TLSR: A Tree Link State Routing for Infrastructure-Based Mobile Ad Hoc Networks*

Chi Trung Ngo and Hoon Oh**

School of Electrical Engineering,
University of Ulsan,
Ulsan Metropolitan City, South of Korea
chitrung218@gmail.com, hoonoh@ulsan.ac.kr

Abstract. The existing protocols for the Internet connectivity of mobile ad hoc networks suffer from high overhead since they often rely on flooding in mobility management and/or route discovery. In our approach, mobility management of mobile nodes is achieved in the construction process of tree topology and also contributes to building topology information for a routing protocol, resolving the inherent problem of the excessive control overhead in link state routing protocols. The activities of the routing protocol help updating the topology information. In this way, mobility management and routing protocol collaborate to increase convergence speed of topology and reduce control overhead. In addition, a message aggregation technique is used to reduce the congestion of mobile nodes near the Internet Gateway that process much more control messages. Simulation results show that the proposed method far outperforms AODV-Hybrid and OLSR+.

Keywords: Internet Connectivity, Mobility management, Mobile ad hoc networks, Aggregation, Link state routing, Tree topology.

1 Introduction

The provision of Internet connection is an essential service for extending the usability of mobile ad hoc networks (MANETs). This service can be implemented by integrating an Internet Gateway into MANETs, referred to as Infrastructure-based mobile ad hoc networks (abbreviated to IFMANET). In IFMANET, one mandatory requirement is that IGs have to manage the mobility of the MNs via multi-hop to provide uninterrupted connections for them [1] while they moves freely, thereby causing high control overhead. Furthermore, the routing protocols for IFMANETs can be designed with a new approach to take advantage of powerful IGs.

---

[*] This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1 A2013396).

[**] Corresponding author.

I. Zelinka et al. (eds.), *AETA 2013: Recent Advances in Electrical Engineering and Related Sciences*, Lecture Notes in Electrical Engineering 282,
DOI: 10.1007/978-3-642-41968-3_18,

A lot of researches have been conducted for the provision of Internet connectivity for MANETs. In [1], Sun et al. discussed how Mobile IP and AODV [2] can cooperate to discover multi-hop paths between MNs and IGs. Ratanchandani et al. [3] proposed a hybrid scheme that combines some techniques, such as agent advertisements, TTL scoping, and the detection of agent advertisements, eavesdropping, and agent solicitation. M. Benzaid et al. [4] proposed a hierarchical architecture that uses a proactive scheme to integrate Mobile IP and OLSR [5] routing protocol. In this approach, control overhead can be reduced by exploiting Multipoint Relays (MPRs) technique to limit the number of retransmissions of topology control as well as advertisement messages. Recently, some adaptive gateway discovery algorithms have been considered in IFMANETs. A. J. Yuste et al. [6] proposed an adaptive scheme in which IGs dynamically adjust the interval between two consecutive advertisement messages based on a genetic algorithm to decrease network congestion and achieve lower end-to-end delay.

The aforementioned approaches use one of the ad hoc routing protocols such as AODV, DSR, OLSR, and DSDV for routing while focusing on the invention of an efficient mobility management protocol. These protocols suffer from the overhead of flooding or the low convergence speed of topology information. Furthermore, if both mobility management and routing use flooding, the performance will be worsen due to the double impact of control overhead caused by two sources of flooding. On the other hand, a tree-based approach was proposed [7], in which a number of small trees are managed to reduce tree maintenance cost and every MN in the tree maintains its own tree information such as its descendants, its parent, and its IG. This approach can reduce the control overhead of mobility management; however, the improvement is restricted since it still used flooding, although limited, for route discovery.

In this paper, we propose a tree-based link state routing protocol in which mobility management utilizes tree topology. During mobility management, a gateway can build and update tree topology by having every node send a registration message including its link states to the gateway periodically. The constructed topology information is used later when a routing function explores a routing path. Thus, the proposed approach does not pay additional cost in building the topology information for routing. More importantly, the funneling effect [8] that occurs because the nodes near the gateway process much more messages, can be alleviated significantly by employing a message aggregation technique. According to the previous study [9], message aggregation not only reduces collision, but also improves network throughput.

In what follows, Section 2 gives network model with message and graph definitions. Section 3 details a tree link state routing protocol with a message aggregation technique. Section 4 evaluates our approach and Section 5 concludes the paper.

2 Preliminary

2.1 The Network Model

The network considered in this paper consists of stationary Internet Gateways (IGs) and a number of mobile nodes (MNs). We assume that IGs share their network management information through a high speed wired network. Every MN can act as a router to receive and forward data to other MNs. An IG is equipped with two network interface cards so that it can communicate with MNs using one interface card and also communicate with a wired host in the Internet using the remaining one. Both IG and MN have the same and limited wireless transmission range to reduce interference. An MN can not only initiate communication with any wired host or MN but also be requested for connection from wired hosts in the Internet or MNs in other wireless networks via an IG. Thus, MNs should do their best to register with the IG so that the IG can keep track of the locations of MNs. A connection between two MNs is said to be a link while a connection between an MN and its parent is to be a *tree-link*. A node is a *tree-node* if it belongs to a tree; otherwise, it is an *orphan-node*. The network forms a tree topology as shown in Fig. 1.

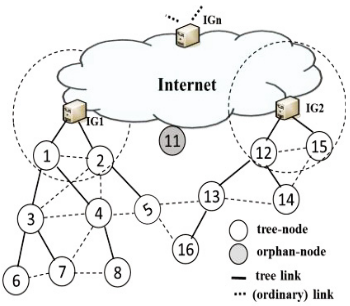

Fig. 1. The network model

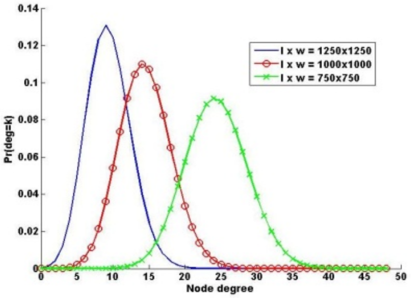

Fig. 2. The distribution of node degree with different network dimensions, $R = 250$ m, and $n = 100$ nodes

2.2 Messages and Notations

We define some notations as follows:

-*HopToIG*: indicates the distance in hops from a node to its IG.
-*NS(i)*: is a set of neighbors of node *i*.
-*C(i)*: is a set of children of node *i*.
-*TS(i)*: is a set of tree-nodes in a given tree whose root is node *i*.
-*P(i)*: is a parent of node *i*.

Every node *i* maintains its topology graph, TG(*i*), defined as follow.

$TG(i) = \{V(i), E(i), V(i) = \{x, y \mid x \in TS(i), y \in NS(x)\}$, and $E(i) = \{(x, y) \mid x \in TS(i), y \in NS(x)\}$. For example, in Fig. 1, TS(4) = {4, 8} and TG(4) = {(1, 2, 3, 5, 7, 8), {(4, 1), (4, 2), (4, 3), (4, 5), (4, 7), (4, 8), (8, 7)}. We assume that IGs can share their topology graphs through a high speed network.

In addition we elucidate some messages for convenience as follows:

- *IG-ADV* = (*MsgId*, W_1, *a*): An IG broadcasts this advertisement message to advertise its presence *periodically*, where *MsgId* distinguishes this message from other messages, W_1 and *a* are the values used in a time generation function for skewed time synchronization.
- *J-REQ* = (*MsgId*, *HopToIG*, W_1, *a*): An MN sends this join request message to join an IG or an MN as a primary parent
- *REG* = (*MsgId*, *NS(x)*): An MN *x* includes its link state, *NS*(*x*), in this message.

3 Tree Link State Routing (TLSR)

3.1 Topology Management

3.1.1 Tree Construction

A tree construction is initiated by IG-ADV issued periodically by an IG. An MN receiving this message tries to join the IG by replying with J-REQ. Upon receiving the J-REQ, the IG takes the MN as its child and responds with ACK. Then, the MN becomes a tree-node as a child of the IG when it receives ACK. Similarly, another nearby orphan-node can overhear the J-REQ and thus can join the tree-node that issued the J-REQ previously. This join process continues. In this process, every MN *i* can maintain its *NS(i)* by receiving or overhearing J-REQ. In addition, an MN can maintain its *C(i)* upon receiving J-REQ or REG message from a node. The process of receiving REG is discussed in the next following sections.

A node includes a current time in its J-REQ as a timestamp. If an MN has overheard J-REQs from multiple tree-nodes, it selects a tree-node that provides the shortest distance to an IG as a primary parent. The ties are broken in the order of the timestamps. The timestamp contributes to increasing the degree of message aggregation by the principle “the early actor gets more children.” The rest of the tree-nodes with the same distance become auxiliary parents.

An MN takes an auxiliary parent with the second earliest timestamp as a primary parent if it loses its primary parent. If it does not have any auxiliary parent, it takes a neighbor that has the shortest hop distance to an IG from its neighbor list for joining. If it does not have a tree-node to join, it becomes an orphan node until the next tree construction cycle. This will not cause any serious problem, except that it loses one registration message. This way of tree construction may cause power consumption biased to a certain node; however, power consumption after next tree construction can be biased to different nodes since nodes issue J-REQ in a random order.

3.1.2 Skewed Time Synchronization

Time at which tree-nodes initiate their registration messages needs to be synchronized in a skewed manner throughout the nodes in a tree structure so that every node issues its registration message earlier than its parent. In this way, a node can maximize the degree of aggregation by performing aggregation after receiving the registration messages from all its children and this way of harmonious transmission can also reduce message collision.

Let T_{shd} be the worst case time delay for a message transmission (J-REQ and REG message) that includes transmission delay, propagation delay, processing delay, and staggering delay or small random delay used to avoid collisions between different transmissions. We denote T_{join} and T_{reg} as an amount of time for a node to finish joining its new parent and to send an REG message to its parent in the worst case, respectively

(a) Wait time generation function

During the tree construction process, MNs have to generate their transmission initiation times of REG according to a time generation function. Let *WTime(d)* be a wait time function for generating a wait time that an MN at depth *d* has to wait before it issues its REG message at every registration interval. Then, this function is subject to the following requirements

Req. 1: *For an MN staying at depth i and an MN staying at depth j, WTime(i) < WTime(j) if i > j.*

Furthermore, nodes near an IG compete more for channel access since they have to process more REGs or the bigger REG by aggregation delivered from their children. We define *WTime_Gap(i)* as the wait time gap between two MNs located at two adjacent depths *i* and *i+1*, thus it would be desirable to have the following inequality requirement.

Req. 2: *WTime_Gap(i) ≥ WTime_Gap(j)* if $i < j$.

Considering these two requirements, we devise the following time generation function to generate the wait time of an MN at depth *d*.

$$WTime(d) = W_1 \times a^{d-1} \tag{1}$$

where, $d \geq 1$, W_1 is the wait time of a node at depth one and a is the base of exponential function ($0 < a < 1$). Since a is a fractional number less than *unity*, *WTime(d)* is a decreasing function that has the maximum value W_1, thus satisfying *Req. 1.* Since the wait time W_1 of a node located at depth 1 should be at least larger than the amount of time for the other nodes to finish joining the tree plus the amount of time to receive all REG messages from its descendants, the W_1 value has to satisfy the following condition.

$$W_1 \geq (H-1) \times (T_{join} + T_{reg}) \tag{2}$$

where, H is a maximum tree depth. Then, we get the wait time gap between two MNs located at two adjacent depths as follows

$$WTime\_Gap(d) = |WTime(d) - WTime(d+1)|, 1 \leq d < H \tag{3}$$

Substituting Eq. (1) for Eq. (3), we get

$$WTime\_Gap(d) = W_1 \times (1-a) \times a^{d-1}, 1 \leq d < H \tag{4}$$

WTime_Gap(d) is an exponentially decreasing function of *a*. Considering a node that stays at depth *d*, the value of *WTime_Gap(d)* increases as *a* gets smaller and/or as depth *d* decreases, thus satisfying *Req. 2.* Since every node should send its REG message before nodes at the one-hop lower depth start sending REG messages in order to maximize aggregation, the minimum of wait time gap from Eq. (4) should be at least greater than the worst case of one-hop registration transmission in one interval, hence we get the following constraint.

$$Min\_WTime\_Gap = WTime\_Gap(H-1) > T_{reg} \tag{5}$$

Moreover, since a node can have its neighbors over three depths, its own depth, its depth minus 1, and its depth plus 1, we simply assume that each node has approximately $nNbrs/3$ neighbors at each depth. According to the join process discussed in Subsection 3.1, a node at depth i competes with its neighbors at depth i and its neighbors at depth (i - 1). Since its parent-to-be at depth (i - 1) has already joined a tree, the total number of competing neighbors is (2 × nNbrs/3 − 1). In the worst case, a node can start its own join process after the join completion of all the competing neighbors. On the other hand, during the registration process, a node only competes with its neighbors that stay at the same depth as its own depth for sending the REG message. Considering that a node's parent already has joined a tree, T_{join} and T_{reg} are given as follows.

$$\begin{gathered} T_{join} = T_{shd} \times (2 \times nNbrs/3 - 1 + 1) = T_{shd} \times \left(2 \times \frac{nNbrs}{3}\right) \\ T_{reg} = T_{shd} \times (nNbrs/3 + 1) \end{gathered} \tag{6}$$

Substituting Eq. (6) for Eq. (2), we get

$$W_1 \geq (H-1) \times T_{shd}(nNbrs + 1) \tag{7}$$

(b) Estimation of depth H and nNbrs

If the IG and all mobile nodes are uniformly distributed in a square area of m x m, the depth distribution of nodes in the tree topology can be obtained using the equation of cumulative distribution functions of distance l between two uniformly distributed nodes which is less than value of L in Eq. (12.1) and Eq. (12.2) in [10]. Referring to Fig. 7 in [10], we can obtain the distribution of depth according to variation of m when R = 250 m; for example, when the number of nodes is 100 and the transmission range of node = 250 m in area 1000 x 100 (m^2). We can see that the probability of depth = 5 is approximately 3 % and it is would be reasonable to infer that the possible maximum depth in this case is $H = 5$.

Furthermore, with the given network dimension $l \times w$, transmission range R, and the number of nodes n, we can calculate the distribution of the number of neighbors, $nNbrs$, of nodes in the network using the probability mass function proposed in Eq. (11) in [11]. Fig. 2 shows the distribution of node degree according to variations of network dimension with R = 250 m, and n = 100 nodes. Since the degree of a node corresponds to the average number of neighbors, $Nbrs$, that the node can have, we obtain $nNbrs$ = 15 with $l \times w$ = 1000×1000 (m^2) and n = 100 nodes.

3.1.3 Aggregated Transmission

A node does not have to report its upstream tree-link to its parent that already knows the link. Thus, we defined the reduced link state of node i, $RLS(i)$, as follows.

$$RLS(i) = \{(i,j) \mid j \in (NS(i) - P(i))\}$$

We also define the aggregated link state ALS(i) of node i as follows.

$$ALS(i) = \cup_{x \in C(i)} ALS(x) \cup RLS(i)$$

where, $ALS(x)$ is aggregated link state included in received REG from node x, a children of node i.

Aggregated transmission is performed as follows. Leaf node j sends REG = (ALS(j) = RLS(j)) to its parent as soon as its *WTime* expires. An intermediate node i saves ALS(k) received from its children k until its *WTime* expires, aggregates the saved ones with RLS(i) into ALS(i) when its *WTime* expires, and then sends REG = (ALS(i)) to its parent. Any aggregated packet that exceeds the maximum transmission unit (e.g. 2272 bytes in 802.11 WLAN) is segmented into the smaller ones before transmission.

3.2 Path Management

3.2.1 Path Discovery

A node acquires a path by sending a *path calculation request* (PC-REQ) message to an IG. However, this can increase path acquisition delay and overhead for the nodes near the IG. Since every internal node in a tree maintains partial topology information, a path can be discovered in a progressive way. A *progressive path* (PP) discovery is performed as follows. A source that needs a path tries to calculate a path using its own TG. If it cannot find a valid path, it sends PC-REQ to its parent and associates a timer, *PC-RES-timer*, with that PC-REQ, waiting for a *path calculation response* (PC-RES) message. Upon receiving the PC-REQ, a node calculates a path using its TG. If it succeeds, it responds to the source with PC-RES including the obtained path. Otherwise, it continues to forward the PC-REQ to its parent. If the source receives PC-RES before the *PC-RES-timer* expires, it sends data packets using the path. Otherwise, it performs the same path discovery process after some intervals.

3.2.1 Path Recovery

During packet transmission, if a node does not receive ACK after it sends a packet to a downstream node on the path, it judges that the corresponding link is broken. Then, the node issues a *route error* (RERR) message that includes the broken link to a source and at the same time, initiates a path discovery with PC-REQ that includes the broken link to find a salvage path to salvage the undelivered packets. Upon receiving PC-REQ, every node eliminates the broken link from its TG before calculating a path. Upon receiving RERR, a source initiates a new path discovery.

4 Performance Evaluation

Firstly, we select the proper value of W_1 and *a* based on the above analysis in section 3.1.2. Then, *TLSR-PP* (TLSR using the Progressive Path discovery), *OLSR+* [12] and

AODV-Hybrid combining AODV [2] and Hybrid scheme [3] are evaluated comparatively.

4.1 Simulation Environment and Performance Metrics

We conducted simulations using the commercial network simulator QualNet 5.02. We used three scenarios, S1, S2, and S3 that have an IG located at the center, the middle of the top, and the corner in the rectangular terrain of 1000 x 1000 (m^2), respectively. These scenarios were selected to generate various sizes of trees and consider different flooding ranges. In order to evaluate the proposed protocol, we make data communications between 15 flows of sources and destinations chosen randomly in MANETs with 512 bytes of packet size. The packet data rate at sources was fixed at 2 packets per second. The Random Waypoint was used to model the mobility of nodes with 30 seconds of pause time. Furthermore, the wireless bandwidth and the transmission range of nodes are set 2 Mbps and 250 m, respectively. All the simulations were performed within 600 seconds and the results were averaged after 10 runs with different seeds. The performance metrics such as *Control Overhead* (*CO*), *Delivery Ratio (DR)*, *End-to-end delay (E2ED)*, are used for evaluation with variation of the number of nodes (*nNodes*) and the maximum speed (*mSpeed*) of node.

4.2 Optimal Selection of W$_1$ and *a*

From the subsection 3.1.2 – (b), the maximum tree depth H and the average number of nodes $nNbrs$ are estimated to be 5 and 15 in the dimension of 1000 x 1000 m^2. Supposing that T_{shd} = 0.03 sec, by applying H = 5 and $nNbrs$ = 15 to Eq. (7), then W$_1$ should be greater than 1.92 (sec): W$_1 \approx$ 2 (sec). Moreover, with W$_1$ = 2 (sec) and H = 5, from Eq. (5) we can get the appropriate range of a = [0.6, 0.82] since Min_WTime_Gap > T_{reg}(= 0.18 calculated from Eq. (6)).

(a). DR with varying W$_1$ (b). E2ED with varying W$_1$

(c). DR with varying *a* (d). E2ED with varying *a*

Fig. 3. Examination of W_1 and a (S3, *nNodes* = 50/100, *mSpeed* = 10 m/s)

Then, we performed simulation using the dimension and the transmission range given in section 4.1, and scenario S3 with *nNodes* = 50, 100 and *mSpeed* = 10 m/s in order to choose an optimal value of *a* in [0.6, 0.82] complying with $W_1 = 2$ (sec).

Referring to Fig. 3-(a), the graph with 100 nodes indicates that the W_1 value should be increased as *nNodes* increases, which well complies with our previous analysis (see Eq. (7)). According to Fig. 3-(b), E2ED has a decreasing pattern from 2 to 2.6(s) and then an increasing one from 2.6 to 3(s) for both graphs. The reason is that the large value of W_1 may lower the accuracy of topology information received at an IG due to the long waiting time. It seems that $W_1 = 2.6$ (sec) is appropriate overall.

To choose an optimal value of *a* ranging between 0.6 and 0.82, we performed simulation by fixing $W_1 = 2.6$ (sec). Fig. 3-(c) shows that the DR decreases sharply when *a* increases from 0.6 to 0.9 in the scenario with 100 nodes. On the contrary, as can be seen from Fig. 3-(d), the E2ED of scenario with 100 nodes increases significantly over a = 0.6, while it shows slight increase with 50 nodes. From these simulation results, we conclude that the performance of TLSR is sensitive to the values of W_1 and *a*, and its best results are achieved when the values of W_1 and *a* are set to 2.6 (sec) and 0.6, respectively, in the given scenarios.

4.3 Comparison with AODV-Hybrid and OLSR+

4.3.1 Evaluation with Varying Speed

Fig. 4 – Fig. 6 compare three protocols with varying *mSpeed* when *nNodes* is 50 in scenario S3. As can be seen from Fig. 4, control overhead of TLSR-PP is considerably low compared to the other two's due to the use of aggregation and the removal of flooding. Referring to Fig. 5, the delivery ratio of TLSR-PP and OLSR+ slightly decreases with the increase in node mobility since nodes fail frequently to send the registration messages to update its link state with the IG in case of high speed, thus the IG does not have enough update on network topology to find paths for sources. This leads to the decrease in performance. On the contrary, since AODV-Hybrid can explore a path on demand, it is little sensitive to mobility. That is why the performance of AODV-Hybrid is not affected by node mobility. Nevertheless, its overall performance is still lower than that of TLSR-PP. This also explains why as node speed increases, the end-to-end delay of both TLSR and OLSR+ tends to increase linearly, while it seems to keep constant in case of AODV-Hybrid (see Fig. 6).

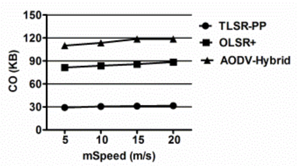

Fig. 4. Control overhead with varying *mSpeed* (S3, *nNodes* = 50)

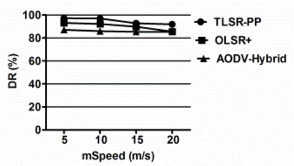

Fig. 5. Delivery ratio with varying *mSpeed* (S3, *nNodes* = 50)

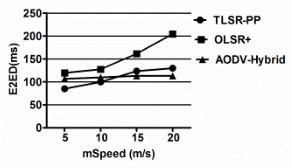

Fig. 6. End-to-end delay with varying *mSpeed* (S3, *nNodes* = 50)

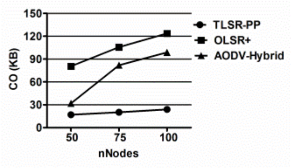

Fig. 7. Control overhead with varying *nNodes* (S1, *mSpeed* = 10 m/s)

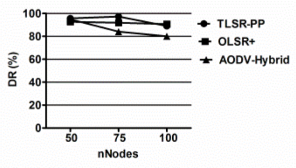

Fig. 8. Delivery ratio with varying *nNodes* (S1, *mSpeed* = 10 m/s)

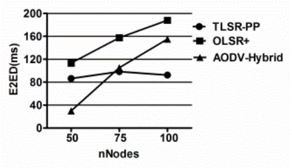

Fig. 9. End-to-end delay with varying *nNodes* (S1, *mSpeed* = 10 m/s)

4.3.2 Evaluation with Varying Node Density

Fig. 7 – Fig. 9 compare the three protocols with varying *nNodes* when *mSpeed* is fixed at 10 m/s in scenario S1. As can be seen in Fig. 7 the control overhead of TLSR-PP is sustained low compared to OLSR+ and AODV-Hybrid. This result comes from the utilization of message aggregation and the efficient topology management without using the flooding. However, because in the scenario S1 the IG is placed at the middle of the terrain, the increase in *nNodes* will increase the number of children of an IG and also increase the number of small sub-trees formed by those children. So, the TLSR-PP discovery does not make big improvement since PC-REQ will mostly reach an IG. Thus, the delivery ratio of TLSR-PP degrades slightly as *nNodes* increases.

5 Conclusion

We proposed a novel tree link state routing protocol (TLSR) in which topology management function and routing function cooperates closely. Not only do MNs exclude flooding in both mobility management and routing, but also they can make use of unicast mechanism to increase the reliability of transmission. To resolve a funneling effect, the TLSR employs a message aggregation technique that relies on skewed time synchronization to increase the degree of aggregation.

References

1. Sun, Y., Belding-Royer, E.M., Perkins, C.E.: Internet Connectivity for Ad Hoc Mobile Networks. International Journal of Wireless Information Networks 9(2), 75–88 (2002)

2. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999)
3. Ratanchandani, P., Kravets, R.: A hybrid approach to Internet connectivity for mobile ad hoc networks. In: Proceedings of the International Conference on Wireless Communications and Networking (WCNC), pp. 1522–1527 (2003)
4. Benzaid, M., Minet, P., Agha, K.A., Adjih, C., Allard, G.: Integration of mobile-IP and OLSR for a universal mobility. Wireless Networks 10(4), 377–388 (2004)
5. Clausen, T., Jacquet, P.: Optimized link state routing protocol (OLSR). RFC 3626. IETF (2003)
6. Yuste, A.J., Trujillo, F.D., Trivi, A., Casilari, E.: An adaptive gateway discovery for mobile ad hoc networks. In: Proceedings of the 5th ACM International Workshop on Mobility Management and Wireless Access, pp. 159–162 (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. Ahn, G.-S., Hong, S.G., Miluzzo, E., Campbell, A.T., Cuomo, F.: Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, Boulder, Colorado, USA, pp. 293–306 (2006)
9. Youngsoo, K., Sunghyun, C., Kyunghun, J., Hyosun, H.: Throughput enhancement of IEEE 802.11 WLAN via frame aggregation. In: Proceedings of the IEEE 60th Vehicular Technology Conference, pp. 3030–3034 (2004)
10. Dung, L.T., An, B.: A modeling framework for supporting and evaluating performance of multi-hop paths in mobile ad-hoc wireless networks. Computers & Mathematics with Applications 64(5), 1197–1205 (2012)
11. Lifang, G., Harfoush, K., Huimin, X.: Distribution of the Node Degree in MANETs. In: Next Generation Mobile Applications, Services and Technologies (NGMAST), pp. 162–167 (2010)
12. Yu, B., Zhang, W., Li, L.: Design and implementation of ad-hoc dynamic gateway based on OLSR and mobile IP. In: Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 2997–3000 (2009)