Research Article

Multipath Load Balancing Routing for Internet of Things

Chinyang Henry Tseng

Computer Science and Information Engineering, National Taipei University, New Taipei 237, Taiwan

Correspondence should be addressed to Chinyang Henry Tseng; tsengcyt@gm.ntpu.edu.tw

In the next-generation technology, Internet of Things (IoT), billions of smart objects will communicate with one another to make human lives more convenient. IoT is based on wireless sensor network (WSN), and Zigbee is one of the most popular WSN protocols. A mature IoT environment involves heavy WSN data transmission causing bottleneck problems. However, Zigbee's AODV routing stack does not have load balance mechanism to handle bursty traffic. Therefore, we develop Multipath Load Balancing (MLB) Routing to substitute Zigbee's AODV routing. Our proposed MLB consists of two main designs: LAYER_DESIGN and LOAD_BALANCE. LAYER_DESIGN assigns nodes into different layers based on node distance to IoT gateway. Nodes can have multiple next-hops delivering IoT data. All neighboring layer nodes exchange flow information containing current load, used by LOAD_BALANCE to estimate future load of next-hops. With MLB, nodes can choose the neighbors with the least load as the next-hops and thus can achieve load balance and avoid bottlenecks. Compared with Zigbee's AODV and multipath version AODV (AOMDV), experiment results demonstrate that MLB achieves better load balance, lower packet loss rate, and better routing connectivity ratio in both grid and random uniform topologies. MLB provides a more convincing routing solution for IoT applications.

1. Introduction

Recently, the demands of Internet of Things (IoT) [1] keep growing. In the beginning, wireless sensor network (WSN) [2] enables ubiquitous sensing technologies. As the WSN technology evolves, the proliferation and application of these sensing devices create the Internet of Things (IoT). IoT is the next revolution, where the interconnection among smart objects creates an intelligent environment. It is estimated and expected to reach 24 billion IoT devices by 2020 [1]. As more and more IoT devices are connected and communicated, IoT applications generate tremendous IoT traffic. Since IoT traffic is for the communication between objects, the transmission reliability is critical, especially in a relatively unstable WSN, compared with wired network. As Figure 1 shows, IoT technology is applied in many domains, including environmental monitoring, transportation, automotive vehicles, industry [3], medical technology [4], healthcare, smart home [5], and smart city [6].

WSN is the most essential component of IoT, which comprises everything of WSN plus a thick layer of software installed across computational devices and the cloud. In other words, IoT is developed based on WSN, in which Zigbee is one of the most popular WSN protocols. In IoT, the low end sensors rely on WSN where data is transmitted from sensors (things) to the sink node (IoT gateway) using a multihop fashion. More static and mobile sinks can be deployed to collect data from sensors. Multiple sensor networks are connected together over the Internet. Therefore, performing data management is important. IoT research needs to find more efficient and effective ways of data management, such as collecting, modelling, reasoning, and distribution. We focus on data transmission reliability between things and IoT gateways.

We focus on Zigbee instead of Wifi because Zigbee is more health-friendly. Since IoT makes humans surrounded by wireless-connected objects, it is important to make all smart objects with low radio transmission power to make the environment healthier. Zigbee is with low transmission power, 1 mW, and is an appropriate option for IoT. Zigbee stack [7] adopts Ad Hoc On-Demand Distance Vector (AODV) [8] to automatically construct an ad hoc network as routes are needed. AODV optimizes routing paths to be the
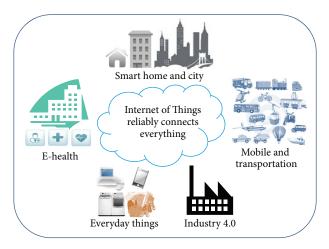
Figure 1: IoT application domains.

shortest but does not support multipath routing. Multipath routing is important to perform load balancing by selecting the less busy channel as the next-hop when network is under heavy traffic. In addition, once traffic bottleneck occurs, the unsuccessful delivery will trigger AODV route error (RERR) messages, which might generate more REER messages to jam the network. In the worst case, excessive AODV RERR messages can paralyze the network, particularly the links close to Zigbee sink. Therefore, we intend to enhance Zigbee routing by substituting AODV with our proposed routing protocol, Multipath Load Balancing, MLB, Routing.

In order to provide a reliable routing service for data-intensive IoT applications, we propose Multipath Load Balancing (MLB) Routing. Instead of cluster design, we use distributed architecture for MLB to avoid the situation that cluster heads become bottlenecks. MLB takes traffic load as the cost function and adaptively updates load information with neighbors to calculate the least busy routes. MLB offers multiple paths for the next-hop options to enhance reliability while evenly distributing traffic. The main compared target of MLB is Ad Hoc On-Demand Multipath Distance Vector (AOMDV) [9], which is a multipath version of AODV by giving equivalent paths with the same hop counts to IoT gateway. The shortcoming of AOMDV is that its equivalent paths must be thoroughly disjoint and cannot share nodes on their distinct paths. This limits the number of available alternatives. AOMDV does not consider traffic load when selecting the sending path because it is designed simply for multiple paths without considering load balancing issue. To become a better multipath solution for load balancing than AOMDV to support data-intensive IoT services, MLB has a reliable layered architecture and utilizes traffic load as the cost function. Layered architecture allows the routing computation to be done among neighbors locally and taking traffic load as the cost solves imbalanced load more directly than AOMDV. Thus, MLB can enhance network reliability by providing multiple next-hops and guarantees the shortest paths selected.

MLB consists of two main components: *LAYER_DESIGN* and *LOAD_BALANCE*. In LAYER_DESIGN, IoT gateway is the top level and we define that nodes *closer* to IoT gateway are in the *inner* layers and nodes *farther* from IoT gateway are in the *outer* layers. Each sensor node may play both roles of outer-layer and inner-layer nodes depending on the relative distance to IoT gateway compared with their neighbors. Each outer-layer node only needs to know the *local* information of next-hop nodes in the immediate inner layer to IoT gateway and the path from source to IoT gateway is constructed hop by hop. This structured and inductive two-layer relationship establishes the reliable routing service. Furthermore, LOAD_BALANCE allows each outer-layer node to calculate which inner-layer node is with the least traffic load. Consequently, the inner-layer node with less traffic is selected as the next-hop to IoT gateway. Through the cooperation of LAYER_DESIGN and LOAD_BALANCE, load balancing optimization is accomplished. In MLB routing table, multiple paths are recorded and allow more fault-tolerance once some next-hop fails. In case that any node fails to operate normally, MLB allows outer-layer nodes to recalculate their best inner-layer nodes toward IoT gateway without broadcasting route error messages. Therefore, MLB can quickly adapt sensor nodes to dynamic flow change and malfunctioned links.

The main contributions of MLB are multipath routing with load balancing, robustness, and reliability. First, load balancing is done by selecting the best inner-layer node with the least traffic load. Second, robustness is achieved because the synergy of LAYER_DESIGN and LOAD_BALANCE provides multiple inner-layer next-hops to IoT gateway for each outer-layer nodes, and ROUTE_RECOVERY can detect link failure for quick link switch. Since MLB eliminates bottlenecks by load balancing design and provides multipath routing, MLB provides a much more reliable routing service than current Zigbee's AODV related solutions.

To evaluate the performance of load balancing and reliability, simulation results are demonstrated based on three evaluation metrics, Load Balance Degree (LBD), packet loss rate (PLR), and connectivity ratio (CR). LBD illustrates load balancing performance for each layer in the routing topology and shows MLB balances network traffic more effectively than AODV and AOMDV, especially in the first layer. PLR directly shows the reliability of data delivery and CR shows the reliability of the entire routing topology. Because of the effectiveness of load balancing, PLR in MLB is much lower than PLR in AODV and a little lower than PLR in AOMDV. CR shows the routing service in MLB is very stable with perfect connectivity with larger among of data traffic. On the other hand, CR shows the routing services in AODV and AOMDV suffering from different degrees of disconnections due to the unbalanced traffic loading. Therefore, simulation results show that MLB provides better load balancing with more reliable packet delivery comparing with Zigbee's AODV and AOMDV.

The rest of this paper is organized as follows. Section 2 discusses the related works. Section 3 illustrates how Zigbee-MLB assists IoT communication. Section 4 presents MLB. Section 5 demonstrates simulation results. Finally, Section 6 concludes this paper.

2. Related Works

To avoid traffic congestion occurring at some bottleneck point, many load balance wireless routings are proposed. Pure cluster solutions [10, 11] aim at building as uniformly small-sized clusters as possible to achieve cluster load balancing. In minimum radius clustering algorithm [10], each cluster head extracts and sends local information to sink. After aggregating information from all cluster heads, SINK generates a balancing module to achieve cluster load balancing. Zhang and Yang [11] proposed a distributed algorithm to ensure the mean square deviation value of the number of nodes within each cluster as small as possible.

Advanced cluster solutions [12, 13] balance energy consumption on each node and indirectly achieve load balancing. Liao et al. [12] proposed a load-balanced clustering algorithm, DSBCA, for wireless sensor networks. To build a more balanced clustering structure and avoid forming large clusters, DSBCA considers the connectivity density and distance between nodes and base station. In each cluster, the node with the highest weight is selected as the cluster head. Weight calculations include the residual energy of node, the initial energy node, and the times of the node being elected as cluster head. During data transmission, each cluster head needs to aggregate all the data from its cluster members and send to the base station. However, data is sent to the base station through the cluster head which might become another potential bottleneck. Wu and Liu [13] proposed a centralized power efficient routing algorithm, EHGUC-OARP, for energy harvesting-wireless sensor networks. The base station uses EHGUC algorithm to form clusters of unequal sizes and select cluster heads of all formed clusters. When EHGUC is applied, the clusters with smaller size are closer to the BS. Subsequently, the base station uses OAPR to construct an optimal routing among all cluster heads.

Multipath Energy Aware AODV (ME-AODV) routing [14] utilizes network topology to divide the network into one or more logical clusters and restricts the flooding of route request outside the cluster. ME-AODV uses the remaining battery power of the nodes as the cost function instead of the hop count used in traditional AODV and adds multipath concept. The node uses all available paths in a round robin fashion in order to evenly distribute the energy consumption over the entire network. ME-AODV still has the drawback that cluster headers and cluster gateways have heavier load and become the bottleneck candidate. Neighbor-aware Adaptive Load Balancing algorithm [15] uses the information of parent and child nodes along with a probability factor to balance the traffic and prolongs network lifetime. All nodes send traffic load information to the gateway, which subsequently calculates and broadcasts the determined probability factor used by the whole network.

Different from mentioned solutions above, MLB uses traffic load as the cost function and noncluster structure can prevent potential bottlenecks occurring at cluster heads. MLB does not consider energy in its design because we focus on the IoT applications where sensors contain more energy and higher computation ability to send large amount of network data. Therefore, reliable routing services become more critical than saving energy for MLB.

3. How MLB Assists IoT Communication

As shown in Figure 2, WSN is the most essential component of IoT. IoT comprises everything that WSN plus a thick layer of software installed across computational devices and the cloud. Therefore, IoT can be explained as a general purpose WSN. In other words, WSN is a part of the IoT while IoT is not a part of SN. With regard to IoT communication, IoT follows the architecture of a three-layer WSN. Data is transmitted from Phase I sensors (things) to Phase II sink node (IoT gateway) using a multihop fashion. More static and mobile sinks can be deployed to collect data from sensors. WSN data is then sent to Phase III computational devices for further data analysis and IoT applications. Multiple sensor networks may be connected together over PHASE-IV-Internet.

Our work focuses on the reliability of data transmission of Phase I to successful data collection of PHASE II. For example, hospital may let patients wear Electrocardiography sensors to real-time monitor patients' heart health. Large amount of Electrocardiography data is continuously transmitted over WSN and cannot be lost because it involves human life. Only successful data transmission and collection of Phases I and II can provide computational devices complete data for correct data analysis and application. To find more efficient and effective ways of data transmission, we focus on data transmission reliability between things and IoT gateways (sensors and sinks).

MLB is proposed to cooperate with a large-scale wireless Zigbee network. When the traffic load increases significantly due to large number of IoT objects, routers with more neighbors might experience much heavier traffic load and become the bottlenecks, especially for the routers close to IoT gateway. To prevent bottlenecks from happening, MLB guides Zigbee routers to select the next-hops with the least traffic loads. Each MLB router in a Zigbee network serves network traffic with equal possibility to prevent itself from becoming a potential bottleneck.

Take a closer look at MLB in Figure 3. MLB provides an alternative routing service for Zigbee network without modifying existing Zigbee stack. Once a Zigbee router forwards IoT data, MLB guides Zigbee network layer to choose a next-hop with the least load toward IoT gateway. Besides, MLB also ensures that the current router's next-hop is closer to IoT gateway than the router itself to prevent routing loops. Therefore, MLB design can easily cooperate with the existing Zigbee stack.

4. MLB: Multipath Load Balancing Routing

As IoT applications grow rapidly, IoT sensors may deliver massive and critical data to IoT gateway so reliable IoT routing service is highly desirable. Current solutions, such as Zigbee and relative works in Section 1, cannot avoid bottlenecks, which may paralyze the entire network if the network traffic grows and congestion occurs nearby IoT gateway. To solve this problem, we propose Multipath Load Balancing (MLB) Routing to provide a reliable routing service for IoT applications, particularly data-intensive applications. Compared with AODV's improved multipath version, AOMDV, MLB
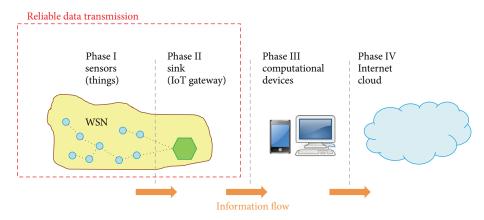
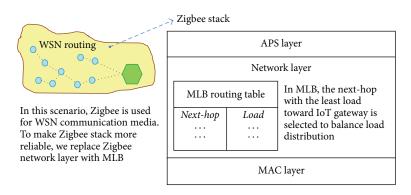FIGURE 2: Architecture of Internet of Things.

FIGURE 3: MLB routing in Zigbee stack.

has the same advantage of multipath but has better traffic load distribution and network reliability, MLB consists of two main designs: LAYER_DESIGN and LOAD_BALANCE. MLB is introduced from these three aspects in the following.

4.1. LAYER_DESIGN. When a node forwards a data packet to the gateway, it requires a routing service to generate the best next-hop choice for data forwarding. If the chosen next-hop fails to operate, a traditional single-hop routing service requires long response time to compute a new next-hop. Multipath routing services provide several next-hop choices so these services can quickly provide a new next-hop in case of the failure of current next-hop. As a result, multipath routing services provide a more flexible and reliable data forwarding services than traditional single-hop routing services.

For IoT applications, sensors require a reliable routing service to forward sensor data to the IoT gateway. In MLB, LAYER_DESIGN provides a reliable multipath data forwarding service with simple layered routing design. In LAYER_DESIGN, layer value presents the number of hops to the IoT gateway for each node. Layer 1 nodes, which are 1 hop away from the IoT gateway, have direct wired connections to the IoT gateway to avoid the gateway becoming a bottleneck. If all layer 1 nodes send packets via Zigbee wireless links to the gateway for a period of time, the gateway will encounter traffic jams because of shared media among wireless links. Besides, data traffic usually accumulates at layer 1 nodes so wired links

are required. Then layer 1 nodes can collect data from other nodes through Zigbee wireless links. The IoT gateway can have several layer 1 nodes collecting data from Zigbee links simultaneously without becoming a bottleneck itself.

To allow other sensor nodes to join LAYER service, layer 1 nodes broadcast their beacon messages to present layer 1 routing service. Other nodes hearing these beacon messages from layer 1 nodes can claim their layer number as 2 and announce their layer number through their beacon messages to present layer 2 routing service. Then layer 3 routing service can be presented in the same way and so on. A beacon message consists of its address, layer value, and network loading. The usage of network loading is defined in LOAD_BALANCE. While receiving beacons from neighbors, each node updates its neighbor table, which records neighbors' information including address, layer value, and network loading. Through beacon messages and neighbor tables, nodes can quickly establish their LAYER services toward the IoT gateway.

Figure 4 shows an example topology for presenting LAYER service. For layer 2 nodes, since layer 1 nodes are closer to the gateway, layer 1 becomes inner layer for layer 2 and layer 2 becomes outer layer for layer 1. Again, layer 2 can be the inner layer for layer 3 so a node may play both roles of outer-layer and inner-layer nodes in different relationships. Nodes can forward their packets to their neighboring inner-layer nodes as their next-hops until the packets reach the gateway.
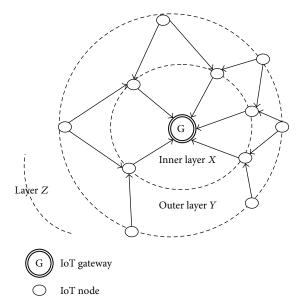
FIGURE 4: Multipath layer routing (LAYER_DESIGN) in MLB.

Since each node may have several neighboring inner-layer nodes, it can have multiple next-hops to forward packets. Again, these next-hops may have multiple forwarding choices from their inner-layer nodes so multipath routing establishes. For example, layer 3 node may have 3 neighboring layer 2 nodes, and each of these layer 2 nodes may have 3 neighboring layer 1 nodes, and thus 9 possible paths exist for this layer 3 node. If one of these paths becomes unavailable, the rest of 8 paths can be used. For traditional single path routing protocols, such as AODV, the node can have 1 path at a time, and it must generate another path in case this path becomes unavailable. Therefore, LAYER_DESIGN can provide a lot more paths than AODV and provide more reliable routing service. In addition, AOMDV requires that its multipaths must be disjoint paths. In other words, these paths cannot share the same nodes so AOMDV may have less available paths than LAYER_DESIGN according to network topologies.

In LAYER_DESIGN, given a node X, if one of the inner nodes becomes unavailable, node X can still use other inner nodes as its next-hop so LAYER_DESIGN can quickly adjust its path locally. If all inner nodes become unavailable, node X searches for layer values of other nodes in its neighbor table. At this time, the nodes with the highest layer value are usually peer of nodes for node X, and node X uses them as its new inner layer. Then node X updates its layer value, which is usually larger than its old value by 1 and announces it in its new beacon message immediately.

4.2. LOAD_BALANCE. To accomplish load balancing, each node chooses the next-hop node in the inner layer with lowest network loading while forwarding data packets. Each node announces its network loading in its beacon message to allow its outer-layer nodes to retrieve its network loading value. As a node forwards a data packet, the node chooses the next-hop with the lowest network loading among neighboring nodes

in the inner layer. Since each data packet is forwarded to the node with the lowest network loading, LOAD_BALANCE is done based on LAYER_DESIGN service.

If the network loading is determined based on current network loading of a node during a short period of time, nodes change their next-hops too often. This can trigger network loading dramatically and cause potential bottlenecks. So LOAD_BALANCE determines the network loading by the Estimated Network Loading based on exponential weighted moving average formula [16], in which newer data has heavier weighting and higher influence on next estimation value and the influence of data decreases exponentially with time. Therefore, the estimated loading can reflect a long term network loading so nodes switch next-hops smoothly.

Given a time slot x, its Estimated Network Loading (ENL) is denoted as ENL_x, and current Sample Network Loading (SNL) is denoted as SNL_x. If SNL_x is not 0, ENL_x becomes $(1 - w) * ENL_{x-1} + w * SNL_x$, where w is the weight of SNL to determine the influence of current traffic load for long term traffic estimation. If SNL_x is 0, ENL_x is set to 1/2 ENL_{x-1} to prevent ENL_x from becoming 0 in case w is 1. ENL_0 is initially set to SNL_0. If w is large, SNL has high impact of ENL so ENL changes fast, and ENL changes slow if w is small. Experiment results in Section 5.1 can show the impact of w to determine the best practice for LOAD_BALANCE. By using LOAD_BALANCE upon LAYER_DESIGN, traffic goes through nodes with lowest long term traffic loading dynamically. Therefore, MLB accomplishes loading balancing with reliable multipath layered routing.

5. Experiment Results for Evaluating MLB

Section 5 presents experiment results evaluating MLB compared with AOMDV and AODV based on the three evaluation metrics, LBD, PLR, and CR, which are presented in the following subsections. The experiment platform is ns2 with the following parameters: simulation time is 300 seconds, MAC layer is 802.15.4, field size is 250 meters × 250 meters, and transmission range is 50 meters. Data traffic type is constant bit rate, data packet size is 100 bytes, and data sending interval is 1 second. In order to show data traffic patterns in different types of network topologies, the experiment is conducted in a grid topology with 85 nodes and a topology with 100 nodes deployed by random uniform distribution. Layer 1 nodes have direct wired link with the IoT gateway. In order to demonstrate extreme situations as stress tests, this experiment launches data traffic from half of the nodes in the two topologies. The layer number of the data sending nodes is larger than the layer number of the other nodes, which only forward data traffic. Each topology uses MLB, AOMDV, and AODV separately to evaluate their performance in terms of LBD, PLR, and CR.

5.1. Load Balance Degree. In the experiment, nodes in layer 1 have direct access to the IoT gateway, and nodes from other layers transmit data via Zigbee connections to the nodes in layer 1. So the data traffic accumulated at layer 1 is much larger than other layers, and thus load balance in layer 1 is critical.
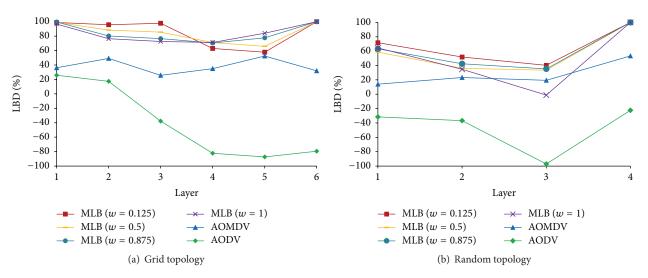
(a) Grid topology

(b) Random topology

FIGURE 5: Load Balance Degree in two topologies.

To evaluate the performance of load balance, we present LBD, which is calculated by standard deviation (SD) of traffic load and Load Average (LA) among all nodes in a layer:

$$LBD_Y = \left[1 - \left(\frac{SD_Y}{LA_Y}\right)\right] \times 100\%, \qquad (1)$$

where and LBD_Y, SD_Y, LA_Y denote LBD, SD, and LA in the layer Y.

If SD_Y is 0, it shows all nodes in layer Y have the same traffic load. In this case the load balance performance is the best and LBD_Y is 100%. If LBD_Y is 0%, it shows SD_Y equals LD_Y. In other words, the deviation of traffic load in layer Y equals the average load so the load balancing in this layer is poor. If LBD_Y is negative, the load balancing in this layer is even worse because it indicates traffic load of some nodes is even larger than average plus standard deviation.

Figure 5 shows LBD in two types of topologies using AODV, AOMDV, and MLB with four different w ($w = 0.125$, $w = 0.5$, $w = 0.875$, and $w = 1$). Since load balance in layer 1 is critical, LBD_1 is the key observation point. In both grid and random topologies, MLB with $w = 0.125$ provides the best LBD, particularly the inner layers, layers 1, 2, and 3.

In the grid topology, LBD_1 in MLB cases is 100% so MLB can provide the best load balance performance in the most critical layer, layer 1, which usually accumulates all data traffic. In addition, MLB with $w = 0.125$ can keep such an optimal load balancing performance through layers 1 to 3. On the other hand, LBD in AOMDV case is about 40% and is much worse than MLB. LBD in AODV case even quickly drops to negative value.

In the random topology, the node distribution is not uniformly deployed compared with the grid topology so the load balance task becomes much more than challenging. The number of connections of grid topology is fixed and the traffic is easier to be predicted and optimized. In general, LBD values in the random topology become worse than those in the grid topology. LBD in MLB with $w = 0.125$ case is still the best

and its LBD_1 is about 75%, which is much better than the case of AOMDV and AODV. Therefore, MLB shows much better load balancing performance than both AOMDV and AODV do. AOMDV provides multiple paths for data traffic so its LBD is much better than AODV, which is used in Zigbee stack. MLB can even provide better load balancing performance than AOMDV. In the grid topology, MLB can even provide the optimal load balancing performance in the inner layers. In the most outer layer, LBD values of all MLB cases are 100 % for both topologies. It shows MLB can perform optimal load balancing performance as data traffic initiates. When data traffic aggregates and arrives at layer 1, MLB can still perform great load balancing performance.

5.2. Packet Loss Rate. PLR directly reflects network performance of IoT applications. As data traffic sources increase, the network loading of the entire network increases. Since data traffic is usually accumulated at layer 1, the network loading of layer 1 can increase significantly in case of poor load balancing performance. So PLR can directly reveal the impact of load balancing performance for IoT applications.

Figure 6 shows PLR of MLB, AOMDV, and AODV versus different number of data traffic sources. Since load balancing performance of AODV is poor, PDR increases significantly as the data traffic sources increase. Because data delivery in the random topology is more challenging than the grid topology due to the node deployment policy, PLR increases much more in the random topology, which is actually a more realistic topology than the grid one. From Figure 6, AODV's PLR increases dramatically so AODV is not a scalable choice for IoT applications. In the grid topology, both AOMDV and MLB have low PLR, but only MLB can still keep low PLR in the random topology. AOMDV's PLR in the random topology can become up to 60% compared with 40% in MLB case. It shows MLB is a scalable solution compared with AOMDV in the challenging random topology. Therefore, according to the results in Figure 6, MLB is a more reliable and scalable
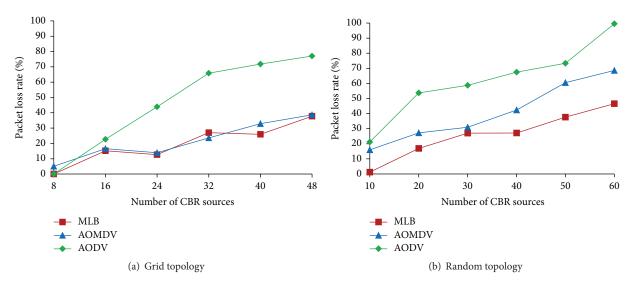
(a) Grid topology

(b) Random topology

FIGURE 6: Packet loss rate.
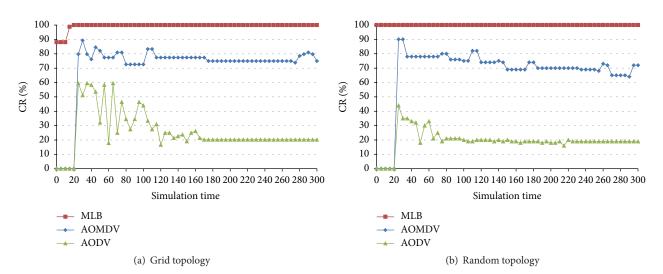

(a) Grid topology

(b) Random topology

FIGURE 7: Connectivity ratio.

routing solution than AOMDV and AODV for IoT applications.

5.3. Connectivity Ratio. CR is the ratio of nodes having routes toward the IoT gateway over all sensor nodes. CR illustrates the routing connectivity between nodes and the gateway in real time and thus shows the routing reliability of routing protocols for IoT applications. As the network traffic increases, the traffic jams may trigger routing errors. If the routing protocol cannot repair the errors instantly, these routing errors can propagate and trigger more routing error messages, which cause more nodes that are unable to connect the gateway. CR traces the ratio of these nodes that cannot connect to the gateway and monitors the impact of routing errors in real time.

Figure 7 shows CR of MLB, AOMDV, and AODV in 300 seconds. In the beginning, MLB takes shorter time to establish the routes from nodes to the gateway than AOMDV and AODV. Unlike AODV and AOMDV, nodes in MLB just need to connect the upper layer nodes without building a path to the gateway by flooding routing messages, and thus the routing establishing time is shorter in MLB than in AOMDV and AODV. After the routing topology has been stabilized, as half of the nodes send data traffic and the number of nodes is up to 100, CR in MLB is 100% in both topologies. This shows MLB has great routing reliability under the high traffic condition.

On the other hand, CR in AOMDV is about 80%, and this may result from the regulation of AOMDV multipath routing: the routes in AOMDV must have disjoint nodes. This rule may limit the route recovery capability in AOMDV and result in the routing vulnerability. CR in AODV is even worse due to the routing errors caused by the high traffic volume. Since AODV only supports one next-hop for each route, the routing recovery capability is the worst so the routing reliability is the worst. Therefore, the results of CR show that MLB is the most reliable routing protocol among the three routing protocols.

6. Conclusion

As IoT applications grow rapidly, reliable routing is highly desirable to allow IoT sensors delivery data packets to IoT gateway through multihop transmissions accurately. For preventing bottleneck issues in Zigbee's AODV routing services, MLB is proposed to provide a load balancing, robust and reliable routing service for IoT applications. To achieve these goals, MLB consists of LAYER_DESIGN and LOAD_BALANCE. LAYER_DESIGN provides a multipath layer routing service toward IoT gateway for IoT applications, and LOAD_BALANCE estimates load information for data sender choosing the inner-layer next-hop with the least network loading. The synergy of LAYER_DESIGN and LOAD_BALANCE eliminates the bottlenecks and thus provides a load balancing and reliable routing service. The experiment results demonstrate that MLB achieves much better load balancing than AODV and AOMDV according to LBD values. Based on PLR and CR, MLB provides more reliable routing than AODV and AOMDV. In conclusion, based on the load balancing design, MLB provides the most reliable routing service for IoT applications compared with the current famous in-use routing solutions, Zigbee' AODV and its improved multipath version, AOMDV.

Competing Interests
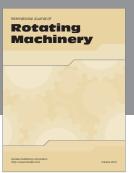
The author declares no competing interests.

References

[1] M. A. Feki, F. Kawsar, M. Boussard, and L. Trappeniers, "The internet of things: the next technological revolution," *Computer*, vol. 46, no. 2, pp. 24–25, 2013.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] L. D. Xu, W. He, and S. Li, "Internet of things in industries: a survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[4] Y.-W. Wang, H.-L. Yu, and Y. Li, "Internet of things technology applied in medical information," in *Proceedings of the IEEE International Conference on Consumer Electronics, Communications and Networks (CECNet '11)*, pp. 430–433, April 2011.

[5] G. Chong, L. Zhihao, and Y. Yifeng, "The research and implement of smart home system based on internet of things," in *Proceedings of the International Conference on Electronics, Communications and Control (ICECC '11)*, pp. 2944–2947, IEEE, Zhejiang, China, September 2011.

[6] P. Vlacheas, R. Giaffreda, V. Stavroulaki et al., "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 102–111, 2013.

[7] Zigbee Alliance, *ZigBee Specification Version 1.0*, 2004, http://www.zigbee.org.

[8] C. Perkins, E. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," IEEE RFC 3561, 2003.

[9] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[10] M. Hammoudeh, O. Aldabbas, S. Mount, S. Abuzor, M. Alfawair, and S. Alratrout, "Algorithmic construction of optimal and load balanced clusters in wireless sensor networks," in *Proceedings of the 7th International Multi-Conference on Systems Signals and Devices (SSD '10)*, pp. 1–5, IEEE, Amman, Jordan, 2003.
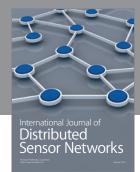
[11] J. Zhang and T. Yang, "Clustering model based on node local density load balancing of wireless sensor network," in *Proceedings of the 4th International Conference on Emerging Intelligent Data and Web Technologies (EIDWT '13)*, pp. 273–276, Xi'an, China, September 2013.

[12] Y. Liao, H. Qi, and W. Li, "Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1498–1506, 2013.

[13] Y. Wu and W. Liu, "Routing protocol based on genetic algorithm for energy harvesting-wireless sensor networks," *IET Wireless Sensor Systems*, vol. 3, no. 2, pp. 112–118, 2013.

[14] A. Bhatia and P. Kaushik, "A cluster based minimum battery cost AODV routing using multipath route for ZigBee," in *Proceedings of the IEEE 16th International Conference on Networks (ICON '08)*, pp. 1–7, New Delhi, India, December 2008.

[15] K. Kim, K. Cho, and S. Bahk, "Neighbor-aware adaptive load balancing algorithm for dense wireless sensor networks," in *Proceedings of the IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS '09)*, 2009.

[16] J. M. Lucas and M. S. Saccucci, "Exponentially weighted moving average control schemes: properties and enhancements," *T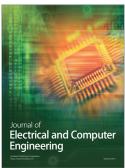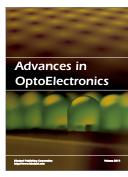echnometrics*, vol. 32, no. 1, pp. 1–12, 1990.