Research Article

Design and Optimization of Hierarchical Routing Protocol for 6LoWPAN

Ming Liu, Yang Xu, Shuai Wu, and Tian Lan

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Correspondence should be addressed to Yang Xu; xuyang@uestc.edu.cn

Received 20 May 2014; Accepted 13 July 2014

Academic Editor: Haigang Gong

Copyright © 2015 Ming Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To build scalable and lightweight wireless sensor networks with energy saving devices, 6LoWPAN-based protocols become emergent technologies that successfully constructed ubiquitous networks and seamlessly connected the traditional Internet with WSNs. However, the existing researches cannot effectively maintain the routing and the network topology for the problems of low cost sensors’ malfunctions. In this paper, we improved the HiLow protocol in aspects of interconnection between the Internet and 6LoWPAN and designed a network routing path recovery approach. In addition, we designed and modified the address mapping table in gateway to store and map nodes’ EUI-64 address and 16-bit short address. Nodes from the Internet can access the 6LoWPAN nodes by an IPv6 address, which is generated by the EUI-64 address and IPv6 prefix. Due to the benefits of additional data structure and our designs, the simulation results manifested that our protocol improves the routing efficiency and can effectively reduce the cost for network reorganization caused by node failures and mobiles.

1. Introduction

IPv6 over low power WPAN (6LoWPAN) protocol has introduced a powerful technology for WSNs to innovate personal area networks into low power personal area networks (LoWPANs) with IPv6, which is based on IEEE 802.15.4 [1]. 6LoWPANs are characterized by low energy consumption, low cost, low data rates, and flexible deployment devices and make the communication between the wireless sensor network and the conventional internet possible. Since the emerging of 6LoWPAN, the growing interest around 6LoWPAN has led the tide of new communication proposal creations and has been implemented by TinyOS and validated both in TOSSIM [2] and in a live WSN testbed. In all its multiple applications, 6LoWPAN was confined to interact with the surrounding environment or react to certain events. Thereby an effective and feasible routing protocol should be designed to help in balancing the traffic load and avoiding energy-based network partition and it can work with static and mobile networks, is self-managed, and is memory efficient as it tends to minimize the energy needed to perform routing.

The main contributions of this paper are as follows. (1) We formulate a prime numbered address allocation method to generate the nodes’ 16-bit short address in HiLow. (2) We design and modify the address mapping table in gateway and map every node’s 16-bit short address to its EUI-64 address; the IPv6 address of a node consists of its own EUI-64 address and the IPv6 prefix, which is distributed from the gateway. This modification makes the 6LoWPAN nodes able to be translated into external Internet nodes. (3) We propose a path recovery approach to solve the malfunctions of the key or coordinator nodes failed. It is based on our first innovation; nodes can decompose the production to its destination, which is the production of the prime addresses of all the relay nodes.

The remainder of this paper is organized as follows. Section 2 overviews and analyzes the related work on routing protocols and address allocation approach in 6LoWPAN, especially the working principle of the HiLow protocol. Then, we propose our novel designs and improvement work on HiLow protocol in detail in Section 3. The simulation results are presented in Section 4. Section 5 provides a conclusion of
our paper and discusses a few future directions for further improving the performance of our approach.

2. State of the Art

All the related research and developments have pointed out that the main reason of designing routing protocol for 6LoWPAN is to meet more complex applications and reduce the implementation complexity and achieve maximum scalability and robustness.

Venkataraman et al. [3] and Gomez et al. [4] modified the AODV protocol to avoid the network resource competitions in 6LoWPAN, but as they are still based on broadcast routing lookup mechanism, the network consumptions are huge, which are not fit for the 6LoWPAN low power restrictions. Shin et al. [5] proposed a lightweight configuration protocol for static 6LoWPAN network and assumed the network is flat and has a fair connectivity, but it has limitations in real domains. Felsche et al. [6] proposed a hierarchical routing protocol for 6LoWPAN. However, when the sensor network scales up, the address allocation will exacerbate the performance of the routing algorithm.

The distributed address allocation (DAA) [7] protocol can make the network fast by assigning the address with providing communication between the main nodes that allow connecting to network, but in real practical situation there are more cases of complicated network structure and unequally distributed nodes, and it has a week point in wasting the address as the network gets bigger and dynamical.

The 6LoWPAN ad hoc on-demand distance vector routing (LOAD) [8] is a simplified on-demand routing protocol based on AODV [9] and is one of the most promising ones according to the activity related to this protocol within the 6LoWPAN. The author designed an alternate path mechanism to solve the path failure, but it does not meet the situations of mass of full function devices out of order in sudden.

The hierarchical routing over 6LoWPAN (HiLow) [10] protocol is the most representative hierarchical routing algorithms that can be deployed in 6LoWPAN, but HiLow only defines address allocation algorithm and routing algorithm within a 6LoWPAN network without solving the interoperability issues with the external Internet IPv6 networks. It has not proposed a method to map the 16-bit short address to the IPv6 128-bit address and has not detailed the path recovery operation for node failures. In addition, HiLow only roughly sorts nodes into two types, and this category is not appropriate for home 6LoWPAN networks. Therefore, the above deficiencies in HiLow need to be modified to suit the practical 6LoWPAN network applications.

Therefore, these typical routing protocols and implementations are not suitable for large and scalable 6LoWPAN based sensor networks.

3. Design and Optimization of Modified HiLow Routing Protocol

Our design follows the basic idea of HiLow protocol; for the deficiencies of HiLow analysed in Section 2, we designed our modified HiLow routing (MHR) protocol. In MHR we make three improvements: prime numbered address allocation approach, modified address mapping table in gateway, and path recovery method.

3.1. The Basic Model. HiLow protocol deploys coordinator as a root of a routing tree and spreads branches using parent and child nodes and it operates multithops on the top of adaptation layer by using 6LoWPAN mesh-header formatted packets. Nodes that want to join a 6LoWPAN try to discover an existing 6LoWPAN by scanning. If there is no one in its personal operation space (POS), it becomes the coordinator node of a new 6LoWPAN. Otherwise, it selects an available node as parent node, and the child node tries to associate with the parent at the MAC layer and be allocated a 16-bit short address from the parent which is defined as follows:

\[ C_N = MC \times AP + N \quad (0 < N \leq MC), \]  

where \( C_N \) is the first children node, \( MC \) is the maximum number of children that a parent node can have, \( AP \) is the address of the parent, and \( N \) is the \( N \)th child node.

After obtaining the 16-bit short address, sensor nodes could distinguish each other and exchange packet within the PAN. To transmit a packet, HiLow assumes that every node knows its own depth of the routing tree that nodes can directly compute the address of next hop. Similarly, when the relaying node receives a packet from a source node, it obtains its own parent node address as follows:

\[ AP = \left\lfloor \frac{AC - 1}{MC} \right\rfloor, \]  

where \( AP \) is the address of parent, \( AC \) is the address of the current node which has received a packet, and \( \lfloor \cdot \rfloor \) is the symbol of the floor operation. The current node can determine whether it is the ascendant or the descendant node of the destination by using this formula. When the current node receives a packet, the next hop node to forward the packet can be calculated by the following three basic cases.

(i) If the current node \( C \) is one of the ascendant nodes of the destination \( D \), the next hop is \( AA(DC + 1, D) \).

(ii) If \( C \) is one of the descendant nodes of \( D \), the next hop node is \( AA(DC - 1, C) \), the parent node of \( C \), which can be calculated by formula (2).

(iii) Otherwise, the next hop is \( AA(DC - 1, C) \).

HiLow maintains a neighbor table and periodically transmits a beacon to check association status with its parent node. The neighbor table contains information on the parent and its child. When a node fails to check the association, it should try to reassociate with its previous parent by utilizing the information in its neighbor table.

3.2. Address Allocation Design. In our design, we proposed a bidirectional prime numbered address allocation (BAA) method for HiLow, and in BAA, each node is assigned a prime number as its short address and any paths can be identified by
the location identifier (LID), which is the production of the prime addresses of all nodes in the path.

Each LID can be decomposed to a sequence of prime numbers that identifies a unique path in the tree from the root, if each intermediate node in this path has its unique prime-numbered address. When a node checks who is the next relay, it can simply adopt a decomposing operation with its children that only the address of the next transmitter could be divided.

The address allocation process as shown in Figure 2 includes three phases as follows.

(i) Neighbor discovery: it includes two procedures, router solicitation (RS) and router advertisement (RA).

(ii) Node registration: two new ICMPv6 messages were defined, node registration (NR) and node confirmation (NC) message, which contains the LID of the parent node.

(iii) Location registration: the current node uses location registration (LR) and location confirmation (LC) messages to inform its LID to the parent node and the gateway.

As HiLow does not deal with external communications, we propose that IPv6 address is generated by combining IPv6 prefix and EUI-64 address. Meanwhile, the gateway maintains a mapping table that maps every node’s 16-bit short address to its EUI-64 address. When 6LoWPAN topology changes, it only changes the relation in mapping table accordingly. For external visit, gateway firstly inquires the mapping table to get the destination’s prime numbered 16-bit short address.

Figure 3 illustrates our configured IPv6 node address. PAI is the gateway identifier, and this field identifies the PAN to which the node belongs; this is optional and can be set to zero. PI is the proxy identifier and consisted of the node type and relationship that can produce the RA message, and this is helpful for grouping a set of nodes; the field is also optional and can be set to zero. NI is the node identifier and set by our approach.

3.3. Network Initialization. In this section, based on our prime numbered address allocation method, we constructed the network based on the following definition.

Definition 1. The product of all prime numbers in set Q can only be divided by the product of prime numbers that form the subset of set Q.

According to Definition 1, each LID can be decomposed to a sequence of prime numbers that indicates a path. If all the relay nodes have been assigned a different prime-numbered addresses, the product of all those prime numbers Q identifies the link from the node to the gateway. That is because a node can determine which neighbor is its ancestor node by checking whether the prime-numbered address of this neighbor can divide the Q value.

Staring with an existing root, the network construction process can be briefly described in Algorithm 1, and in the network initialization process, two parameters are required: a prime numbered 16-bit address AP and LID Q which is the product of all nodes in the path to the root. For example, if the path from a Node c to the root is A₁, A₂, ..., Aₙ, the LID of this node is Qₖ = A₁ × A₂ × ··· × Aₙ. The location of a node can be identified by its parent node, and the node can get its path from its own LID. If the LID of its parent node is Qₚ, the LID of Node c is calculated as Qₖ = A₁ × Qₚ. As shown in Figure 1, Node 17’s LID = 1 × 5 × 7 explicitly identifies the path from the root to Node 17.

As shown in Figure 4, the parent node of Node C is Node P, and the child nodes of Node C are Node Kᵢ (i = 1, 2, 3, ..., n). The LIDs of Node C and Node D are Qₚ and Qₐ and those of Node P and their child Node Kᵢ are Qₚ and Qₐ, respectively.

3.4. Mapping Table in Gateway. HiLow only defines address allocation and routing mechanism within a 6LoWPAN network without solving the interpretability issues with
the external Internet IPv6 networks. It has not proposed a method to map the 16-bit short address to the IPv6 128-bit address.

In our improvements, some items are added to the original mapping table (MT), including several node types, one relationship type, and two values for routing algorithm. The mapping table includes the following entries:

(i) PAN ID (16 bits)
(ii) Nodes Short Address (16 bits)
(iii) Nodes EUI-64 Address (64 bits)
(iv) Nodes Device Type (4 bits)
   0000: Gate
   0001: Common Router
   0010: Common End Node
   0011: RN
   0100: CN
   0101: MN
   0110-1111: Reserved
(v) Nodes Relationship (2 bits)
   00: Parent
   01: Child
   10: Common Neighbor
   11: Reserved
(vi) Nodes Depth (8 bits)
(vii) Interval (8 bits)
(viii) Stability (8 bits).

HiLow has not detailed the path recovery operation for node failures. In 6LoWPAN networks, different kinds of nodes have different requirements, especially the gateway which must deal with communication between 6LoWPAN and the Internet. In addition, nodes with mobility (MN) and nodes that deal with much more packets than common nodes (CN) should avoid being selected as a parent node, while real-time nodes (RN) that require high quality real-time reaction should select parent node with small depth, like alarm nodes. By converging node type and other factors, an evaluation process is proposed to help new-joined nodes to select an optimal parent node. The function \( F(t, e, n, d) \) is used for calculating the evaluation value, which depends on node type \( (t) \), energy left \( (e) \), the number of child nodes \( (n) \), and depth \( (d) \). New-joined node receives the evaluation value from all available parent nodes and makes a comprehensive evaluation in combination with its own node type and the link quality.

As the pseudocodes described in Algorithm 2, if the destination of packet \( x \) is the gateway, the gateway just accepts \( x \) and delivers it to upper layer. If packet \( x \) is directed to the Internet, the gateway gets next hop by inquiring the routing table and forwards \( x \). If packet \( x \) is from the Internet to 6LoWPAN, the gateway gets the EUI-64 address of destination node and uses it to inquire the MT for 16-bit short address. Then, it computes for next hop using the algorithm described in our design and forwards \( x \). If packet \( x \) is from 6LoWPAN to the Internet, the gateway first inquires the MT to judge whether destination of \( x \) is in MT. Packet \( x \) is directly delivered to destination if the destination is a neighbor of the gateway.

3.5. Path Recovery Algorithm. For energy exhaustion or unpredictable problem, nodes get failed and routing broken. What is worse, all descendant nodes of failed node are disconnected from the 6LoWPAN network and the routing table is broken. Though there already are some path recovery mechanisms used in 6LoWPAN, they have problems with efficiency and cost. HiLow designates platforms with the two sets of devices. So we take different measures to deal with different types of node failures.

On the basis of our BAA designed, we propose the path recovery method for nodes’ malfunctions; as presented in Figure 5, each node in this tree has its constant prime address and LID signs by blue annotation. When node 2 gets failed, as show in Figure 5, its descendant nodes parted into 4 descendant trees.
for each node \( i \) in network do
  if destination \( (x) \) in external then
    \( N \leftarrow \text{nextHop}(x) \);
    GatewayToTransmit \( (N) \);
  else
    \( A_{\text{EUI}} \leftarrow \text{getEUI}(x) \);
    \( A_{\text{short}} \leftarrow \text{inquire}_{\text{MT}}(A_{\text{EUI}}) \);
    \( N \leftarrow \text{nextHop}(A_{\text{short}}) \);
  end if
end for

for \( N \neq \emptyset \) do
  \( A_{\text{short}} \leftarrow \text{nextHop}(x) \);
end for
\( A_{\text{short}} \leftarrow \text{inquire}_{\text{MT}}(x) \);
end for

**Algorithm 2**: Gateway routing algorithm.

The process for path recovery is described in Algorithm 3. At first, all descendant nodes that disconnected from their parent node are formed into a new tree topology, with the direct child nodes of failed parent node acting as their roots (lines 2–6). Each node of the new descendant trees will scan for the available parent nodes and select an optimal one from its neighbors (line 6); while all its descendant nodes keep their associations, each descendant node can keep the original address and gains a new LID from its current parent without sending a rejoining application. The router then collects and modifies all new LIDs in the new descendant tree (line 10–13).

For the network maintains, all head nodes of the descendant tree have to scan for a new parent node and resign themselves. Within the scope of their communication, Node 3 was selected as their parent node and each descendant node regains a new LID but the inner association links are unchanged. This recovery method only involves two procedures: the descendant tree nodes scan for a new parent and all the descendant nodes regain a new LID. In contrast with the existing network restriction and nodes rejoin method, it removes the new address allocation of the rejoin nodes, which reduces the network restructuring costs and simplifies the network maintenance.

**4. Experiment and Results**

To evaluate the performance of our approach, we based in the Intel Berkeley Research lab [11] to design our simulation scenario: as shown in Figure 6, the lab simulation used Mica2Dot sensors with weather boards collected times tamped topology information, along with humidity, temperature, light, and voltage values once every 31 seconds. Data was collected using the TinyDB in-network query processing system, built on the TinyOS platform, and some additional nodes according to our simulation need to join the current network.

We also design a corresponding simulation scenario upon Network Simulator version 2 (NS2). The simulation experiment is constructed on Windows 7 with Intel Core i5-3470 processor (3.2GHz) and 8 GB RAM. The goal of the simulation is to show that our approach can provide a more stable transport environment in 6LoWPAN network. The results are also compared to the protocols of DAA, LOAD, and HiLow which are contrasted in Section 2.

In Table 1, we present the parameters configured for the conducted simulation experiments. The sensor nodes are located at known coordinates, 500 rounds are taken, and 10 packets are delivered in each round. The size of the network in each experiment has nuance setting for different algorithms. The parameters that we use in our experiments are average number of address obtained nodes per round, average delay of transmission (measured as the number of successfully delivered packets versus required packets), and average energy depletion ratio per round (measured as the energy consumption versus the initial energy).

**4.1. Address Allocation Efficiency**. We set up 3 scales of network simulation scenarios: the new-joined nodes of the
Algorithm 3: Path recovery process.

\begin{algorithm}
\begin{algorithmic}
\State\For {each node in 6LoWPAN} \Do
\If {node $\subseteq$ failedNode} \Then
\State neighbors $\cdot$ delete(parent);
\State node $\cdot$ form (tree) $\leftarrow$ getchild();
\State node $\leftarrow$ getClosCapNb(neighbors, Cap);
\State node $\cdot$ parent $\leftarrow$ FindNewParent(node);
\Else
\State nodes regular transmission();
\EndIf
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}

Figure 5: Path recovery of MHR.

Figure 6: Illustration of the simulation scene [11].
network at a steady rate and the number of coordinators and their positions are fixed. We test the success rate of short address allocation of the two algorithms; the success rates of node joining the network to obtain short address when $MC = 3$, $MC = 5$, and $MC = 7$ are shown in Figures 7, 8, and 9.

From these figures, we can find that BAA can guarantee that the success rates of obtaining short address are higher than those in DAA, and it also solves the problem of shortage of address space. Since the maximum number of its descendants is positively related to the MC value of the node, the smaller the parent node's MC value is, the less likely for the child to obtain a short address from the parent node. With the increasing value of $MC$, the parent node can allocate short address to more descendant nodes; thus, the success rate of network node to obtain short address improves. When there are a small number of nodes in the network, nodes have fewer opportunities to obtain short address and the success rate of nodes to obtain short address is low.

With the increasing number of nodes in the network, the chances that each node joins the network or is added as a ancestor node increase. The node that did not obtain the address previously then retries to find a new ancestor to obtain the address; therefore, the success rate of gaining short address increases and the success rate of short address acquisition in the network improves significantly.

By analysing of the factors of nodes joining failure, we find the main reasons. Firstly, the adaptation layer of 6loWPAN limits the $MC$ number of each coordinator; when node received the PANatCapacity message [12], the joining failed. The second factor is associated with the performance of 802.15.4 MAC layer. When the network size grows, the data arrival rate increases and the stability of 802.15.4 MAC layer will decline. In the simulation process, the cause of the failure for most of the nodes is due to the timeout of waiting ACK from the MAC layer. Due to the multihop topology of

| Simulation parameters                      | Value     |
|-------------------------------------------|-----------|
| Network area size                         | $200 \times 100 \text{m}$ |
| Maximum number of sensor nodes            | 200       |
| Available energy on sensors               | 35 J      |
| Sensing radius of sensor node             | 100 m     |
| Bandwidth of sensor node                  | 250 kb/s  |
| Number of simulation rounds               | 500       |
| Number of packets in each round           | 1         |
| Size of packet in each round              | 200 bits  |
| Probability of node failure               | 0.02      |
| Energy consumption of sending package     | 40 nJ/bit |
| Energy consumption of receiving package    | 80 nJ/bit |
| Energy consumption of amplifier           | 200 pJ/bit/m$^2$ |
6LoWPAN, the beacon message may conflict with data or control frames between coordinators, which makes the data or control frame not able to be correctly passed. In this case, the child nodes cannot track the beacon correctly to select time for sending data.

4.2. Communication Delay Reduction. On the basis of the network deployment, we add some additional nodes and form a 200-size tree topology scale scene, and the simulated time is set to 100 seconds, and we send 20 sets of end-to-end data. Node 9 is used as the source node, and Node 2 is used as the destination node; the topology is shown in Figure 5. Figures 10 and 12 show the delay of each hop consumption and the total end-to-end delay of these three types of routes in the process of successful transmission, respectively. Before LOAD sends the data to the next node, it is necessary to find and establish a route to the destination node. The process spends a total of 0.4875 seconds. After that, the routing nodes only need to find the next hop address in the local routing table. There are three jumps sending the data to the destination Node 2, and the second and third hops, respectively, consume 0.00389 seconds and 0.00517 seconds.

When sending the first data package, LOAD needs to broadcast routing lookup packets. After getting the route towards the destination node, at the time of 0.0578 seconds, the data ID is delivered to the destination node. Because the node’s routing table already has the route to the destination node, there is no need to initiate route lookup in the transmission process, so end-to-end delay decreases significantly.

After that, the floating of delay is around 0.02 seconds. Because MHR and HiLow facilitate real-time routing, the next hop node can be calculated directly without having to initiate route lookup process, so the delay of end-to-end data transmission is low, and the delay is about 0.015 seconds. In the process of LOAD route lookup, through constantly broadcasting and updating the routing table, the number of intermediate nodes from the source node to the destination node is small, and so after the stable transmission, the gap of end-to-end delay between both routing is not too big.

In the MHR and HiLow, before the data is forwarded, the algorithm calculates the next hop directly. The time consumption of these four hops is 0.01159 seconds, 0.01191 seconds, 0.01063 seconds, and 0.01127 seconds. The total delay of these three routing algorithms is shown in Figure 11. Although LOAD routes the data to the destination node after three hops, while the MHR and HiLow need four hops, the lookup process of LOAD routing is relatively long, so the total delay of on-demand routing LOAD is much larger than that of real-time routing MHR and HiLow.

4.3. Node Failure Recovery. The experimental setting of this scenario is the same as the former experiment. When nodes send the 100 sets of packages, the failure probability is set to this scenario in Table 1.
When nodes failed, the LOAD protocol will initiate a routing search process to find a new next hop node by broadcasting. Therefore, Figure 13 shows a sudden abrupt volume of transmission delay. After finding another route, compared with the previous one, the transmission delay increases slightly but tends to be stable. The only path between the source node and the destination node is cutoff, and the data packet cannot be routed to the destination node. Figure 13 shows that at time 10 the curve terminates. The reason why the delay in initiating a route lookup for MHR increases is that finding the next hop node consumes a certain amount of time. After finding another next hop node, the transmission delay is stabilized.

Energy consumption of nodes is shown in Figure 14. When node malfunctioned, in HiLow, it is not meaningful to draw energy curve. MHR broadcasts $mhr\_query$ to find the next hop node, and LOAD also broadcasts RREQ to find a new route; thus, the value of energy consumption in network for these two routing algorithms will be a sudden increase and shows that the slope of the straight line has a certain decline. Compared with that in LOAD at this time, the rate of energy consumption in MHR is slow, because after nodes in LOAD receive RREQ route request, nodes broadcast the RREQ continuously until they find the target node. After nodes in MHR receive the $mhr\_query$ message, they will determine whether to respond to $mhr\_response$ message according to whether there is a path to the destination node instead of broadcasting the message blindly. After finding the next hop node, MHR and LOAD forward the data directly through the unicast. Therefore, after some time, the energy consumption tends to be flat.

The MHR method maintains the advantages of real-time routing in tree topology: the route between any two nodes is calculated through the unique address instead of the traditional way such as broadcasting. When the network has shown a failure of nodes or link cases, the MHR can achieve fewer number of routing lookup messages and ensure the success rate of the end-to-end data transmission and low latency.

5. Conclusion and Future Works

In this paper, we make three improvements on hierarchical routing of 6LoWPAN networks based on HiLow, with the consideration of practical application scenarios; we altered the normative address allocation method and modified mapping table in gateway, which makes the external Internet able to access 6LoWPAN network nodes. In addition, we designed a corresponding path recovery for nodes’ failure or physical movement in the routing process. Our simulation results show the efficiency improvement and feasibility of our design.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research has been sponsored by the NSFC 61370151, 60905042, and 61202211, National Science and Technology Support Program of China 2012BAI122B05, and Huawei Technology Foundation no. YB2013120141.

References

[1] J. W. Hui and D. E. Culler, “Extending IP to low-power, wireless personal area networks,” IEEE Internet Computing, vol. 12, no. 4, pp. 37–45, 2008.
[2] R. Dutta, S. Gupta, and D. Paul, “The simulation and analysis of modified LEACH protocol using TOSSIM in wireless sensor networks,” International Journal of Computer Applications, vol. 82, no. 5, pp. 13–17, 2013.
[3] R. Venkataraman, M. Pushpalatha, and T. Rama Rao, “Regression-based trust model for mobile ad hoc networks,” IET Information Security, vol. 6, no. 3, pp. 131–140, 2012.
[4] C. Gomez, E. Kim, D. Kaspar et al., *Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing*, 2012.

[5] H. Shin, E. Talipov, and H. Cha, "Spectrum: lightweight hybrid address autoconfiguration protocol based on virtual coordinates for 6LoWPAN," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1749–1762, 2012.

[6] M. Felsche, A. Huhn, and H. Schwetlick, "Routing protocols for 6LoWPAN," in *IT Revolutions*, vol. 82 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 71–83, Springer, Berlin, Germany, 2012.

[7] J. P. Sheu, S. C. Tu, and L. H. Chan, "A distributed IP address allocation scheme in ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 1, pp. 10–20, 2008.

[8] M. J. Chang, Y. H. Yang, C. H. Chao et al., "Multipath design for 6LoWPAN ad hoc on-demand distance vector routing," *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, pp. 24–40, 2010.

[9] F. Yang and B. Sun, "Ad hoc on-demand distance vector multipath routing protocol with path selection entropy," in *Proceedings of the International Conference on Consumer Electronics, Communications and Networks (CECNet ’11)*, pp. 4715–4718, XianNing, China, April 2011.

[10] L. M. L. Oliveira, A. F. de Sousa, and J. J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks," *International Journal of Communication Systems*, vol. 24, no. 11, pp. 1445–1466, 2011.

[11] S. Chandrasekaran, O. Cooper, A. Deshpande et al., "TelegraphCQ: continuous dataflow processing," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 668–668, ACM, 2003.

[12] LAN/MAN Standards Committee, "IEEE Standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements. Part 15.4: wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15. 4TM-2003, 2003.
