A Mesh Network for Mobile Devices using Bluetooth Low Energy

CONFERENCE PAPER · NOVEMBER 2015

READS
20

7 AUTHORS, INCLUDING:

Hari prabhat Gupta
Indian Institute of Technology (BHU) Varanasi

15 PUBLICATIONS  13 CITATIONS

Available from: Hari prabhat Gupta
Retrieved on: 28 December 2015
A Mesh Network for Mobile Devices using Bluetooth Low Energy

Shruthi Sirur, Praneeth Juturu, Hari Prabhat Gupta, Pramod Reddy Serikar, Yaswanth Kumar Reddy, Sulekha Barak, and Bonggon Kim†,
{shruti.sirur, praneeth.j, harip.gupta, pramod.reddy, yk.reddy, s.barak, bg43.kim}@samsung.com,
Samsung R&D Institute Bangalore, India, †Samsung Electronics, Suwon, South Korea.

Abstract—An important issue of research in wireless networks is to dynamically organize the nodes into a wireless network and route the data from the source to the destination. In most of the existing routing techniques in wireless networks assumes that all nodes are static and do not change their positions till the end of the network. Although it is not a realistic assumption, it eliminates the effects of mobility of nodes in routing the data. In this paper, we present a mesh network for mobile devices using Bluetooth Low Energy (BLE). We propose a weight balancing technique to optimize the communication for routing the data using BLE mesh network. We develop a prototype of BLE mesh network using the Android operating system.

Index Terms—Bluetooth, Mesh, Connectivity, Mobility.

I. INTRODUCTION

A wireless mesh network is a network topology in which each smart device (e.g., smartphone, tablet, wearable, and laptop) transfers the data for the network. All devices in the network cooperate for sharing the data in the network. Mesh networks use flooding and routing techniques for relaying the data [1], [2]. Each device in flooding technique simply broadcasts the data to all attached mesh network devices [3], [4]. In routing technique, each device sends the data to other nearby device in the network.

A Bluetooth Low Energy (BLE) wireless technology, which has been a hallmark feature of Bluetooth Core Specification v4.0 featuring ultra-low power consumption, low latency, and enhanced range. With the adoption of Bluetooth Core Specification v4.1, a BLE device can be a master in one network and can act as a slave in another network, thus allowing the possibility for the formation of a mesh network. In most of the existing routing techniques in wireless networks assumes that all nodes are static and do not change their positions till the end of the network. Although it is not a realistic assumption, it eliminates the effects of mobility of nodes in routing the data.

In this work, we propose a BLE Mesh Network (BMN) for mobile (non-static) devices, which is suitable for forming instant groups among smart devices. The major contributions of our work are as follows:

- We propose a BMN architecture for routing the data among smart devices. We assume that the devices are not static. The BMN architecture uses a routing technique since it consumes lesser power than flooding technique. The BMN architecture does not require any modification in the existing BLE stack.
- We develop an algorithm using Direct Acyclic Graph (DAG) for creation and maintenance of the BMN. To reduce the energy consumption and delay and increase the throughput, it uses the weight balancing technique. The novelty of the propose BMN is that all nodes have nearly equal hop count distances from the root.
- The BMN uses a contingency plan to find the supplementary root device if the main root device leaves (due to mobility of the devices) the BMN.
- We developed a prototype using BLE v4.1 compatible smart devices to check the connectivity of the proposed BMN. The results show that the BMN consumes low power, data loss, and delay.

The rest of the paper is organized as follows: Section II proposes BMN architecture for mobile (non-static) devices. The control and data messages using in BMN are defined in Section III. Section IV proposes the routing creation and optimization technique for the BMN. Section V and Section VI present the results from an experimental evaluation of our work and concludes the paper, respectively.

II. BLE MESH NETWORK (BMN)

The proposed BMN architecture is shown in Fig. 1. We divide the architecture into three layers based on its implementation on Android platform.

A. Android BLE framework Layer

This layer can be divided into the following three components:

- **BLE L2CAP Connection**: A BLE Logical Link Control and Adaptation (L2CAP) channel is a logical connection between two endpoints of peer devices.
- **Scan Manager**: The scan manager handles the advertisement messages on the advertisement channel. It sends the advertisement messages to the mesh service layer.
- **Advertisement Manager**: The advertisement manager transfers the advertisement messages from mesh service layer to android BLE fragmentation layer.

B. Mesh Service Layer

It is the main layer of BMN and developed by Samsung R&D Bangalore. It consists the following two components:

- **BLE Service Library**: It acts as an abstraction to the Android BLE framework. It manages BLE connection establishment, lists, and broadcasts advertisements as shown
The BLE connection component is used to create a dedicated L2CAP connection to transfer the control and data messages. The BLE advertisement listener listens the BLE advertisements from other nearby smart devices. The advertisement messages are parsed by the message processor. The BLE advertisement broadcast sends BLE advertisements to the advertisement manager.

- MESH Service Library: The main components of this library are mesh node, broadcast scheduler, message processor, parser, and a resource manager as shown in Fig. 1. Mesh node is an implementation of a node in the mesh networking topology. Mesh node is identified by the mesh instance Id and an objective function. The objective function is a function of height of the parent. Mesh node uses the message processor and the parser to decode the received advertisement and data messages and act upon mesh specific messages. Message processor is also used to encode the mesh messages before sending or advertising over BLE. The broadcast scheduler is used to schedule these advertisement messages.

C. Application Layer

This can have various mesh applications like chat application, torch light sensor activation, and file sharing in a group. We used a chat application in the proposed prototype of the BMN.

III. INFORMATION EXCHANGE IN BMN

The proposed BMN uses BLE advertisement and data channels for exchanging the information. We use the following messages in BMN.

A. Control messages on Advertisement Channel

The BMN uses the following advertisement messages to exchange information before connection establishment between two BLE devices. Fig. 2 shows the structure of control messages.
the BLE advertisement channels. Next, the user selects any node as a root node and the root node starts broadcasting the DIO. Whenever the selected root node or any connected node in the mesh network receives a DIS message from another node, it broadcasts a DIO message. The DIO message of a node has the rank of the node as shown in Fig. 2. The rank depends on the residual energy, degree, and hop counts of the node. When a node receives DIO messages and if it is not part of the BMN, the node waits for some time for collecting more DIO messages and selects a node which has the lowest rank as a parent. The node establishes BLE connection with the lowest ranking node, by sending the DAO message. This lowest ranking node becomes the primary parent. The other nodes (expect primary parent) from which DIOs were received are connected as alternate parent(s). The alternate parent(s) is useful when primary parent is not available (due to connection break or goes out of communication range) or to find a shortest path between the node and the destination. At the end of this algorithm, all nodes are connected. The pseudo code and example scenario of BMN and complete DAG are shown in Algorithm 1 and Fig. 4, respectively.

Algorithm 1: BMN Formation
1: All nodes broadcast DIS messages.
2: One node acts as a root node.
3: If a BMN node receives DIS message then Broadcast DIO message with its rank.
4: If a non-BMN node receives DIS message, then
   4.1: Wait for a short interval to receive multiple DIOs.
   4.1: Connect and send DAO to the node that has responded with best DIO and acts as the primary parent.
4.2: Other DIOs are stored and after successful connection with each others, store them in alternate parent list.
5: All nodes in vicinity repeat steps 3 and 4 to form the BMN.
6: If a node wants to send data to a destination which is not its child and if its alternate path list is not empty, then
   6.1: The node sends a query message to its primary parent and the nodes in an alternate path list.
   6.2: Nodes that receive the query message check their routing tables for destination and send the depth of the destination node.
6.3: The source node compares all the depths and chooses the minimum as the next hop.

B. BMN Weight balancing and Data transmission

The proposed BMN is suitable for mobile or static BLE devices. We also assume that the any BLE device can act as a root node. In a DAG, a network is not suitable if the distance between the root and leaf nodes is high. In this section, we first propose a procedure for balancing the tree with reference to the root. The novelty of the proposed BMN is that all nodes have nearly equal hop count distances from the root. The pseudo code of BMN weight balancing is shown in Procedure 1. After weight balancing of the BMN, the network is ready for data transfer. Procedure 2 shows the complete process of data transmission.

C. BMN Contingency Plan

To bring in mobility, we draw a contingency plan to find alternate root when the main root node leaves the network. Whenever a new immediate child is added to the root then it chooses one node among its children and assigns the role of alternative root. This choice is made based on some parameters (height of the subtree, number of nodes in the subtree, number of DIOs received from the nodes of the same rank, residual power) in the same order.

Procedure 1: BMN Weight Balancing
1: The root calculates the weighted depth of the tree in current status and also when each of the children becomes a parent.
2: Comparing these values, the root decides if it is the better root or if the root has to be changed to one of its children.
3: If one of the children is selected, then root removes all the entries whose next hop are the elected child and forwards the remaining routing table to that child.
4: The same process is repeated until the root decides that none of its children can act as a better root than itself.

Procedure 2: BMN Data Transmission
1: When the source node wants to send data then
   1.1: The node checks its routing table for destination address.
   1.2: If destination is present then it forwards the data.
   1.3: If not present in routing table then it checks for the best parent (shortest path) and forwards to it.
2: If the destination node is not present in its parent’s routing table then the node forwards it to its parent and the same process is repeated until the message reaches.
3: The root node forwards data message to the next node of the destination and the data reaches the destination.

V. Performance Evaluation

We conducted experiments using the BLE devices to evaluate the effectiveness of the proposed BMN. We implement the BMN on smartphones (Samsung Note 4, Samsung Galaxy S5, and Samsung Galaxy S6) running the Android lollipop system for sharing the text messages. We deployed the BLE devices uniformly at random, inside Samsung R&D Bangalore. Fig. 5 shows the Graphical User Interface (GUI) of the BMN. The following subsections detail the results.
latency is almost same in all experiments for a given value of hop counts. It indicates the stability of the proposed BMN.

TABLE II: Experimental results for latency of BMN.

<table>
<thead>
<tr>
<th>Exp. No.</th>
<th>1-hop</th>
<th>2-hop</th>
<th>3-hop</th>
<th>4-hop</th>
<th>5-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. 1</td>
<td>0.663 s</td>
<td>1.130 s</td>
<td>1.743 s</td>
<td>2.570 s</td>
<td>2.813 s</td>
</tr>
<tr>
<td>Exp. 2</td>
<td>1.320 s</td>
<td>1.610 s</td>
<td>2.370 s</td>
<td>3.100 s</td>
<td>3.250 s</td>
</tr>
<tr>
<td>Exp. 3</td>
<td>1.570 s</td>
<td>2.115 s</td>
<td>2.385 s</td>
<td>3.425 s</td>
<td>4.715 s</td>
</tr>
<tr>
<td>Exp. 4</td>
<td>1.555 s</td>
<td>2.115 s</td>
<td>2.885 s</td>
<td>3.445 s</td>
<td>3.835 s</td>
</tr>
<tr>
<td>Exp. 5</td>
<td>3.205 s</td>
<td>3.440 s</td>
<td>3.995 s</td>
<td>4.560 s</td>
<td>5.290 s</td>
</tr>
</tbody>
</table>

C. Data loss in BMN

Finally, we consider the data loss during the data transmission in the BMN. We consider the five experiments for estimation of the data loss. Here, the data loss is due to the failure of the relay device or outside the communication range of the sender device. In this experiment, each message size is 512 bytes. The maximum hop count between source and destination is 5 hops. Table III illustrates that the maximum data loss is two percent.

TABLE III: Experimental results for data loss.

<table>
<thead>
<tr>
<th>Exp. No.</th>
<th>Data send</th>
<th>Data Received</th>
<th>Data loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. 1</td>
<td>25600 Bytes</td>
<td>25088 Bytes</td>
<td>2%</td>
</tr>
<tr>
<td>Exp. 2</td>
<td>51200 Bytes</td>
<td>50176 Bytes</td>
<td>2%</td>
</tr>
<tr>
<td>Exp. 3</td>
<td>102400 Bytes</td>
<td>101376 Bytes</td>
<td>1%</td>
</tr>
<tr>
<td>Exp. 4</td>
<td>153600 Bytes</td>
<td>153088 Bytes</td>
<td>0.333%</td>
</tr>
<tr>
<td>Exp. 5</td>
<td>204800 Bytes</td>
<td>203776 Bytes</td>
<td>0.5%</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

In this work, we presented a BLE mesh network for mobile devices. We considered the residual energy, hop count, and degree for forming the routing path. The BMN does not require any modification in the existing Android operating system and BLE stack. From the performance evaluation section, we can conclude that the proposed BMN consumes less power and requires low latency. It also shows that the system is scalable for a large number of static and mobile devices and variable size of data messages. It is suitable for many applications such as audio, text, or file transfer.

REFERENCES


