan overhears a MN-Hello from a neighbor member, it sends J-REQ to the neighbor to join. Upon receiving ACK, it becomes a child of the neighbor. A member that receives J-REQ takes the sender as its child.

3.2 VMPT Maintenance

3.2.1 VMPT Maintenance Process

A mobile member node *unicasts* MN-Hello to its primary parent while an IG *broadcasts* IG-Hello periodically. By overhearing MN-Hello, a node can update its neighbors. We describe how VMPT links are maintained dynamically below.

A. Children Maintenance: If a parent either receives or overhears MN-Hello from its child, it updates its children. If the parent overhears MN-Hello from its neighbor that has HopToIG less than its own HopToIG by one, it takes the node as a new child. If it does not receive MN-Hello from a certain child for the specified interval, it deletes the child.

B. Multi-Parent Maintenance: A node that overhears MN-Hello from a neighbor which has HopToIG equal to the HopToIG of its primary parent takes the node as one of its parents. If a node does not overhear MN-Hello from one of its parents for the specified interval, it deletes the parent.

3.2.2 Parent-Change Process

Suppose that a node x overhears MN-Hello or receives IG-Hello from its neighbor y. If $d(x, x.IG) > d(y, y.IG) + 1$, x joins y by initiating the MN-MN Join Process or following the MN-IG Join Process. If it changes its parent, it updates its VTIS.

3.2.3 Children-Release Process

If an orphan node receives MN-Hello from its children, it replies with CR-REQ to ask them to leave. The child that either fails to send MN-Hello or receives CR-REQ will select another parent as a primary parent from its VTIS to send MN-Hello. In case that there does not exist a parent to select, it becomes an orphan and then selects a neighbor member y that gives the minimum $d(y, y.IG)$ among its neighbor list and initiates the MN-MN Join Process or IG-MN Join Process.

4 Performance Evaluation

Three scenarios S1, S2, and S3 are used according to IG locations (center, top center, and corner). If IG is located in the center, mobile nodes will have a shorter average distance to the IG, resulting in a decreased control overhead. In the other two cases, nodes will have a relatively higher distance, increasing control overhead. We use four metrics *control overhead*, *registration ratio*, *registration latency* and *registration jitter* to evaluate the different mobility management schemes which are *Reactive*, *Proactive*, *Hybrid*, *Tree-Based*, and *VMPT*.

We performed comparison in the QualNet version 3.9 by changing IG location (at center (S1), top center (s2), and corner (s3)), varying number of nodes (*nNodes*) as 50, 75, and 100 or maximum speed (*mSpeed*) as 0, 10, 20, 30, 40, and 50 (m/s) in the terrain of 1000 x 1000(m^2). Figure 2 shows an S1 with a different number of nodes. The used parameter values are given in Table 1.

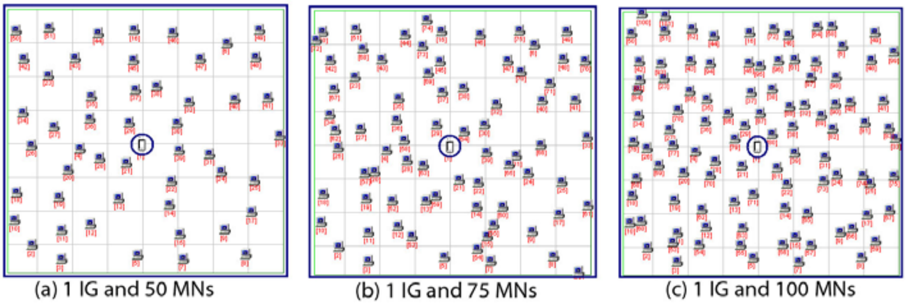

Fig. 2. Three deployment scenarios (A circled one indicates IG)

4.1 Performance Evaluation of Mobility Management Schemes

In this section, we study performance of the Tree-Based and the traditional mobility management methods. Fig. 3 - Fig. 6 compare Tree-Based approach against the traditional registration methods - Reactive, Proactive, and Hybrid. According to Fig. 3, Hybrid always generates more overhead than the simple Proactive unlike we have expected. One reason is that it is very hard to set a right value of k in Hybrid. Another reason is that Hybrid and Reactive use a ring-expanding search algorithm to find a IG or some registered node and in this process, if it fails to find any, it increases TTL by 2 and then reinitiate a search. On the contrary, Tree-Based shows the smallest overhead of all even though it pays some cost to maintain trees. This is because Tree-Based does not use a flooding in maintaining trees and managing node mobility.

Referring to Fig. 4, Proactive shows the lowest registration ratio of all. This is because when a node receives advertizement message, it waits until the registration interval timer expires, leading to the change of topology that frequently

Table 1. Simulation parameters and values

Parameters	Value
Mobility Pattern	Random Waypoint
Pause Time	30
Number of Nodes	Varying density (1 fixed IG)
Dimension	1000 x 1000
Transmission Range	250 m
Wireless Bandwidth	2 Mbps
Registration Interval	5 seconds
Advertisement Interval	2 seconds
Advertisement zone (IG)	2 hops
Increment in the expanding ring search (MN)	2 hops
Simulation Time	600 seconds

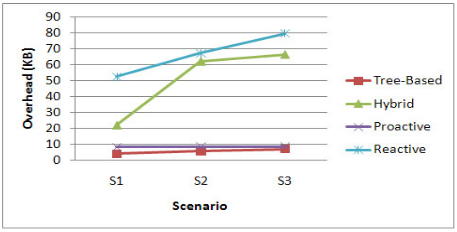

Fig. 3. Control overhead for different scenarios (nNodes = 50, mSpeed = 10 m/s)

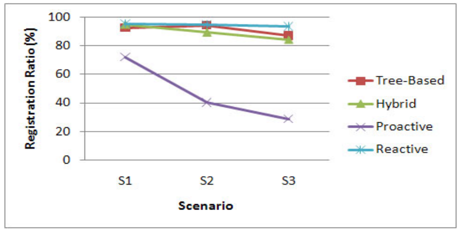

Fig. 4. Registration ratio for different scenarios (nNodes = 50, mSpeed = 10 m/s)

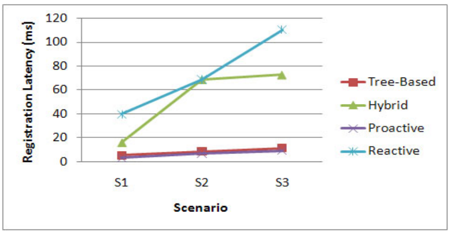

Fig. 5. Registration latency for different scenarios (nNodes = 50, mSpeed = 10 m/s)

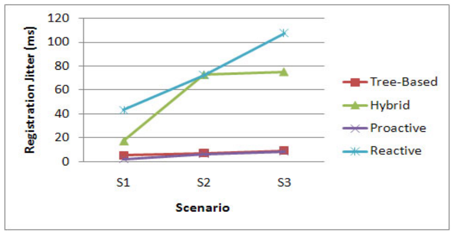

Fig. 6. Registration jitter for different scenarios (nNodes = 50, mSpeed = 10 m/s)

invalidates the established routes. This is even severe as the average distance of a node to IG becomes longer as in S2 and S3. Note that S3 has longer distance to IG on average than S2 since the former has the IG located at the corner.

In Fig. 5, Proactive shows the lowest registration latency since it does not perform any route discovery and it just sends registration message if the route is valid. In case of Reactive, it discovers the route to IG if the route is not valid and then sends registration message, causing the highest registration rate and the highest registration latency. Fig. 6 shows the curve pattern of registration jitter similar to the curves of registration delay.

Proactive is not recommended overall since it shows the lowest registration ratio. So, we may say that Hybrid is one of dependable solutions among the traditional approaches. However, the Tree-Based is always very stable over the variations of IG location scenarios. According to simulation results, it is conjectured that the message flooding affects very negatively network performance.

4.2 Performance Evaluation of VMPT

We found out that performance of theTree-Based scheme was highly dependable over the traditional schemes according to simulation study. Now, we evaluate the effectiveness of the newly proposed method, VMPT against the Tree-Based.

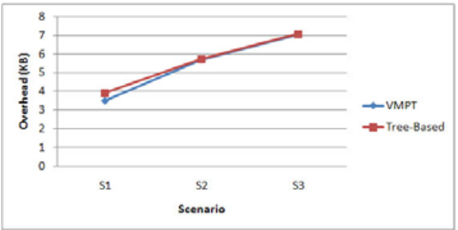

Fig. 7. Control overhead for different scenarios (nDodes = 50, mSpeed = 10 m/s)

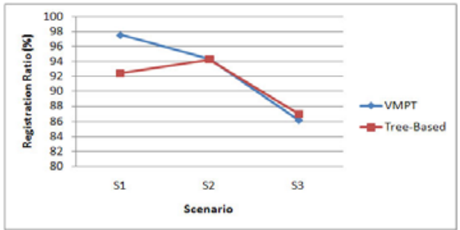

Fig. 8. Registration ratio for different scenarios (nNodes = 50, mSpeed = 10 m/s)

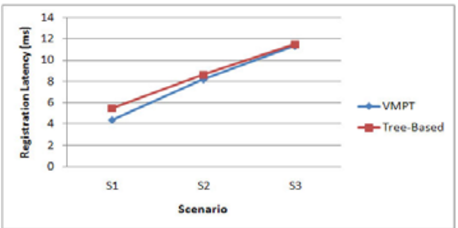

Fig. 9. Registration latency for different scenarios (nNodes = 50, mSpeed = 10 m/s)

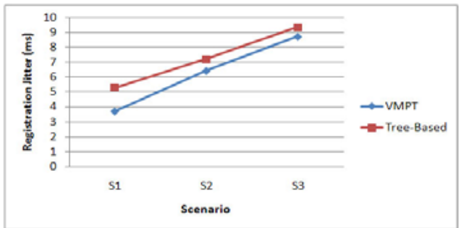

Fig. 10. Registration jitter for different scenarios (nNodes = 50, mSpeed = 10 m/s)

In Fig. 7 Fig. 10 show a comparision between VMPT and Tree-Based schemes for different scenarios. Referring to Fig 7, VMPT and Tree-Based have almost the same overhead. However, it is worth noting that VMPT has more parents in S1 rather than in S2 or S3 since tree depths in S1 are smaller. Therefore, VMPT has slightly lower overhead in S1 because it maintains more parents and thus it initiates Join Process less frequently which involves J-REQ message. Consequently, the two graphs in Fig. 8 show a relatively big gap in S1. Referring to Fig. 9 and Fig. 10, VMPT shows a slight improvement over Tree-Based for the same reason that is explained by multiple parents.

Again VMPT shows a highly stable performance for the variations of the other parameters as shown in Fig. 11 to Fig. 14. Paths from MN to IG In the

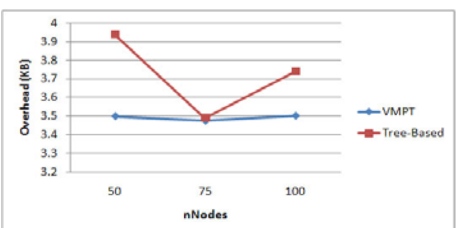

Fig. 11. Control overhead with varying nNodes (S1, mSpeed = 10 m/s)

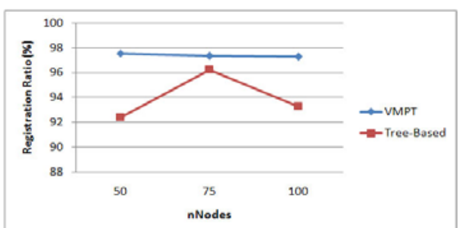

Fig. 12. Registration ratio with varying nNodes (S1, mSpeed = 10 m/s)

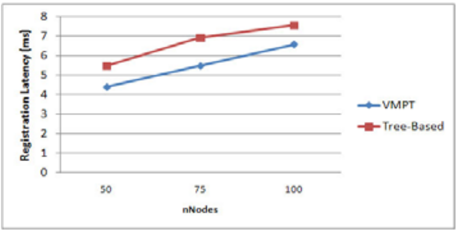

Fig. 13. Registration latency with varying nNodes (S1, mSpeed = 10 m/s)

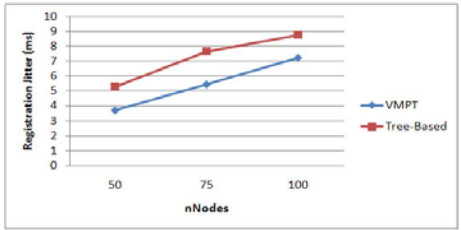

Fig. 14. Registration jitter with varying nNodes (S1, mSpeed = 10 m/s)

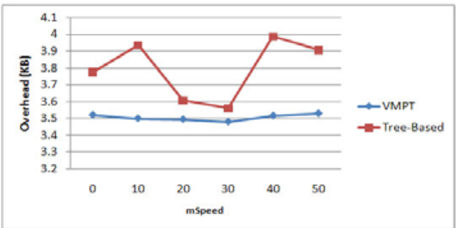

Fig. 15. Control overhead with varying mSpeed (S1, nNodes = 50)

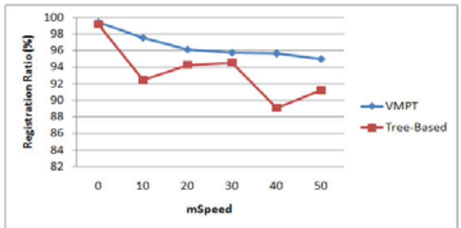

Fig. 16. Registration latency with varying mSpeed (S1, nNodes = 50)

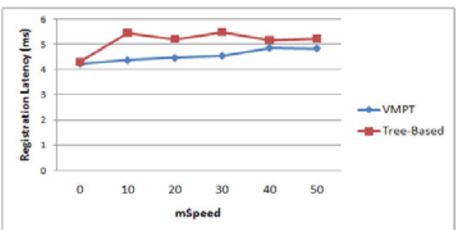

Fig. 17. Registration latency with varying mSpeed (S1, nNodes = 50)

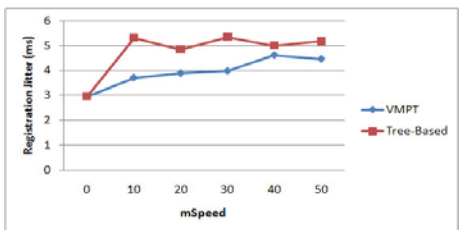

Fig. 18. Registration jitter with varying mSpeed (S1, nNodes = 50)

sparse network are more likely to be corrupted and may have the longer average length. Nevertheless, VMPT remains stable over the wide range of *nNodes* since it maintains multiple parents and thus can easily adapt to the change of network topology. Latency and jitter increase slightly for the two approaches; however, the gap between two graphs indicates that VMPT is more stable than Tree-Based.

Fig. 15 Fig. 18 show proformace curves over the change of maximum node speed from 0 to 50 m/s. VMPT that enables a quick repair of broken tree paths based on multiple parents shows a very robust performance over the Tree-Based against the high mobility of nodes. On the contrary, the Tree-Based becomes unstable since it pays more cost to handle the broken paths as node mobility increases.

5 Conclusions

In this paper, we proposed a new VMPT mobility management method which maintains multiple parents whenever possible. We compared the new method with Tree-Based and the traditional approaches - Proactive, Reactive and Hybrid. The simulation results show that the VMPT method outperforms the other ones and also are very robust against the change of IG location scenarios and the change of the parameter values that characterize the considered network.

References

1. Perkins, C.: IP Mobility Support, Request For Comments(Standard) 2002, Internet Engineering Task Force (October 1996)
2. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Second IEEE Workshop on Mobile Computing Systems and Applications, February 1999, pp. 90–100 (1999)
3. Perkins, C.E., Bhagwat, P.: Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. ACM SIGCOMM: Computer Communications Review 24(4), 234–244 (1994)
4. Jonsson, D., Alriksson, F., Larsson, T., Johansson, P., Maguire, J.G.: MIPMANET - Mobile IP for mobile ad hoc networks. In: Proceedings of IEEE/ACM Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc 2000), August 2000, pp. 75–85 (2000)
5. Ammari, H., El-Rewini, H.: Integration of mobile ad hoc networks and the Internet using mobile gateways. In: Proceedings of 18th Parallel and Distributed Processing Symposium 2004, April 2004, pp. 218–225 (2004)
6. El-Moshrify, H., Mangoud, M.A., Rizk, M.: Gateway discovery in ad hoc ondemand distance vector (AODV) routing for Internet connectivity. In: Radio Science Conference, NRSC 2007, March 2007, vol. 1-8 (2007)
7. Oh, H.: A tree-based approach for the Internet connectivity of mobile ad hoc networks. Journal of Communications and Networks 11(3), 261–270 (2009)
8. Broch, J., Maltz, D.A., Johnson, D.B.: Supporting hierarchy and heterogeneous interfaces in multi-hop wireless ad hoc networks. In: Proceedings of International Symposium on Parallel Architecture, Algorithms, and Networks, June 1999, pp. 370–375 (1999)
9. Jubin, J., Tornow, J.D.: The DARPA packet radio network protocols. Proceedings of the IEEE 75(1), 21–32 (1987)
10. Ruiz, P., Gomez-Skarmeta, A.: Enhanced Internet connectivity for hybrid ad hoc networks through adaptive gateway discovery. In: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004), November 2004, pp. 370–377 (2004)
11. Prashant, R., Robin, K.: A hybrid approach to Internet connectivity for mobile ad hoc networks. In: Proceeding of IEEE WCNC 2003, March 2003, vol. 3, pp. 1522–1527 (2003)
12. Sun, Y., Belding-Royer, E.M., Perkins, C.E.: Internet connectivity for ad hoc mobile networks. International Journal of Wireless Information Networks special issue on Mobile Ad Hoc Networks (MANETs): Standards, Research, and Applications 9(2), 75–88 (2002)