National Institute of Technology, Warangal

Internet of Things

Minor 2 Assignment

Topic - ***<u>Load Balancing in IOT Protocols</u>***

Divyanshu Shrivastava
Roll No. - 187909
Reg No. – MC18111
MCA – 3rd Year

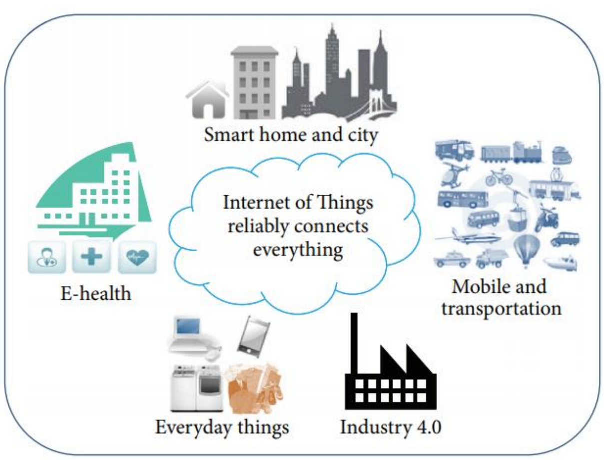
INDEX

1. Description

A large number of generated events from IoT objects causes overhead on the network. Therefore, to optimize the usage of IoT network, it is essential to provide solutions for network problems including scalability, routing, reliability, security, energy conservation, network lifetime, congestion, heterogeneity, and quality of service (QoS). In this regard, load balancing as an efficient method takes a significant role in distributing loads among different routes. Imbalance traffic load across the network causes high latency in some routes and loss of data packets and decreases packet delivery ratio. Although load balancing has a critical importance in the IoT, there is still a lack of an organized and comprehensive review about analyzing and examining its remarkable methods.

2. Introduction

Recently, the demands of Internet of Things (IoT) keep growing. In the beginning, wireless sensor network (WSN) 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. 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. IoT technology is applied in many domains, including environmental monitoring, transportation, automotive vehicles, industry, medical technology, healthcare, smart home, and smart city.



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 adopts Ad Hoc On-Demand Distance Vector (AODV) to automatically construct an ad hoc network as routes are needed. AODV optimizes routing paths to be the 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), 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.

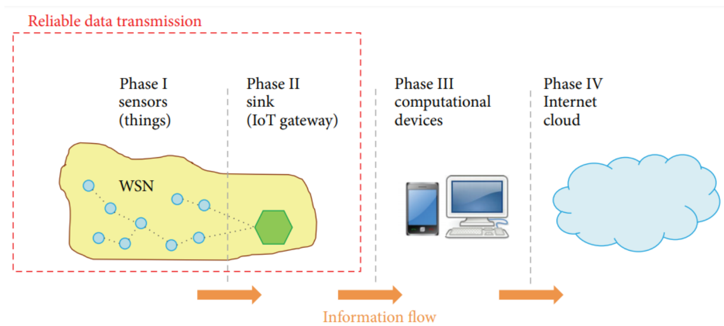
The main contributions of MLB are multipath routing with load balancing, robustness, and reliability. First, load balancing is done by selecting the best inner-layer node with the least traffic load. Second, robustness is achieved because the synergy of LAYER_DESIGN and LOAD_BALANCE provides multiple 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.

3. Performance Evaluation

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.

4. 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.



5. 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 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.

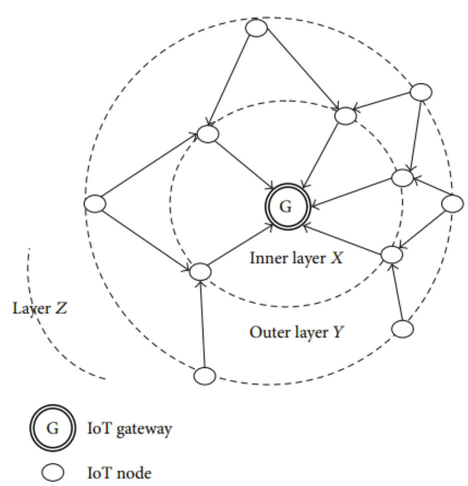
5.1. LAYER_DESIGN

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 layers 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.

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.



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, if one of the inner nodes becomes unavailable, node 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 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, and node uses them as its new inner layer. Then node updates its layer value, which is usually larger than its old value by 1 and announces it in its new beacon message immediately.

5.2. 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, its Estimated Network Loading (ENL) is denoted as, and current Sample Network Loading (SNL) is denoted as. If is not 0, becomes, where is the weight of SNL to determine the influence of current traffic load for long term traffic estimation. If is 0, is set to 1/2 to prevent from becoming 0 in case is 1. ENL_0 is initially set to SNL_0. If is large, SNL has high impact of ENL so ENL changes fast, and

ENL changes slow if is small. 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.

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.

7. *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. https://www.researchgate.net/publication/305820593_Multipath_Load_Balancing_Routing_for_Internet_of_Things