Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 750681, 9 pages
doi:10.1155/2012/750681

Research Article

A Routing Scheme for IPv6-Based All-IP Wireless Sensor Networks

Wang Xiaonan and Zhong Shan

Department of Computer Science and Engineering, Changshu Institute of Technology, Jiangsu, Changshu 215500, China

Correspondence should be addressed to Wang Xiaonan, wxn_2001@163.com

Received 23 March 2012; Revised 18 October 2012; Accepted 24 October 2012

Academic Editor: Deyun Gao

The paper proposes a routing scheme for IPv6-based all-IP wireless sensor networks. The paper creates the IPv6 address structure and the IPv6 address configuration algorithm for all-IP wireless sensor networks. Based on the IPv6 address structure, the paper proposes the routing algorithm in the link layer for all-IP wireless sensor networks. In the routing scheme, a sensor node stores the next channel-sampling time of its neighbor nodes. In this way, only during the next channel-sampling time, a sensor node keeps active. and during any other time it returns to sleep. Therefore, the routing performance is improved. and the power consumption is reduced. Finally, the paper discusses the reduced IPv6 protocol stack performing the routing scheme. The routing scheme's performance parameters are evaluated, including routing power consumption and delay, and the data results show the correctness and efficiency of the scheme.

1. Introduction

With the dramatic growth of the WSN (wireless sensor network) application space and the emergence of a variety of new applications, WSN is required urgently to achieve the point-to-point communication with the Internet [1]. IPv6 has both abundant address resources and steady communication performance, so it becomes the ideal solution for the point-to-point communication between WSN and the Internet. In the IPv6-based all-IP WSN, a sensor node has a globally unique IPv6 address and uses the IPv6 protocol to achieve the point-to-point communication with the Internet.

Compared to traditional WSN, all-IP WSN has more extensive applications space. For example, in the modern agriculture field, farm laborers can use the Internet to access all-IP WSN in the agricultural environment and acquire the real-time agricultural parameters for monitoring without geographical location constraint.

At present, the following key technologies on implementing IPv6-based all-IP WSN need further researches [2, 3].

Address Autoconfiguration. The IPv6 address auto-configuration is the important technical feature of IPv6, and it can configure an IPv6 address for each interface in the absence of intervention. The feature is consistent with WSN's design goals, such as self-organization and self-configuration. At present, there are still some problems in implementing the existing IPv6 address auto-configuration strategies in WSN. For example, the stateful address configuration causes a lot of control messages, and the stateless address configuration needs to perform DAD (duplicate address detection) to ensure an address's uniqueness. Therefore, it is necessary to propose an IPv6 address auto-configuration algorithm for all-IP WSN.

Routing Mechanism. The WSN architecture is different from the IPv6 one. For example, in WSN, a sensor node works as both a node and a router. Therefore, the existing WSN routing schemes are not suitable for all-IP WSN. Therefore, it needs to propose a routing algorithm for all-IP WSN.

IPv6 Protocol Stack Optimization. Due to resource constraints of WSN and the fact that IPv6 was not initially designed for embedded applications, it is necessary to reasonably reduce the IPv6 protocol stack so that the reduced IPv6 protocol stack can be implemented in all-IP WSN.

Therefore, the paper proposes a routing scheme for IPv6-based all-IP WSN, and the paper has the following contributions:

(1) the paper creates the IPv6 address structure and the IPv6 address auto-configuration algorithm for all-IP WSN,

(2) based on the proposed IPv6 address structure, the paper proposes a routing algorithm in the link layer for all-IP WSN,

(3) the IPv6 stack performing the proposed routing algorithm is optimized.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work on all-IP WSN. We discuss the all-IP WSN architecture in Section 3 and the routing scheme in Section 4. The performance of the proposed scheme is analyzed in Section 5. We conclude the paper with a summary in Section 6.

2. Related Work

Reference [4] proposed an address configuration scheme for a sensor node based on location information, but the scheme was built on IPv4 and was not suitable for IPv6-based all-IP WSN. References [5, 6] proposed a reduced IPv6 stack which introduced an adaptation layer to achieve fragmentation and reassembly of an IPv6 packet. Reference [7] proposed a scheme for all-IP WSN. In the scheme, the network architecture was based on the logical grids, and the location information on sensor nodes was utilized to achieve both the address configuration and the routing discovery.

Reference [8] adopted the attributes of IEEE 802.15.4 [9] to enhance the routing performance of all-IP WSN. Border routers were used to store routing information on all sensor nodes, and the extension header, routing header, was included in each packet to achieve routing. Therefore, the fragment efficiency was reduced.

Reference [10] proposed a scheme on routing discovery and maintenance. However, the scheme only discussed how to establish routing paths reaching the destination subnet. In addition, the scheme did not analyze the relationship between the address hierarchical structure and the routing process. Reference [11] proposed a multipath routing scheme for 6LoWPAN to reduce the cost for reestablishing routing paths. In the scheme, the source node established multiple disjoint routing paths reaching the destination node through one routing discovery process and then ranked the multiple disjoint routing paths according to the link cost. The routing path with the minimum link cost was the primary path which was used to route the data. If the primary path failed, then the source node chose another routing path as the primary path to continue routing the data. The scheme effectively reduced the cost for reestablishing routing paths, but maintaining multiple disjoint routing paths increased the network resource consumption and reduced the routing performance.

The scheme [12] employed dispatch type in 6LoWPAN to determine the source/destination node of one packet, and intermediate nodes routed one packet to the next hop according to dispatch type encapsulated in the packet. Therefore, the routing delay was increased. In addition, the scheme added one header structure between the adaptation layer and the IP layer, so the transmission power was also increased. SPMIPv6 (Sensor Proxy Mobile IPv6) [13] presented the network architecture and the message formats and also evaluated the performance parameters, including the signaling cost and energy consumption. The data results showed that SPMIPv6 reduced the energy consumption significantly.

References [14–17] proposed the 6LoWPAN architecture where a sensor node only moved within a PAN (personal area network) with multiple gateway nodes. If a node's position changed, then it had to register the new position with all the gateway nodes. However, the architecture made no mention of IPv6 address configuration algorithm for all-IP WSN.

References [18, 19] proposed a scheme for all-IP WSN based on NEMO [20]. The scheme proposed the reduced IPv6 protocol stack, but it mentioned neither the IPv6 address auto-configuration algorithm nor the routing algorithm.

3. Architecture

The routing scheme is achieved in the link layer and adopts the 6LoWPAN architecture [6] where the MAC protocol is IEEE 802.15.4 [9]. IEEE 802.15.4 defines two kinds of nodes: FFD (full-function device) and RFD (reduced-function device). The scheme divides one all-IP WSN into multiple PANs, and from the routing perspective, one PAN includes three types of nodes: gateway nodes, cluster head nodes, and cluster members. Gateway nodes and cluster head nodes are FFD nodes with routing function, and cluster members are RFD nodes without routing function. Gateway nodes connect WSN to the IPv6 networks, and they communicate with each other in the multicast way through the IPv6 networks. One PAN includes only one gateway node with powerful hardware resources.

Taking into account the full integration of the WSN architecture and the IPv6 architecture, the scheme divides one PAN into multiple clusters, and each cluster has only one cluster head node. One gateway node and multiple cluster head nodes form a tree topology which is called a cluster tree, where the root node is the gateway node and intermediate/leaf nodes are the cluster head nodes. All the cluster trees in one WSN constitute the backbone routing system, as shown in Figure 1.

4. Routing Scheme

In the scheme, a node has an initial ID set by manufacturer, for example, the MAC address. The initial IDs of sensor nodes are independent of each other and unique in one all-IP WSN.

4.1. IPv6 Address Structure. According to the WSN characteristics, the following IPv6 address structure is proposed, as shown in Table 1.

Table 1: IPv6 address structure.

80 bits	16 bits	16 bits	16 bits
Global routing prefix	PAN ID	Cluster head ID	Cluster member ID
		Sensor node ID	

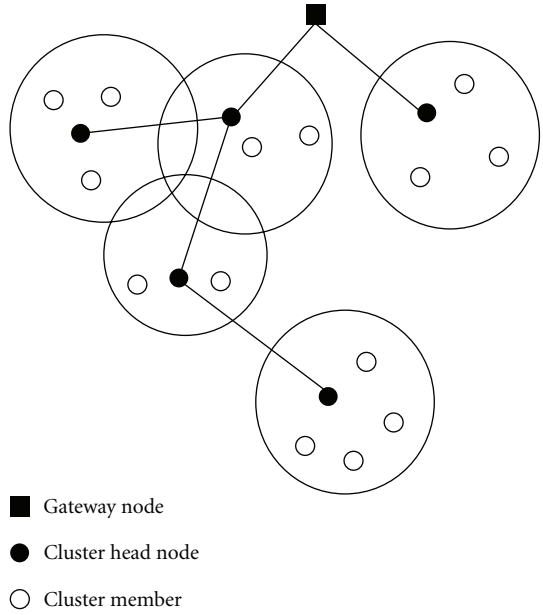


Figure 1: Cluster tree.

In Table 1, an IPv6 address is made up of two parts. The first part is global routing prefix, and global routing prefixes of all sensor nodes in one all-IP WSN are the same. The second part is sensor node ID, which is composed of PAN ID, cluster head ID and cluster member ID, and whose writing format is: PAN ID : cluster head ID: cluster member ID. PAN ID uniquely identifies one PAN which includes only one gateway node. PAN IDs of all sensor nodes within one PAN are the same, and its value is equivalent to the PAN ID of the gateway node in the same PAN, namely, the gateway node's initial ID. Cluster head ID uniquely identifies a cluster in a PAN, cluster head IDs of all cluster members in a cluster are identical, and its value is equivalent to the cluster head ID of the cluster head node in the same cluster, namely, the cluster head node's initial ID. Cluster member ID uniquely identifies a cluster member, and its value is equivalent to its initial ID.

In the scheme, the cluster head ID and cluster member ID of a gateway node are 0, and the cluster member ID of a cluster head node is 0.

Due to the resource constraints of sensor nodes, the routing scheme is achieved in the link layer. A sensor node's link address is its sensor node ID, that is, a gateway node's link address is its PAN ID, a cluster head node's link address is its cluster head ID, and a cluster member's link address is its cluster member ID.

4.2. Cluster Tree. IEEE 802.15.4 employs 3 mechanisms to achieve the data transmission [9]. These mechanisms include CSMA-CA, frame acknowledgment, and data verification. The scheme adopts the frame acknowledgment mechanism. In the frame acknowledgment mechanism, if a node is unable to handle the received frames, the frames are not acknowledged. If a node does not receive an acknowledgment frame within the specified time, it assumes that the transmission fails and retries the frame transmission. If an acknowledgment is yet not received after several retries, then the transaction is terminated.

4.2.1. Establishment of Cluster Tree. The routing scheme is built on cluster trees, and a cluster tree has the following characteristics:

(1) one cluster tree is made up of one gateway node and multiple cluster head nodes, where the root node is the gateway node, and the intermediate/leaf nodes are the cluster head nodes,

(2) one cluster tree forms one PAN which is uniquely identified by the PAN ID of the tree's root node, and PAN IDs of all cluster head nodes and cluster members in one PAN are identical,

(3) the gateway nodes are preset, and their IPv6 addresses are predetermined.

An FFD node becomes a cluster head node and acquires its sensor node ID through joining a cluster tree. Since a gateway node is preset, its neighbor FFD nodes first join the tree and become cluster head nodes through becoming the gateway node's child nodes.

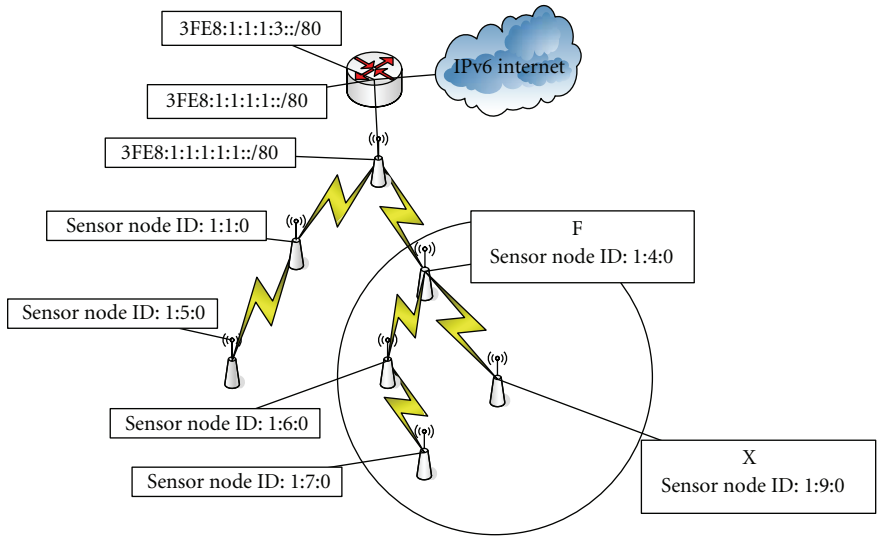


Figure 2: FFD becomes a cluster head node.

The process of an FFD sensor node X joining a cluster tree is as follows.

(1) X broadcasts a request beacon frame whose command identifier is 0×07.

(2) After X's neighbor gateway nodes/cluster head nodes receive the frame, they, respectively, return a response beacon frame whose payload includes the depth of the position in the corresponding cluster tree, the next channel-sampling time, and the PAN ID of the corresponding cluster tree.

(3) X selects the gateway node/cluster head node F with the minimum depth as its parent node and records F's cluster head ID, F's next channel-sampling time, and its own depth which is equal to F's depth plus 1. Then, X combines F's PAN ID with its initial ID to form a sensor node ID, marks itself as a cluster head node, and registers the sensor node ID with the gateway node of the cluster tree which it joins.

(4) X successfully joins a cluster tree to become a cluster head node and to obtain a sensor node ID, as shown in Figure 2.

In the scheme, a node's channel-sampling time means the time when a node awakens and can receive/send frames. In general, the number of nodes in a WSN is vast, so the topology of a cluster tree is maintained through a child node recording the sensor node ID of its parent node. In the scheme, the depth of a node's position in a tree is used as a metric to select a parent node in order to shorten the length of the routing path reaching the gateway node and reduce the routing cost and delay.

In Figure 2, the FFD node X with the initial ID 9 broadcasts a request beacon frame, and the cluster head nodes with the cluster head IDs 4, 6, and 7 return a response frame, respectively. X selects the cluster head node F with the cluster head ID 4 as its parent node and joins the cluster tree to become a cluster head node and obtain its sensor node ID.

4.2.2. Establishment of a Cluster. In the scheme, an RFD node joins a cluster to become a cluster member and acquire its sensor node ID.

After an FFD node H becomes a cluster head node, it periodically broadcasts a beacon frame whose payload includes the depth of its position in the cluster tree, the next channel-sampling time, and its PAN ID. After an RFD node X receives a beacon frame from H, it checks if it is marked as a cluster member. If not, then X marks itself as a cluster member, records the depth of H's position in the cluster tree, H's next channel-sampling time, and H's PAN ID, and combines H's PAN ID and cluster head ID with its own initial ID to obtain a sensor node ID.

From the previous cluster formation process, it can be inferred that a cluster is made up of only a cluster head node and multiple cluster members and has the star topology.

In the scheme, after a cluster member receives a beacon frame from its parent node, it updates the information on its cluster head node with the one in the frame's payload.

4.3. Failure of a Sensor Node. If a cluster head node X does not receive a beacon frame from its parent node within the specified time, then it considers that its parent node fails. In this situation, X chooses the neighbor cluster head node F with the minimum depth as its parent node, combines F's PAN ID with its initial ID to acquire a new sensor node ID, and records the depth of its position in the cluster tree. If X's sensor node ID/depth value is different from its original one, then it registers the sensor node ID with the gateway node, of the corresponding cluster tree.

A node acquires its sensor node ID from a neighbor node, as shown in Section 4.2. When a parent node fails, its child nodes choose the neighbor cluster head node with the minimum depth as a new parent node. If the velocity of convergence is not fast enough, then it means that the failed parent node is far from the leaf nodes which cannot be its child nodes' neighbor nodes. Therefore, the child nodes

cannot select the left node as its new parent node, and the routing loops are avoided.

If a cluster member X does not receive a beacon frame from its cluster head node within the specified time, then it considers that its cluster head node fails. Therefore, X chooses the neighbor cluster head node H with the minimum depth as its cluster head node and combines H's PAN ID and cluster head ID with its initial ID to acquire a new sensor node ID.

4.4. Registration. A gateway node maintains a cluster head node table which records the information on the cluster head nodes in the same cluster tree, and the information includes a cluster head node's cluster node ID and the depth of its position in the cluster tree.

The process of a cluster head node X registering a sensor node ID with the corresponding gateway node G is as follows.

(1) X sends G a Reg command frame whose payload includes its sensor node ID and the depth of its position in the cluster tree.

(2) After G receives the Reg frame, it checks if there is X's record in its cluster head node table. If there is, then G updates the depth value with the depth value in the frame. Otherwise, G adds a record into the table to store X's sensor node ID and depth value.

4.5. Routing

4.5.1. Routing Table. In the scheme, a gateway node/cluster head node stores a temporary routing table, and each routing entry consists of four fields: destination cluster, next hop, next channel-sampling time, and life time. Among them, the destination cluster records the sensor node ID of the destination cluster head node, the next hop field records the sensor node ID of the next hop reaching the destination cluster, the next channel-sampling time records the next channel-sampling time of the next hop, and its value is periodically updated with the one in the beacon frame from the next hop, and the life time field records the life survival time of the corresponding routing entry and automatically attenuates with the machine clock. When the life time is equivalent to 0, the corresponding entry is automatically removed from the routing table. Each time one routing table entry is used, its life time is set to the initial value, namely, the maximum life time.

4.5.2. Routing Process. An IPv6 node N uses the IPv6 address of a cluster member X to request the collected data, and the routing process is as follows.

(1) N sends a request packet to X.

(2) Through the IPv6 networks, the packet is routed to the gateway node G which identifies the cluster tree where X locates.

(3) G checks if there is a routing entry reaching the destination cluster in its routing table. If not, G establishes a routing path reaching the destination cluster head node H.

(4) G performs the fragmentation of the request packet, encapsulates each fragment with the mesh delivery header, where the source address is G's PAN ID and the destination address is X's sensor node ID, and the IEEE802.15.4 header, where the source address is G's PAN ID and the destination address is the cluster head ID of the next hop in the corresponding routing entry, and sends the frames.

(5) After the next hop receives the frames, it sends the frames to the next hop in the corresponding routing entry. In this way, the frames finally reach H which then forwards the frames to X.

(6) After X receives all the frames, it reassembles the fragments into the request packet, performs the fragmentation of the response packet, encapsulates each fragment with the IEEE802.15.4 header, where the source address is X's cluster member ID and the destination address is H's cluster head ID, and then sends the frames at H's next channel-sampling time. After H receives the frames, it sends the frames to its parent node. In this way, the frames finally reach G.

(7) G reassembles the fragments into the response packet and then sends the response packet to the IPv6 network, where the response packet reaches N in the IPv6 routing way, as shown in Figures 3(a) and 3(b).

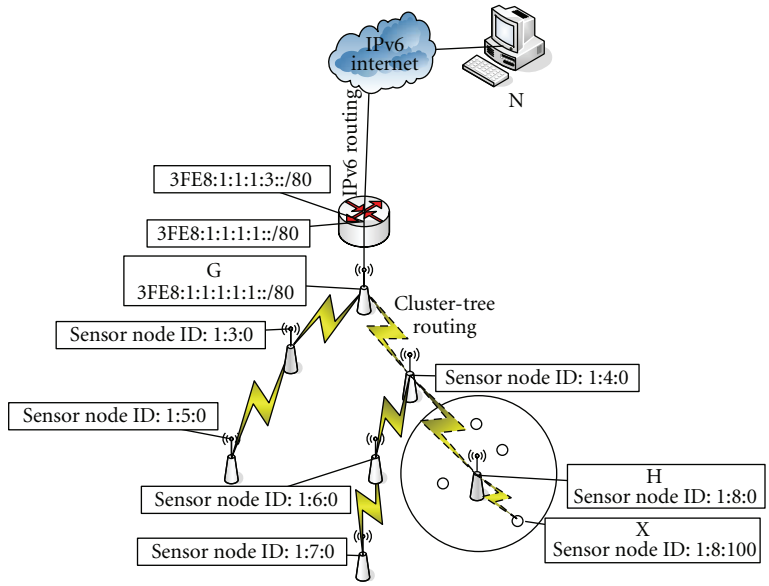
In Figures 3(a) and 3(b), the IPv6 node N sends a request packet whose destination address is 3fe8 : 1 : 1 : 1 : 1 : 1 : 8 : 100/80. The packet first reaches the gateway node G. G establishes the routing path reaching the destination cluster head node H and then routes the corresponding frames to H through the routing path. Finally, H forwards the frames to the destination cluster member X. After X deals with the request packet, it sends the response frames to G which routes the corresponding response packet to N.

In the scheme, a cluster head node periodically sends a beacon frame. If a cluster head node receives a beacon frame from the next hop, it updates the next channel-sampling time field of the next hop. Since a node only communicates with the next hop, only during the next channel-sampling time of the next hop it keeps active in order to send/receive the frames, and during any other time, it returns to sleep. In this way, the power consumption is reduced.

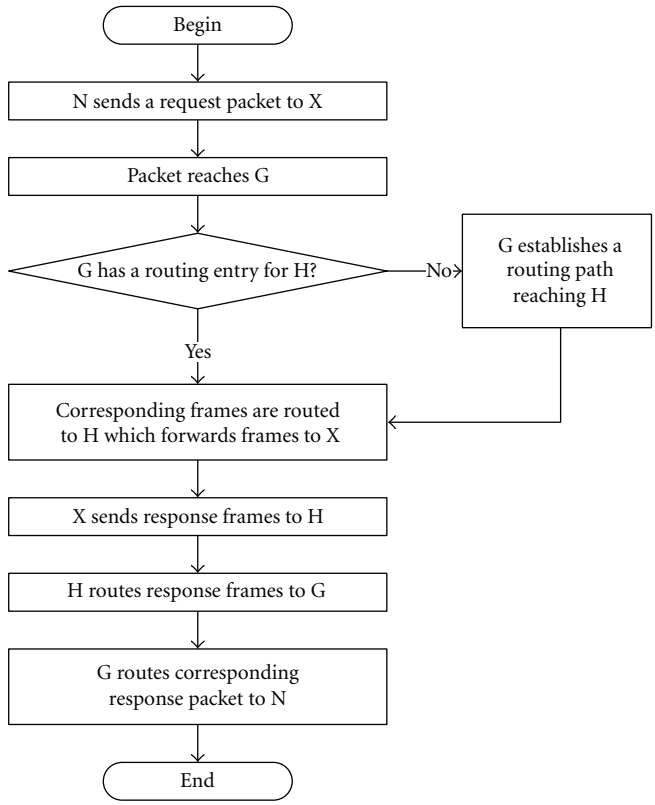
4.6. Data Frame. In IEEE 802.15.4, the command frame and data frame are defined by the frame type of the frame control. For example, if the frame type is 001/011, then the frame is the data frame/command frame. The IEEE802.15.4 data frame adopted by the scheme is shown in Table 2.

Routing of a data frame is performed through a cluster tree. The PAN ID of the source address of a data frame is the same as the one of the destination address of the frame, so the link address is 16-bit short address. If the next hop of a data frame is a cluster head node, then the destination address is the cluster head ID of the next hop. If the next hop of a data frame is a cluster member, then the destination address is the cluster member ID of the next hop.

The MAC payload format adopted by the scheme is shown in Table 3.



(a)



(b)

Figure 3: Routing process.

Table 2: Data frame.

2 bytes	1 byte	2 bytes	2 bytes	*n* bytes	2 bytes
Frame control	Sequence number	Destination address	Source address	MAC payload	Frame check sequence

Table 3: MAC payload format.

Mesh delivery header	Fragment header	Compression control header	IPv6 header and payload

Table 4: Mesh delivery header format.

10	O	F	Hop limit	Source address	Destination address
2 bits	1 bit	1 bit	4 bits	2 bytes	8 bytes

Table 5: First fragment header.

11000	Datagram size	Datagram tag
5 bits	11 bits	16 bits

Table 6: Subsequent fragment header.

11100	Datagram size	Datagram tag	Datagram offset
5 bits	11 bits	16 bits	8 bits

Table 7: Compression control header.

HC1	HC2
4 bits	4 bits

Table 8: HC1.

TF	Next header	HC2
1 bit	2 bits(01)	1 bit

Table 9: HC2.

Length	RSV
1 bit	3 bit

The mesh delivery header format is shown in Table 4.

In Table 4, O/F defines the type of the source/destination address. If O/F is 0, then the source/destination address is an EUI-64 address, or it is a short 16-bit address. The hop limit is the depth of the destination cluster head node's position in its cluster tree, and it decreases by 1 with one hop. The source address is the PAN ID of the gateway node of the tree where the destination cluster locates, and the destination address is the sensor node ID of the destination cluster member.

The fragment header includes the first fragment header format and the subsequent fragment header format, as shown in Tables 5 and 6 [6].

In Tables 5 and 6, the datagram size is the total size of an IPv6 packet. The datagram tag uniquely identifies an IPv6 packet, and the datagram tags of all fragments of an IPv6 packet are identical. The datagram offset is the offset value from the first fragment. The datagram offset of the first fragment is 0, so it is omitted from the first fragment format.

The compression control header is shown in Table 7.

In Table 7, HC1 specifies the way the IPv6 header is compressed, as shown in Table 8, and HC2 specifies the way the IPv6 payload is compressed, as shown in Table 9.

In Table 8, TF defines the compression method for the traffic class and flow label. Next header specifies the type of the next header, and 01 means UDP. HC2 determines the type of HC2 followed.

In Table 9, length determines the computing method for the compressed UDP length.

To sum up, the differences between the structure of the adaptation layer in the scheme and the one in 6LoWPAN are as follows.

(1) In the scheme, only the first fragment includes the IPv6 header and other fragments do not include the IPv6 header.

(2) The size of the control information in the scheme is reduced.

Therefore, the scheme reduces the transmission power consumption.

4.7. Security Consideration. The proposed scheme adopts the public-key infrastructure for 6LoWPAN [21] to achieve the security.

5. Performance Evaluation

5.1. Routing Analysis. The existing routing schemes [18, 19] are achieved in the network layer, while the proposed routing scheme is performed in the link layer. Therefore, in the existing routing schemes, the IPv6 header in each frame includes the 128-bit source/destination IPv6 address. In the proposed routing scheme, each frame contains 16-bit source link address and 64-bit destination link address. Therefore, for each frame the size of the redundant data in the existing schemes is 178 bits. It is assumed that the power consumed by transmitting one 127-byte frame between two neighbor nodes is e, and the distance from the destination cluster head to the root node of the corresponding tree is d. The power consumed by processing a frame is less by several orders of magnitude than the one by transmitting a frame [22], so it is negligible. Thus, the redundant power E consumed by transmitting one frame is shown in

$$E = \frac{(64 + 112) \cdot e \cdot d}{127 \times 8} = 0.175 \cdot e \cdot d. \tag{1}$$

In formula (1), e can be calculated from formulas (2), where $E_T(k, r)$ is the total power consumed by sending k bits, E_{tx} is the power consumed by sending 1 bit, ε is the magnification of the signal amplifier, r is the distance from the sending node to the receiving node, $E_R(k)$ is the total power consumed by receiving k bits, and E_{rx} is the power consumed by receiving 1 bit. Consider,

$$\begin{aligned} e &= E_T(k, r) + E_R(k), \\ E_T(k, r) &= k(E_{tx} + \varepsilon r^2), \\ E_R(k) &= kE_{rx}. \end{aligned} \tag{2}$$

According to references [23, 24], E_{tx} and E_{rx} are set to 50 nJ/bit, ε is set to 10 pJ/b/m^2, and k is set to 1016. It is assumed that FFD nodes are distributed evenly and r is set to 20 m. Then, Figure 4 is acquired.

It is assumed that the delay taken by transmitting 1 bit between two neighbor nodes is t'. Then, the redundant delay D taken by transmitting one frame is show in

$$D = \sum_{d}(64 + 112) \cdot t' = 178 \cdot \sum_{d} t'. \tag{3}$$

When t' is set to 4 μs, Figure 5 is acquired.

Figure 4: Redundant power consumption.

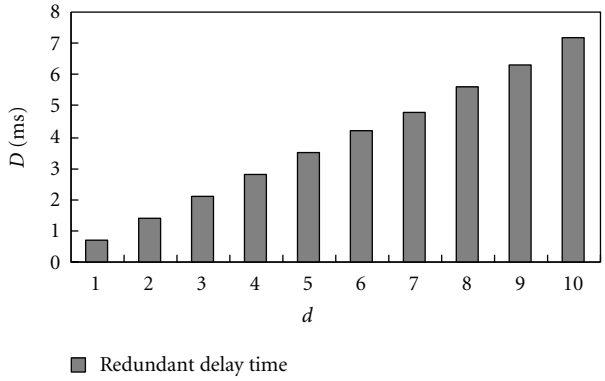


Figure 5: Redundant delay.

5.2. Routing Simulation. In ns-2, the simulation region is $100\times100\ m^2$, and the region includes 4 IPv6 ingress gateways, 20 FFD nodes, and 100 RFD nodes. The MAC protocol adopts IEEE 802.15.4, an FFD/RFD node's communication range is 20 m, and the bandwidth of WSN is 250Kbps. The FFD nodes are distributed uniformly around the simulation area, and the RFD nodes are distributed randomly around the simulation region.

We select the existing scheme [18] to compare with the proposed scheme due to the following reasons:

(1) the existing scheme is a routing scheme for IPv6-based all-IP WSN,

(2) the existing scheme has better performance than the typical routing protocols, such as AODV and LOAD.

The goal of the scheme is to reduce the routing cost and delay, so we evaluate the routing cost and delay, as shown in Figures 6 and 7.

The results of Figures 6 and 7 are analyzed as follows.

(1) The proposed routing scheme is achieved in the link layer, so the routing performance is better and the packet loss rate is lower.

(2) Only the first fragment includes the IPv6 header, and other fragments do not include the IPv6 header, so the fragment utilization is improved. As a result, the routing performance is improved.

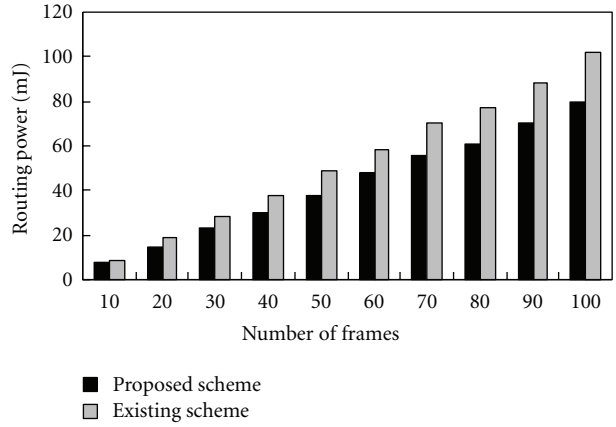


Figure 6: Routing power.

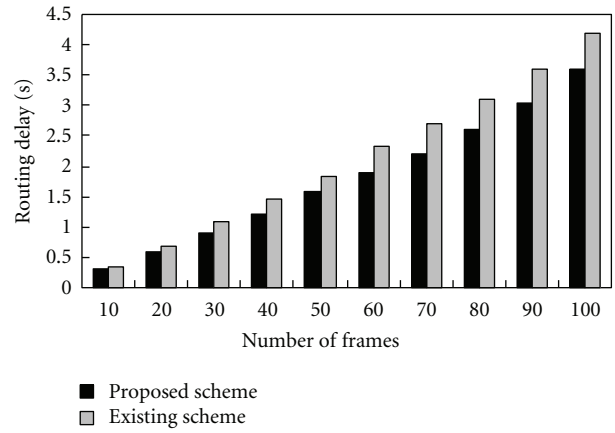


Figure 7: Routing delay.

(3) In the proposed scheme, the IPv6 stack is optimized and the size of the control information is reduced, so the transmission power consumption is reduced.

6. Conclusion

This paper proposes a routing scheme in the link layer for all-IP WSN. The paper evaluated the proposed scheme's performance parameters, including the routing power consumption and the routing delay, and the data results show the efficiency of the proposed scheme.

In the scheme, if a cluster head node near a gateway node fails, then the sensor nodes in the corresponding branch have to acquire new sensor node IDs and register them with the gateway. Therefore, the scalability is limited to some extent. In our future works, we plan to overcome the deficiency and improve the scalability.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61202440).

References

[1] A. Dunkels and J. P. Vasseur, "IP for smart objects," IPSO Alliance White Paper no. 1, september 2008.

[2] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[3] S. Dai, X. Jing, and L. Li, "Research and analysis on routing protocols for wireless sensor networks," in *Proceedings of the International Conference on Communications, Circuits and Systems*, pp. 27–30, New York, NY, USA, May 2005.

[4] A. Dunkels J Alonso T Voigt, "Making TCP/IP viable for wireless sensor networks," in *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, Swedish Institute of Computer Science, Sweden, 2004.

[5] N. Kushalnagar, G. Montenegro, and C. Schumacher, "6LoWPAN: overview, assumptions, problem statement, and goals," IETF RFC, 4919, 2007.

[6] G. Montenegro, N. Kushalnagar, and J. Hui, "Transmission of IPv6 packets over IEEE802," 15.4 Networks. IETF RFC, 4944, 2007.

[7] W. Xiaonan and Z. Shan, "All-IP communication between wireless sensor networks and IPv6 networks based on location information," *Computer Standards & Interfaces*, vol. 35, no. 1, pp. 65–77, 2013.

[8] J. W. Hui and D. E. Culler, "IP is dead, Long Live IP for wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 15–28, ACM Press, New York, NY, USA, 2008.

[9] IEEE Computer Society, "Part 15. 4: wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate Wireless Personal Area Networks (WPANs)," IEEE Standard 802.15.4, 2007.

[10] T. Winter and P. Thubert, "RPL: IPv6 routing protocol for low power and lossy networks: draft-ietf-roll-rpl-07," IETF, 2010.

[11] J.-M. Chang, H.-Y. Yang, and H.-C. Chao, "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.

[12] G. Bag, M. T. Raza, K. H. Kim, and S. W. Yoo, "LoWMob: Intra-PAN mobility support schemes for 6LoWPAN," *Sensors*, vol. 9, no. 7, pp. 5844–5877, 2009.

[13] M. M. Islam and E. N. Huh, "Sensor proxy mobile IPv6 (SPMIPv6)—a novel scheme for mobility supported IP-WSNs," *Sensors*, vol. 11, no. 2, pp. 1865–1887, 2011.

[14] G. Bag, M. T. Raza, H. Mukhtar et al., "Energy-aware and bandwidth-efficient mobility architecture for 6L0WPAN," in *Proceedings of the IEEE Military Communications Conference, (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.

[15] G. Bag, S. M. S. Shams, A. H. Akhbar, M. T. Raza, K. H. Kim, and S. W. Yoo, "Network assisted mobility support for 6LoWPAN," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference, (CCNC '09)*, Las Vegas, Nev, USA, January 2009.

[16] G. Bag, S. M. S. Shams, H. Mukhtar, K. H. Kim, and S. W. Yoo, "Inter-PAN mobility support for 6L0WPAN," in *Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology, (ICCIT '08)*, pp. 787–792, Busan, Korea, November 2008.

[17] M.-K. Shin and H.-J. Kim, "L3 mobility support in large-scale IP-based sensor networks (6LoWPAN)," in *Proceedings of the 11th International Conference on Advanced Communication Technology*, pp. 941–945, IEEE press, New York, NY, USA.

[18] H. Kim and C. hong, "A routing scheme for supporting network mobility of sensor network based on 6LoWPAN," in *Proceedings of the Asia-Pacific Network Operations and Management Symposium*, pp. 155–164, Springer Press, Berlin, Germany, 2007.

[19] H. Kim, C. hong, and T. Shon, "A lightweight NEMO protocol to support 6LoWPAN," *ETRI Journal*, vol. 30, no. 5, pp. 685–695, 2008.

[20] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility(NEMO) basic support protocol," IETF RFC, 3963, 2005.

[21] M. Hasan, A. H. Akbar, R. Riaz et al., "Key management in IP-based ubiquitous sensor networks: issues, challenges and solutions," in *Proceedings of the 1st International Conference of Ubiquitous Information Technology (ICUT '07)*, Dubai, United Arab Emirates, February 2007.

[22] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[23] R. Zhang, L. Zhang, and Y. Feng, "Very low energy consumption wireless sensor localization for danger environments with single mobile anchor node," *Wireless Personal Communications*, vol. 47, no. 4, pp. 497–521, 2008.

[24] E. M. Royer and C. K. Toh, "Review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, 1999.