IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 1, 2017, accepted May 17, 2017, date of publication May 24, 2017, date of current version June 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2707556

Topology Configuration and Multihop Routing Protocol for Bluetooth Low Energy Networks

CHANGSU JUNG[1], KYUNGJUN KIM[2], JIHUN SEO[1], BHAGYA NATHALI SILVA[1], AND KIJUN HAN[1]
[1]Department of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea
[2]Information Research Laboratories, Pohang University of Science and Technology, Pohang 37673, South Korea

Corresponding author: Kijun Han (kjhan@knu.ac.kr)

This work was supported in part by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea, through the BK21 Plus Project (SW Human Resource Development Program for Supporting Smart Life) under Grant 21A20131600005, in part by the National Research Foundation of Korea, Ministry of Education, through the Basic Science Research Program, under Grant 2016R1D1A1B03933566, in part by the Ministry of Land, Infrastructure and Transport of Korea Government and Korea Agency for Infrastructure Technology Advancement through the Smart Civil Infrastructure Research Program under Grant 16SCIPA04, and in part by the Institute for Information & Communications Technology Promotion Grant through the Korea Government (MSIP) under Grant 2017-0-00770.

ABSTRACT This paper proposes a new cluster-based on-demand routing protocol to support multihop communication in Bluetooth low energy ad hoc networks. The proposed scheme includes the topology configuration procedure, topology recovery scheme, and on-demand routing protocol. The topology configuration procedure consists of node discovery, piconet configuration, and scatternet formation in a randomly distributed environment. The proposed on-demand routing protocol is designed to minimize the number of route request messages by forwarding them to a master and relay nodes in each cluster during the route request procedure. The performance evaluation shows that our proposed scheme substantially reduces energy consumption, which is the most critical issue on energy constrained networks.

INDEX TERMS Bluetooth low energy, BLE, multihop routing, cluster.

I. INTRODUCTION

Bluetooth Low Energy (BLE) [1] is an ultra-low power communication technology characterized by low power consumption, low cost, low complexity as well as short message transmission [2]–[5]. Thus, BLE technology is suitable for energy constrained applications of small packet transmission such as Internet of things (IoT), Machine-to-Machine (M2M), and sensor networks operated by battery-powered devices [2].

Despite the superiority of low power consumption of BLE, it has a few challenges to be solved for being adopted to IoT and M2M applications. For high volumes of IoT and M2M communications with BLE devices, a multihop routing protocol is required to transmit data at short intervals from a source to a destination out of the radio coverage. Especially, some applications such as environment monitoring or security applications, using BLE technology, require frequent and periodic data transmission to report the change. Moreover, we can consider another scenario requiring a long-distance communication system to transmit simple short messages. Ad-hoc environments caused by disasters such as earthquake, tsunami, and flood require an alternative communication system promptly because network infrastructures have already collapsed due to disasters [19]. In that scenario, BLE multihop communication is a good alternative solution to send short messages because the number of smartphone subscribers worldwide is around 3.9 billion (almost 50 % of the world population) in 2016 [6] and a majority of smartphones, as well as other devices such as tablets and laptops already embed BLE technology.

However, single hop communication and single role support in BLE networks limits the extensive use of BLE technology in various domains. The restriction of single hop communication was not addressed until the release of BLE 4.0. Only a master and slaves can communicate in the same piconet, a one hop network. Moreover, a BLE node should work as a master and slave in different piconets to make inter-piconet communication possible but the dual role operation is not allowed in BLE 4.0 [1].

New features in Bluetooth 4.1 allow BLE devices to work in dual role as both master and slave in different piconets, while permitting to be included in multiple piconets. In addition, a node participating in multiple piconets, also known as a bridge or a relay node, can establish a scatternet which is a

connected piconets. This node switches piconet using time-division multiplexing technique and delivers packets between piconets [1]. Moreover, there is no limitation in terms of the number of slave devices in a piconet. As a result, BLE devices become capable of communicating with other devices in different piconets as well as transmitting data to a device in multihop distance [4]. Although, BLE version 4.1 supports multihop routing, a specific algorithm for it is not included in the specification.

In this paper, we propose a new protocol to support multihop communication among BLE devices in ad-hoc networks. The new protocol establishes clusters throughout the networks and utilizes an on-demand routing approach. The proposed protocol adopts the flooding avoidance mechanism to reduce the number of route request packets (RREQ) causing significant energy consumption during the route discovery procedure. In addition, the performance of the cluster-based on-demand routing protocol (CORP) is evaluated and compared with a traditional on-demand routing protocol and a tree-based proactive routing protocol of Bluetooth sensor networks with respect to the number of RREQ messages, route discovery latency, and energy consumption.

The rest of this paper is organized as follows. Section II elaborates on the related works. The topology configuration of the proposed scheme is demonstrated in Section III. Section IV explains the proposed on-demand routing protocol and section V presents the performance evaluation. Finally, we present the conclusion in Section VI.

II. RELATED WORKS

We reviewed many studies conducted on scatternet structure, multihop topology scheme and routing protocol. However, a majority of the studies have evaluated Bluetooth [5], [7]–[11], [13], [14], [20], [21], and fewer studies have performed on BLE.

Many scatternet formation algorithms [5], [7], [9]–[11], [13]–[15], [20], [21] for Bluetooth networks do not mention the topology maintenance and the overhead of maintaining scatternet topology even though the management scheme should be essentially considered. Most of the researches are focused on establishing scatternet formation and routing protocols.

Guo *et al.* [4] proposed an on-demand scatternet formation and multihop routing protocol for BLE-based wireless sensor networks. Each node operates as an advertiser and scanner alternately to find neighboring nodes. After that, source node sends a route request message to its master to find the destination. If the master does not know the destination in its slave list, then the master starts to deliver the request message to any slaves in its piconet using a breadth-first search. This procedure follows an on-demand manner. Upon receiving all possible routing paths, an originator selects the shortest path.

BlueHRT (Bluetooth Hybrid Ring Tree) [7] is a scatternet formation protocol configuring a hybrid ring tree topology in non-uniformly distributed Bluetooth networks. The proposed protocol benefits from both tree and ring topology. On the one hand, tree-based topology has shorter path and simple routing protocol but traffic bottleneck in bridge nodes is problematic. On the other hand, ring-based topology has higher reliability and simple routing but it no longer has routing information. BlueHRT establishes a ring topology in a highly-populated area whereas, less populated areas form tree topologies. The tree topologies are further connected to the ring topology. Two topologies are connected with slave/slave bridges or master/slave bridges.

Liu *et al.* [11] introduced an on-demand scatternet route structure for multihop data transmission in Bluetooth networks. A route is created by the traffic request and only includes the nodes on the traffic path. The scatternet establishes the route dynamically and releases at the completion of the communication. This study adopts a combination of single role and double role of a node to construct the scatternet route structure. Double role nodes are only used in critical areas to bridge piconets.

In addition, a scatternet adopts a routing protocol to establish a route path for transmitting data from a source to a destination. In many ad-hoc networks, Ad-hoc On-demand Distance Vector (AODV) protocol [12] has been adopted as a routing protocol, though it causes network degradation results from the flooding mechanism. Therefore, many researches were carried out to alleviate the flooded RREQ packets.

Enhanced Bluetree [13] was proposed to improve network reliability and reduce the length of route paths of a traditional tree topology in Bluetooth sensor networks. Bluetooth nodes are randomly distributed in multihop distance and have low mobility. The proposed protocol consists of two phases' scatternet formation algorithms, namely a tree-shaped topology and a mesh-shaped topology respectively. In the first phase, a root node connects neighboring slave nodes and the connected slave nodes change their roles as masters to connect other downstream slave nodes. This procedure continues until the leaf nodes are connected. In the second phase, a mesh topology is constructed by establishing more connections among nodes to distribute traffic in the root node and shorten the route paths. However, Enhanced Bluetree does not provide a topology management scheme which is required to maintain a mesh topology and to recover from a failure.

An energy-aware routing protocol [15] adopts an on-demand scatternet formation approach in Bluetooth sensor networks. If a source node wants to send traffic, it starts inquiring to find neighboring nodes. When a neighboring node receives a route request message, it saves the source and the previous node's information for the route reply. Moreover, it avoids flooding loops like AODV protocol. When a source selects a relay node for route request, the remaining current level is considered. The destination initiates scatternet formation using a page message, upon receiving a route request. Depending on the remaining current level, the intermediate nodes determine whether to forward or discard the route request message. This scheme avoids energy depletion at critical nodes in the routing path.

Kum *et al.* [16] proposed an on-demand routing protocol with directional flooding, namely AODV-DF, to limit the delivery of RREQ packets in the wireless mesh network. A gateway broadcasts HELLO messages including its address and a hop count value as zero. When one hop distance node receives a HELLO message, the node increases the hop count value in the HELLO message by one and updates its own hop count value. After that, the node re-broadcasts the HELLO message to neighboring nodes. With this procedure, every node knows its own hop count value from the gateway. If a source node wants to find a path to a destination, a source node sends a RREQ message including its hop count value. When a neighboring node receives a RREQ message, the hop count value in a RREQ message is compared with the node's hop count value. If the node's hop count value is smaller than the value in a RREQ, then the node updates a RREQ's hop count value with its hop count and re-broadcasts a RREQ message. When the hop count value of a RREQ message is smaller than the node's hop count value, the node discards a RREQ message because a RREQ packet was delivered by a shorter distance node from a gateway. With this approach, the proposed scheme reduces the number of RREQ messages throughout the network.

Enhanced AODV routing protocol [8] was proposed to avoid RREQ packet flooding in Bluetooth scatternet. The proposed scheme considers the battery power and the traffic intensity for discovering a route path. A RREQ message includes average traffic intensity, predicted power and expected reply time to use as route metrics. When an intermediate node receives RREQ messages, it updates three route metrics with its own values. When a destination receives multiple RREQ messages, it selects the lowest cost path using the best route selection algorithm. In addition, the proposed scheme delivers RREQ messages only to bridge nodes in a piconet to reduce the number of RREQ packets. However, the proposed protocol does not describe how to configure bridge nodes and the whole network topology.

III. TOPOLOGY CONFIGURATION SCHEME

In this section, we describe how each BLE node can discover neighboring nodes, select a master node in a piconet and configure clusters using neighbor count values for constructing entire scatternet topology in the BLE mesh networks. The new topology configuration protocol consists of four phases i.e. node discovery, piconet configuration, scatternet configuration and topology recovery scheme. In this paper, we assume that the BLE nodes are distributed in 10×10 grids randomly and the radio coverage is limited to one hop distance.

A. NODE DISCOVERY PROCEDURE

During the node discovery procedure, BLE nodes collect information about neighboring nodes' ID and neighbor count by exchanging advertisement messages from other nodes. Whenever, a BLE node receives an advertising message from a neighbor node, the BLE node updates the neighbor count value of the node's neighbor list. These values are used to select a master node after the node discovery procedure.

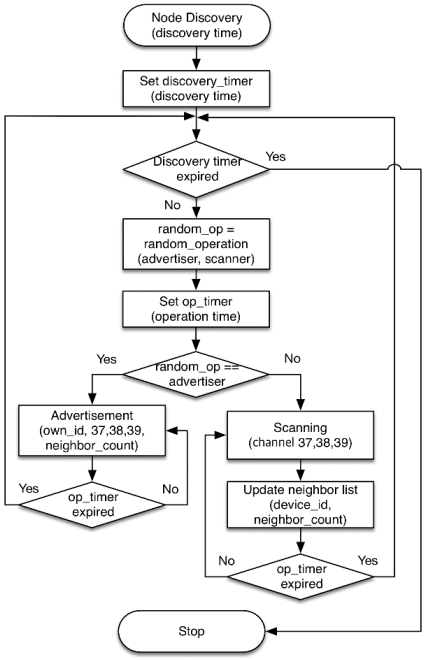


FIGURE 1. Flowchart of the node discovery procedure.

Initially, each node operates as an advertiser or a scanner randomly. Then, it switches the role after a certain duration (operation time) in the given discovery periods (discovery time). In this paper, the operation time for random operation of BLE nodes is fixed at 30ms to satisfy the maximum advertisement interval of BLE 4.1 specification. If a node runs as an advertiser, it broadcasts advertising messages including own id and current neighbor count. At the beginning, the neighbor count is set to zero. When a node operates as a scanner, it receives advertising messages from its neighboring nodes. Sequentially, the scanning node increases its neighbor count value and updates its neighbor list. The updated neighbor list and neighbor count values are utilized to broadcast advertising messages when the node transforms into an advertiser. Figure 1 describes the node discovery procedure.

When a node receives ADV_IND message from other nodes, it updates its neighbor list. The neighbor list consists of six fields, i.e. device id, neighbor count, first piconet master id, first role, second piconet master id, second role. In this paper, the number of piconet where a BLE node can join is restricted to two piconets because of reducing piconet switching overload. Therefore, first and second piconet master ids represent master nodes' ID where a node joins. First and second roles in the neighbor list describe a BLE node's responsibilities in each piconet as a master, slave or relay node.

B. PICONET CONFIGURATION PROCEDURE

At the completion of node discovery process, each node knows the node with the maximum neighbor count value

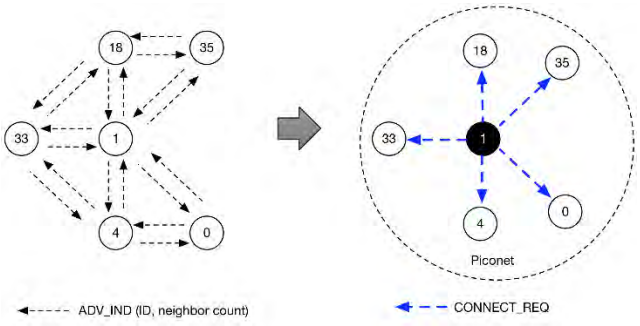


FIGURE 2. Piconet configuration procedure.

within the radio range. The node with the maximum neighbor count operates as the master and unicasts CONNECT_REQ messages to its neighbors from the lowest device ID's node and establishes a piconet, while other nodes act as slaves in the piconet. If multiple nodes have the same maximum neighbor count values, the lowest device id's node is selected as a master. The piconet configuration procedure is carried out in a distributed manner. For example, as depicted in figure 2, node 1 is selected as the master since it has the maximum number of neighbors among other adjacent nodes. Node 1 sends CONNECT_REQ messages to its neighboring nodes and constructs a piconet.

The master node checks whether a neighboring node is already a member of another piconet before sending a CONNECT_REQ message. If a neighboring node is included in another piconet, a master does not send CONNECT_REQ message in this phase. The membership of a node is known through an advertisement message, which includes its master and role as shown in table 1. Slave nodes operate as advertisers and only the master node runs as a scanner and an advertiser interchangeably to manage a piconet and to determine changes of other nodes such as failure.

TABLE 1. Neighbor list table.

Device ID	Neighbor count	First piconet master ID	First role	Second piconet master ID	Second role
6 bytes	2 bytes	6 bytes	1 byte	6 bytes	1 byte

Algorithm 1: Pseudo-code of master initiative relay node configuration

```
1   neighbors = search_neighbor(master)
2   for n_i in neighbors
3     if ((n_i is other piconet member) and (n_i's first role == SLAVE))
4       relay_candidate_list.append(n_i)
5     end if
6   end for
7   for candidate[i] in relay_candidate_list
8     for candidate[i+1] in relay_candidate_list
9       if (the same piconet member(candidate[i], candidate[j]) ==
10        TRUE)
11          relay node = (lowest_dev_id(candidate[i], candidate[j])
12      else
13        relay node = candidate[i]
14      end if
15    if (master's neighbor list has relay node == FALSE)
16      CONNECT_REQ (master, relay node)
17      relay node's second piconet master id = master's device id
18      relay node's first role = RELAY
19      relay node's second role = RELAY
20      insert master's neighbor list (relay node)
21      update relay node's neighbor list (relay_node)
22    end if
23  end for
```

FIGURE 3. Algorithm of master initiative relay node configuration.

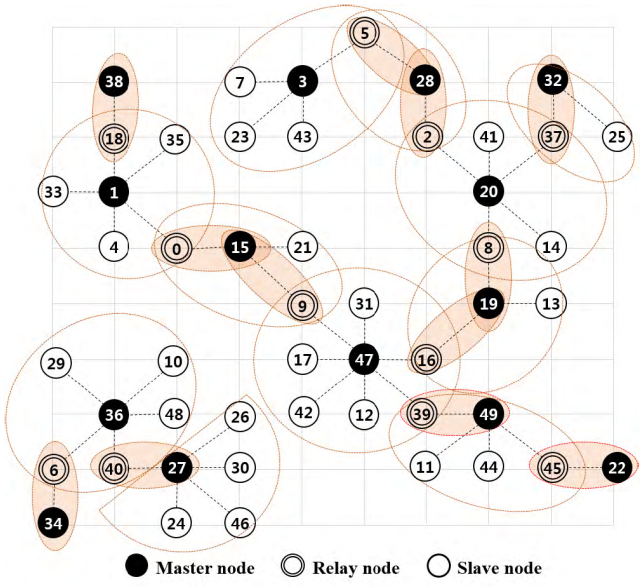


FIGURE 4. Phase 1: master initiative relay node configuration.

C. SCATTERNET FORMATION

After the piconet configuration, the scatternet formation procedure takes place to connect isolated piconets and nodes, which are not involved in any piconets. Finding relay nodes included in multiple piconets is essential to connect isolated piconets. In this paper, a relay node is included only in two piconets to reduce piconet switching overload. The scatternet formation procedure consists of two phases i.e., master initiative and slave initiative relay node configuration.

1) PHASE 1: MASTER INITIATIVE RELAY NODE CONFIGURATION

In the first phase, master nodes select relay nodes among neighboring nodes that are not included in their own piconets to connect adjacent piconets. Master nodes search for neighboring nodes and find candidates for a relay node in figure 3 (line 1∼4) . When a master finds multiple candidates for a relay node in the same adjacent piconet, it selects the lowest device id's node and sends a CONNECT_REQ message to the node to establish a connection (line 11∼16). For example, node 27 has two candidates (node 40, 48) for a relay node. Node 27 selects node 40 because of the lowest device id selection mechanism as shown in figure 4. After that, a master inserts the relay node information to its neighbor list as shown in figure 3 (line 20).

Meanwhile, the relay node updates its second piconet master ID and second role on its neighbor list (line 17∼21).

Algorithm 2: Pseudo-code of slave initiative relay node configuration

```
1   neighbors = search_neighbor(slave)
2   for ni in neighbors
3     if ((ni's first piconet master != slave's first piconet master) and
4       (ni's second piconet master == NONE))
5       if (slave's neighbor list has ni == FALSE)
6         relay_candidate_list.append(ni)
7       end if
8     end if
9   end for
10  relay = lowest_device_id (relay_candidate_list)
11  master = lowest_device_id (relay, slave)
12  relay's second piconet master = master's device id
13  if (slave's device id < relay's device id)
14    CONNECT_REQ (slave, relay)
15    slave's first role = RELAY
16    slave's second role = MASTER
17    relay's first role = RELAY
18    relay's second role = SLAVE
19  else
20    CONNECT_REQ (relay, slave)
21    relay's first role = RELAY
22    relay's second role = MASTER
23    slave's first role = RELAY
24    slave's second role = SLAVE
25  end if
26  slave.insert_neighbor_list(relay)
27  update_neighbor_list(relay)
```

FIGURE 5. Algorithm of slave initiative relay node configuration.

Then the relay node delivers updated information to the first piconet master via advertisement messages, which includes the second piconet master id and the second role as shown in table 1.

The proposed scheme limits the number of relay nodes between piconets to one in order to reduce piconet switching overload. Furthermore, a master node is excluded from being a relay node to avoid packet loss and switching overhead.

Most adjacent piconets are connected after the master initiative relay node configuration but a few piconets remain unconnected because some neighboring piconets are located more than one hop distance away from an adjacent master node. Therefore, the second relay node configuration is required.

2) PHASE 2: SLAVE INITIATIVE RELAY NODE CONFIGURATION

After phase 1, a master informs its slave nodes of finishing the first phase. Slave nodes initiate the second phase's relay node configuration procedure. During the second phase, a slave looks for non-relay nodes included in a different piconet within the radio coverage and the slave node checks their second piconet master ID. If non-relay nodes do not have the second master ID, it means they are included in the one piconet. Then the slave node inserts non-relay nodes into its relay candidate list in figure 5 (line 1∼9). When a slave finds a few candidates of a relay node, the lowest device id's node is selected as a relay node (line 10). After that, the lowest device id's node sends CONNECT_REQ message to make a piconet acting as a master (line 11∼14). Sequentially, it operates as a master in the newly constructed piconet. A slave and a relay node update their first and second role as relay.

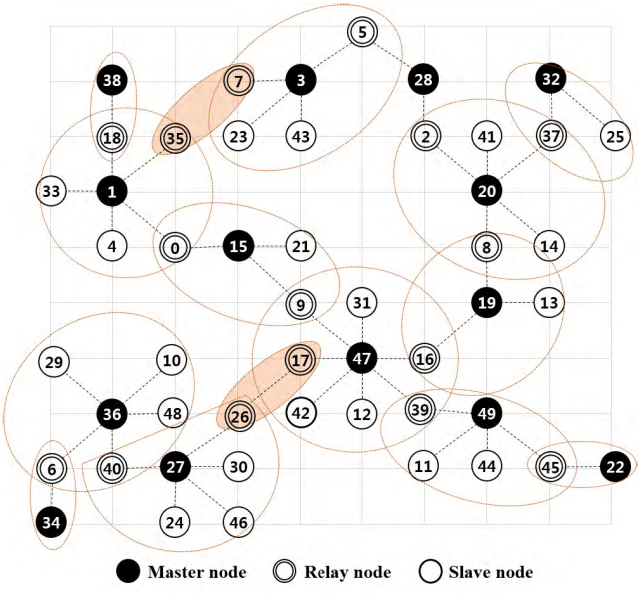


FIGURE 6. Phase 2: slave initiative relay node configuration.

The lowest device id's node also updates its second role as master (line 14∼22). Finally, they update their neighbor list with other node's information to configure another piconet (line 26∼27).

At the completion of the two phases' scatternet formation, the network topology construction is completed and the network is connected by relay nodes as shown in figure 6.

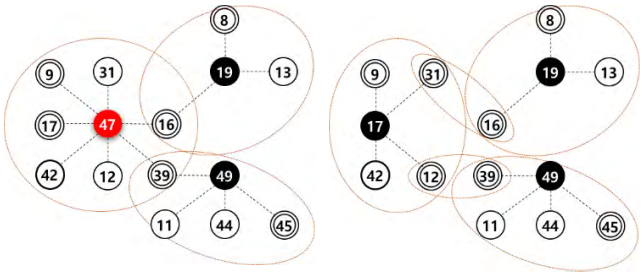
D. TOPOLOGY MAINTENANCE

A node failure in a piconet causes serious problems such as packet losses and a piconet disconnection. Especially a relay or a master node failure results in more critical issues i.e. the network disconnection between piconets and the breakdown of the whole network. In order to recover from the failure, we propose two recovery schemes such as a master and a relay node recovery. According to Bluetooth Specification 4.1, BLE devices in the connected mode can perform advertising, scanning or discovery procedures while maintaining connections in the piconet [1]. Therefore, an assumption is made that every node in a piconet broadcasts advertisement messages periodically after the network topology configuration procedure, while remaining in the connected mode.

This advertisement scheme consumes more energy in the BLE networks but this approach makes an important role to manage the membership of a piconet and the network topology changes.

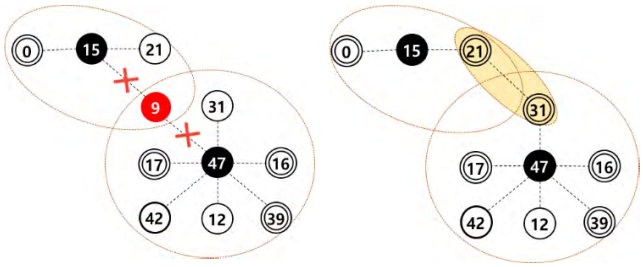
1) MASTER NODE RECOVERY SCHEME

All piconet members broadcast advertisement messages periodically including neighbor counts and other information as mentioned before. Moreover, they recognize other nodes' membership and normal operation. If a master node stops its operation due to a failure, slave nodes can detect the



(a) Master failure (b) Master recovery

FIGURE 7. Master recovery procedure. (a) master failure. (b) master recovery.



(a) Relay node failure (b) Relay node recovery

FIGURE 8. Relay node recovery procedure. (a) relay node failure. (b) relay node recovery.

master's problem through the absence of advertisement. In this case, the piconet master reselection procedure starts to reconfigure the piconet. Slave nodes scan their neighbor list and identify the node with the maximum neighbor count values as previously stated in the piconet configuration. A node with the maximum neighbor sends CONNECT_REQ messages to its neighbors to recover the piconet. Subsequently, other nodes delete the previous master's information from their neighbor list and update new master's information. The two phases' scatternet configuration procedures are accomplished at the completion of piconet construction. These procedures find new relay nodes to connect other piconets as shown in figure 7. For example, master 47 has a failure, node 17 will be a master because it has the maximum number of neighbors among other nodes. Node 17 sends CONNECT_REQ messages to neighbors and reconstruct a piconet. However, other nodes cannot be a member of the piconet because nodes 16 and 39 are placed out of radio coverage. For this reason, the scatternet configuration procedures are required to connect to other piconets. Nodes 31 and 12 perform the second phase of the scatternet configuration procedure to find relay nodes around them. After completing the second phase, nodes 31 and 12 change their roles from slave to relay and update their roles and master information accordingly in their neighbor list table.

2) RELAY NODE RECOVERY SCHEME

As cited earlier, the responsibility of a relay node is to interconnect piconets and deliver packets. Thus, a failure of relay nodes leads to serious consequences such as packet losses and scatternet disconnection. Therefore, the detection and recovery of relay nodes is vital than other nodes. In this section, we describe how to detect a relay node's failure and to find an alternative relay node.

If piconet members do not receive advertising messages from a relay node for a given period, it implies a failure. In this case, the scatternet configuration phase 2 is executed to find an alternative relay node. Figure 8 illustrates the relay node recovery procedure.

For instance, when a relay node 9 has a failure in figure 8, two piconets' members recognize that a relay node has a problem with no advertisement message from the relay node.

Nodes 21 and 31 simultaneously try to find an alternative relay node around them to reconnect two piconets.

TABLE 2. Neighbor list after recovery.

Device ID	Neighbor count	First piconet master ID	First role	Second piconet master ID	Second role
21	2	15	Relay	21	Master
31	4	47	Relay	21	Slave

Nodes 21 and 31 already know their existence from their neighbor list. Node 21 sends CONNECT_REQ message to node 31 because it has the lowest device id. After that, node 21 becomes a master node and node 31 operates as a slave in the newly constructed piconet. However, node 21 and 31 already belong to other piconets and they already operate as slave so they update their first roles as relay and second roles as master and slave respectively as shown in table 2. Node 21 operates as slave whenever it receives data from its master but node 21 runs as master when it delivers data to node 31.

IV. ON-DEMAND ROUTING PROTOCOL

To support multihop routing in BLE networks, we consider an on-demand routing protocol which is widely accepted in mobile ad-hoc network (MANET). Due to on-demand route discovery characteristics, a routing path is constructed whenever a route request occurs. In addition, on-demand routing protocol reduces the consumption of energy by minimizing network connectivity. For this reason, on-demand routing protocol is a suitable candidate to support multihop communication in the energy constrained networks.

We propose the cluster-based on-demand routing protocol in BLE networks in this section. Figure 9 shows a route request and reply message format used in this paper. These packet formats are adjusted to data channel PDU in BLE protocol and there are no changes in the BLE protocol.

A. ROUTE DISCOVERY PROCEDURE

The conventional AODV protocol assumes that a route request message (RREQ) is broadcasted using BLE advertisement channels (37, 38, and 39). The route reply message (RREP) is delivered along the shortest route path via a unicast connection between nodes. A source node broadcasts a RREQ message to discover a routing path when it has data to be transferred to a specific destination.

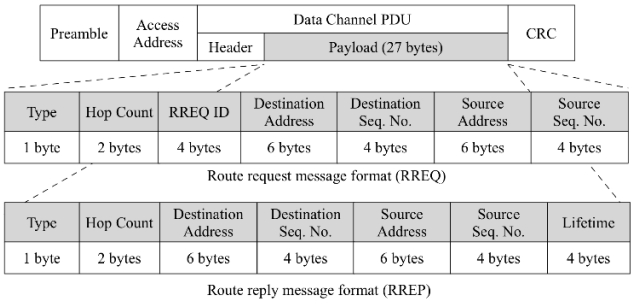


FIGURE 9. Route discovery message format.

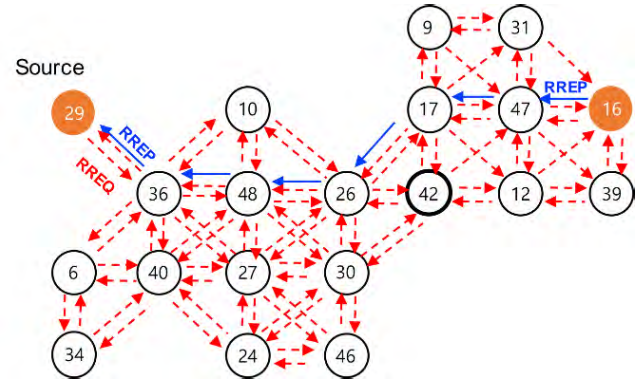


FIGURE 10. Route discovery procedure of AODV.

Upon receiving a RREQ message, the neighbor node checks the destination address of the RREQ message. If the destination address of the RREQ message does not match the node's address, then the node broadcasts RREQ message again to its neighbors as described in figure 10.

However, in the proposed scheme, a source node sends a RREQ message to its master to request for the route path to a destination. First, the master searches for the destination in its neighbor list. If the destination is not a member of the piconet, the master forwards RREQ messages to its relay nodes, which are connected to other piconets. Relay nodes deliver the RREQ message to its other master. The other master searches the destination in its neighbor list. This procedure continues until the destination is found. When a RREQ message reaches the destination piconet, the master of that piconet selects the shortest path among multiple paths and replies to the previous node with a RREP message along the shortest path as described in figure 11. RREQ messages are only delivered to masters and relay nodes in the networks so this scheme can significantly reduce the number of RREQ messages. Furthermore, the minimized number of RREQ messages substantially reduces the energy consumption compared to the conventional AODV protocol.

When a master receives RREQ message from its member, the master looks up its neighbor list first. If a destination is found in the same piconet, a source node sends data to the destination via the master. If a destination is not a member of its piconet, a master performs breadth-first search (BFS)

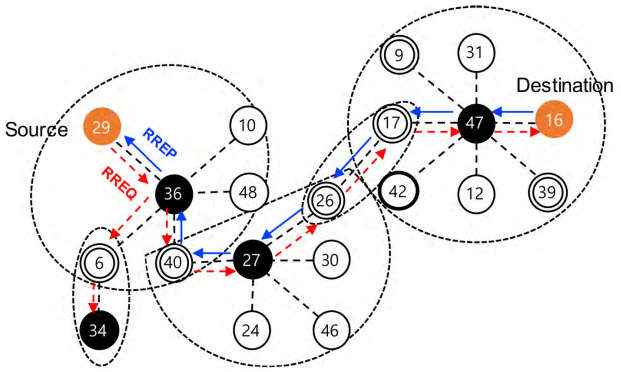


FIGURE 11. Route discovery procedure of the proposed scheme.

Algorithm 3: Pseudo-code of route request procedure

```
1  SEND_RREQ (source's piconet master)
2  loop (destination is found == FALSE)
3     lookup_destination (neighbor list of a master)
4  if (destination is a member of a master == TRUE)
5     destination is found = TRUE
6     ROUTE REQUEST DONE
7     exit
8  else
9     relay_node_list = BREADTH_FIRST_SEARCH (master)
10    ENQUEUE (relay_node_list)
11    if (queue != empty)
12       relay node = DEQUEUE (relay_node_list)
13       SEND_RREQ (relay node)
14       master = relay node's piconet master
15       FORWARD_RREQ (master)
16    else
17       destination is found = FALSE
18       exit
19    end if
20    end if
21 END LOOP
```

FIGURE 12. Algorithm of route request procedure.

to find its relay node. When a master finds a relay node, RREQ message is forwarded to its relay nodes. A relay node forwards RREQ message to another master and the master searches its neighbor list. This procedure continues until the destination is found in a master's piconet. A master manages its neighbor list including relay and slave nodes in its piconet so that a master can forward a RREQ message to its relay nodes. Figure 12 describes the algorithm of route request procedure of the cluster-based routing protocol.

B. ROUTE DISCOVERY LATENCY

In this section, we describe the proposed route discovery approach and illustrate the delay of route discovery. Table 3 summarizes the list of parameters to be calculated for route discovery latency and topology configuration.

In the proposed scheme, intra-piconet and inter-piconet connections are already established during the network topology configuration procedure, therefore it occupies data

TABLE 3. List of route discovery parameters.

Notation	Meaning
T_{RREQ}	The transmission time of RREQ message
T_{RREP}	The transmission time of RREP message
T_{ADV}	Advertisement period (37, 38, 39 channels)
T_{IFS}	Inter-frame space
T_{ROLE_SW}	Role switch time
T_{CONN_SETUP}	Connection setup time
T_{POST_CONN}	Mandatory delay after connection request
L_{ROUTE}	Route path length
N_{RREQ}	The number of RREQ messages
$T_{CLUSTER\_REQ}$	Route request time of CORP
$T_{CLUSTER\_REP}$	Route reply time of CORP
$T_{CLUSTER\_DISCOVERY}$	Total route discovery latency of CORP
$T_{NODE\_DISCOVERY}$	Node discovery time of CORP
$T_{PICONET}$	Piconet configuration time of the CORP
$T_{SCATTERNET}$	Scatternet configuration time of CORP
$T_{TOPOLOGY}$	Topology configuration time of CORP
N_{SLAVE}	The number of slave nodes per piconet
N_{RELAY}	The number of relay node in the network
I_{ROUTE_REQ}	The current consumption during route request
I_{ROUTE_REP}	The current consumption during route reply

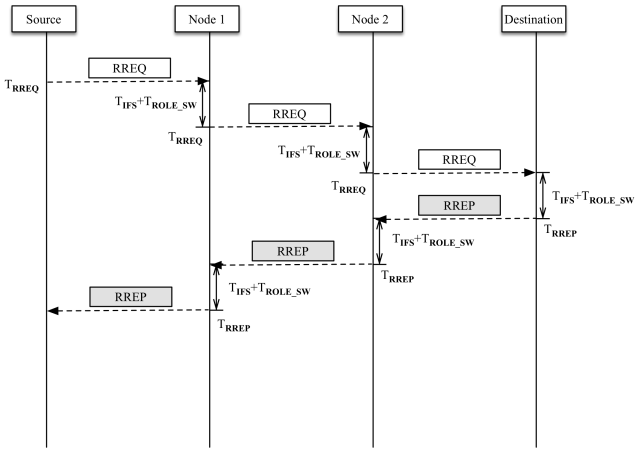
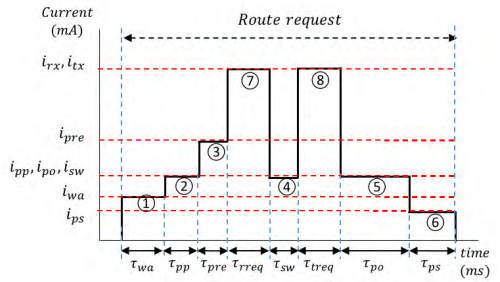


FIGURE 13. Route discovery message flow.

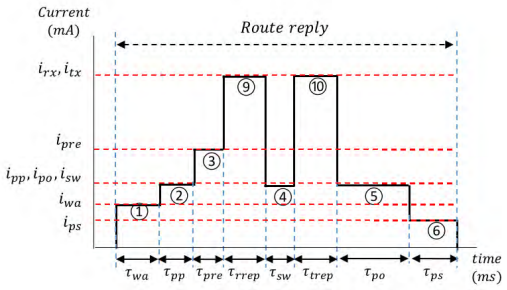
channels to deliver RREQ and RREP messages. The transmission time is the same for both RREQ (T_{RREQ}) and RREP (T_{RREP}), since both messages use the same packet length as shown in figure 9. When forwarding RREQ and RREP messages to an intermediate node, an inter-frame space (T_{IFS}) and a role switch delay (T_{ROLE_SW}) are required as shown in figure 13. The RREQ transmission time (T_{RREQ}) and RREP transmission time (T_{RREP}) are defined by dividing the packet size with BLE data rate (1Mbit/s).

C. ENERGY CONSUMPTION

The energy consumption model used in this paper is referred as the measurement of a CC2541 BLE device of Texas Instruments [17]. We calculate the current consumption and time of each state during route discovery procedure based on the measurement.



(a) Route request procedure



(b) Route reply procedure

FIGURE 14. Current waveform of CORP. (a) route request procedure. (b) route reply procedure.

TABLE 4. The analysis of time and current consumption of CORP.

Route Request / Route Reply			Time		Current		Average Current (mA)
	State	Phase	Notation	Value (ms)	Notation	Value (mA)	
Common	①	Wake-up	τ_{wa}	0.4	i_{wa}	6	2.4
	②	Pre-processing	τ_{pp}	0.315	i_{pp}	7.4	2.331
	③	Pre-Tx/Rx	τ_{pre}	0.08	i_{pre}	11	0.88
	④	Tx-to-Rx	τ_{sw}	0.105	i_{sw}	7.4	0.777
	⑤	Post Processing	τ_{po}	1.28	i_{po}	7.4	9.472
	⑥	Pre-sleep	τ_{ps}	0.16	i_{ps}	4.1	0.656
Route Request	⑦	Rx (RREQ)	τ_{rreq}	0.296	i_{rreq}	17.5	5.18
	⑧	Tx (RREQ)	τ_{treq}	0.296	i_{treq}	17.5	5.18
Route Reply	⑨	Rx (RREP)	τ_{rrep}	0.296	i_{rrep}	17.5	5.18
	⑩	Tx (RREP)	τ_{trep}	0.296	i_{trep}	17.5	5.18

The procedure is the same for both route request and route reply. Accordingly, the current waveform is similar for route request and route reply as shown in figure 14.

The values for time, current, and average consumption are presented in table 4.

V. PERFORMANCE EVALUATION

This section provides the environment of simulation and evaluates the performance of the proposed routing protocol, Enhanced Bluetree, and AODV with respect to the average number of RREQ messages, route discovery latency, and energy consumptions during route discovery procedure. Enhanced Bluetree constructs a rooted spanning tree and a mesh-shaped topology using the two phases' scatternet configuration procedure. This protocol adopts a

proactive approach. After topology configuration, data packets are transmitted including route discovery procedure. Thus, the topology configuration scheme is comparable to CORP.

As the BLE standard is published very recently, hence, no primary simulator exists in the current literature. Especially, multihop communication in BLE networks is just mentioned in the standard specification without any specific information. With this reason, we implemented our simulator to analyze and evaluate our proposed scheme. In fact, energy consumption was carried out by mathematical analysis and simulator.

A few general assumptions were applied in the proposed scheme i.e. all BLE nodes are placed randomly within 10×10 units, the radio range is one hop distance, and frequent mobility of BLE nodes is not considerable. As a result, a BLE node does not recognize other devices placed in two hop distances. Moreover, we focus on some applications which are required to transmit data periodically at short intervals. In the simulation, the number of BLE nodes changed to the range of 50, 60, 70, 80, and 90 accordingly and 100 data were collected for each number of BLE node. For the evaluation of performance, we have implemented a simulator in Python 3.5 and have used NetworkX Python software package [18] to analyze the network topology.

A. AVERAGE NUMBER OF RREQ MESSAGES

We have simulated the average number of RREQ messages from a source to a destination from all routing paths. The conventional AODV scheme uses a flooding mechanism to deliver RREQ messages and each node broadcasts RREQ messages from one to three times during the advertisement period (T_{ADV}). If the number of BLE node is N, the average total number of RREQ messages equals to (N-1) $\times$ 2. On the other hand, the proposed scheme conveys RREQ messages only to master nodes and relay nodes in the network. Henceforth, the total number of RREQ packets is significantly less than the conventional scheme.

Enhanced Bluetree constructs a partially mesh-shaped topology based on a spanning tree. Therefore, RREQ messages are delivered to the connected neighboring nodes and upstream root nodes to find a destination, thus, causing more RREQ messages delivered.

Figure 15 illustrates the gradual growth of RREQ messages with respect to the increase of BLE devices in the proposed scheme and enhanced Bluetree. Enhance Bluetree shows more RREQ messages by approximately 50% than the proposed scheme due to the tree characteristic of enhanced Bluetree.

On the contrary, the conventional AODV scheme shows a significant growth in the number of RREQ messages.

B. TOPOLOGY CONFIGURATION TIME

The topology configuration is carried out in a distributed manner so the configuration time is mainly affected by the number of nodes per piconet and the number of relay nodes. Considering aforementioned factors, the topology

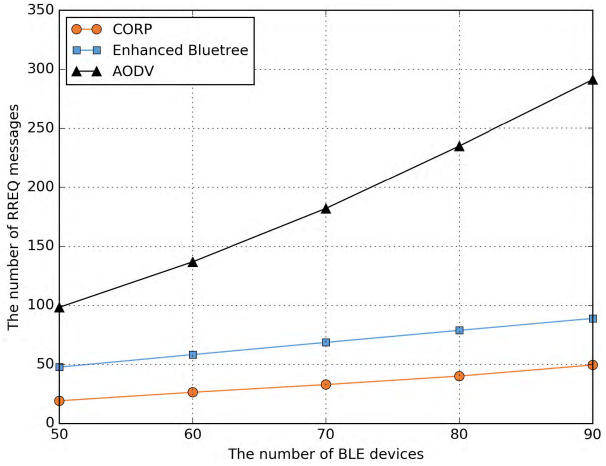


FIGURE 15. Average number of RREQ message.

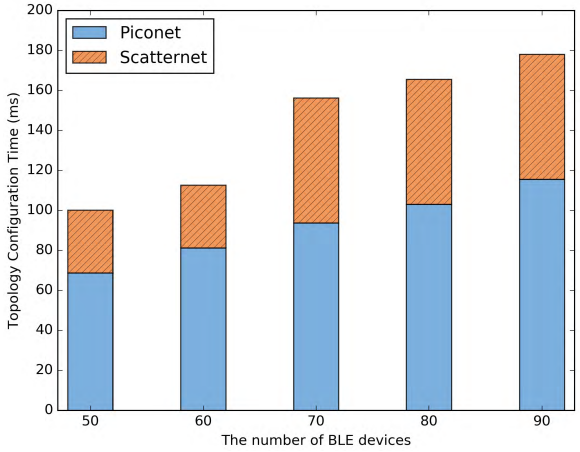


FIGURE 16. Topology configuration time.

configuration time is computed by equation (1). The configuration period of both the piconet configuration and the topology configuration increased steadily with the number of BLE devices as shown in figure 16. Nevertheless, there is not much difference with regard to the scatternet configuration since the average number of relay nodes per piconet has the range of 1 or 2.

$$\begin{aligned} T_{PICONET} &= N_{SLAVE} \times (T_{CONN\_SETUP} + T_{ROLE\_SW}) \\ T_{SCATTERNET} &= N_{REALY} \times (T_{CONN\_SETUP} + T_{ROLE\_SW}) \\ T_{TOPOLOGY} &= T_{PICONET} + T_{SCATTERNET} \end{aligned} \tag{1}$$

C. ROUTE DISCOVERY LATENCY

Route discovery latency is measured by the time elapsed between sending a route request message and receiving a route reply message at the originator. The average route discovery latency of two schemes were calculated using equations (2) and (3) mentioned below in this section. Another simulation obtained the node discovery time ($T_{NODE\_DISCOVERY}$) of the CORP scheme and it is fixed

TABLE 5. Details of route discovery latency.

Notation	Time (ms)
T_{ADV}	30
T_{IFS}	0.15
T_{ROLE_SW}	1.25
T_{RREP}	0.296
T_{RREQ}	0.296
T_{CONN_SETUP}	30
T_{POST_CONN}	1.25
$T_{NODE\_DISCOVERY}$	1200

at 1200ms. The number of successful discoveries for each node was around 10, with the period of 1200ms. For the topology configuration time, we assume that a successful connection between two nodes is established within BLE advertisement period ($T_{ADV} = 30$ms) and the role switch time is 1.25ms. Table 5 shows the details of latency of two schemes.

The route discovery latency of the conventional AODV scheme is defined as equation (2). The route request time (T_{AODV_REQ}) has a strong correlation with the number of RREQ message.

$$\begin{aligned} T_{AODV\_REQ} &= (T_{ADV} + T_{IFS} + T_{ROLE\_SW}) \times N_{RREQ} \\ T_{AODV\_REP} &= (T_{CONN\_SETUP} + T_{IFS} + T_{ROLE\_SW} \\ &\quad + T_{RREP}) \times L_{ROUTE} \\ T_{AODV\_DISCOVERY} &= (T_{AODV\_REQ} + T_{AODV\_REP}) \end{aligned} \tag{2}$$

The route discovery latency of the CORP scheme is defined as below. Even though the node discovery and topology configuration do not occur in every route discovery of the proposed scheme, the durations $T_{NODE\_DISCOVERY}$ and $T_{TOPOLOGY}$ were considered for more accuracy in latency calculation. Consequently, the total route discovery latency is calculated as below. The route request time ($T_{CLUSTER\_REQ}$) is dependent on the number of RREQ messages (N_{RREQ}) and route reply time ($T_{CLUSTER\_REP}$) is closely related to the route path length (L_{ROUTE}). The topology configuration consists of the node discovery time ($T_{NODE\_DISCOVERY}$), piconet configuration time ($T_{PICONET}$) and the two phases' scatternet formation ($T_{SCATTERNET}$). Route discovery latencies of CORP are computed using equations (1) and (3).

$$\begin{aligned} T_{CLUSTER\_REQ} &= (T_{RREQ} + T_{IFS} + T_{ROLE\_SW}) \times N_{RREQ} \\ T_{CLUSTER\_REP} &= (T_{RREP} + T_{IFS} + T_{ROLE\_SW}) \times L_{ROUTE} \\ T_{CLUSTER\_DISCOVERY} &\\ &= (T_{CLUSTER\_REQ} + T_{CLUSTER\_REP}) \\ &\quad + T_{TOPOLOGY} \end{aligned} \tag{3}$$

Enhance Bluetree considers the hop delay, path length, switching delay of node, and the number of relay nodes to compute the discovery latency. We use equation (3) to calculate the route discovery latency of Enhanced Bluetree.

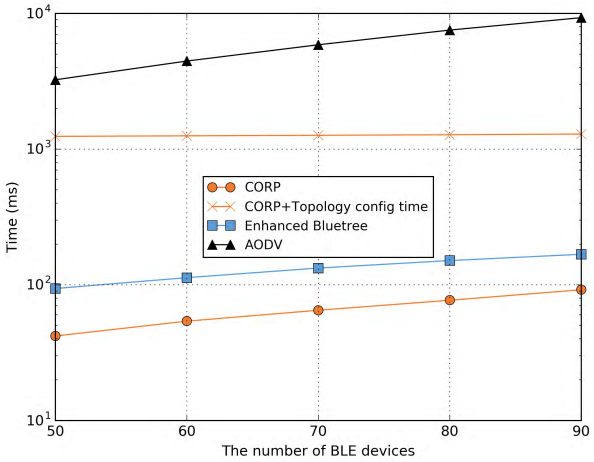


FIGURE 17. The comparison of route discovery latency.

Figure 17 includes graphs for the AODV protocol, Enhanced Bluetree, and the CORP including the node discovery and the topology configuration period, and the CORP without the node discovery and the topology configuration period.

The result shows that the proposed protocol remains steady, while the AODV protocol follows a steep rise as the number of BLE nodes increases. This increment is caused by the increasing number of RREQ message and additional connection setup procedure in the conventional AODV protocol during RREP transmission from a destination to a source. If the network topology does not change frequently, the route discovery latency of the proposed scheme will have the smallest value. Enhanced Bluetree has longer latencies than CORP as Enhanced Bluetree has longer route paths due to the tree topology and more RREQ packets as shown in figure 15.

D. ENERGY CONSUMPTION

We calculate the energy consumption during route discovery procedure by referring to the measurement [17]. The conventional AODV route scheme adopts a broadcast approach for delivering RREQ messages and unicast connections for RREP messages from the destination to the originator. Therefore, additional energy consumption is required to establish connections for route reply messages. On the contrary, the proposed scheme utilizes data channels for delivering RREQ and RREP messages so the total energy consumption becomes less than the conventional AODV scheme.

Even though Enhanced Bluetree is a proactive approach, it requires more RREQ messages for route discovery due to the rooted spanning tree and partial mesh topology.

The current consumption of the route discovery procedure is computed as below and figure 18 represents the current consumption of the three protocols. Enhanced Bluetree consumes less energy than AODV but it shows more energy consumption than CORP by 50%.

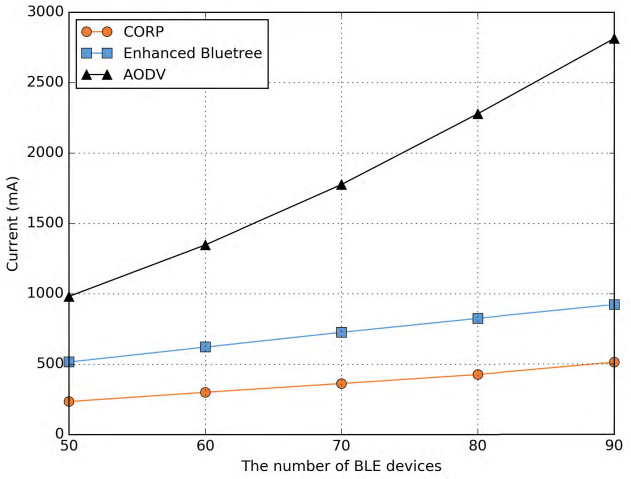


FIGURE 18. The comparison of energy consumption.

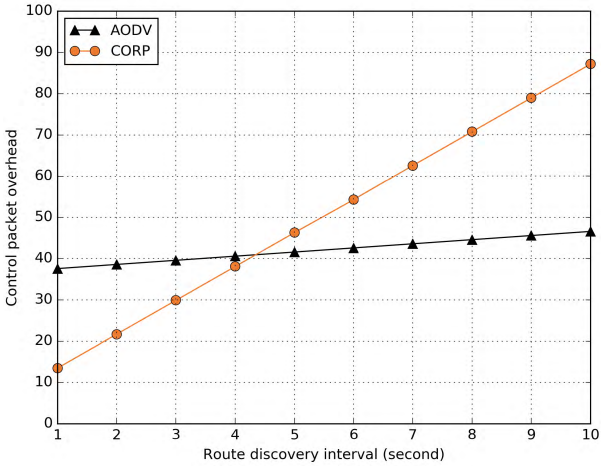


FIGURE 19. Control packet overhead.

E. CONTROL PACKET OVERHEAD

The control packet overhead is measured by the ratio of the number of control packets over the number of data packets delivered for the given period. In this paper, we simulate the control packet overhead for 10 minutes and the interval of route discovery and data delivery changes from 1 second to 10 seconds. During the simulation time, data packets of 20 bytes are delivered from a source to a destination randomly selected after route discovery procedure.

The control packets of CORP include route request, route reply messages for route discovery and periodic advertisement messages for topology maintenance with one second interval.

On the contrary, the control packets of AODV consists of route request and reply messages during the route discovery procedure and HELLO messages with one second interval for the active link connectivity.

However, Enhanced Bluetree, in their research, does not mention how to maintain its topology so we cannot measure the control packet overhead.

As illustrated in figure 19, CORP has a lower overhead than AODV when the route discovery interval is shorter than five seconds. Even though CORP broadcasts advertisement messages to keep its topology, the frequent route discovery in AODV requires more control packets such as route request and reply messages when a source and a destination are selected randomly. The higher overhead of CORP is caused by the periodic advertisement messages for topology management at longer discovery interval.

With this result, the proposed routing protocol is suitable for frequent and periodic data transmission environment.

VI. CONCLUSION

We have proposed the cluster-based on-demand routing protocol for supporting multihop communication in BLE networks. The new protocol establishes clusters throughout the networks and adopts an on-demand routing approach to reduce energy consumption in the energy constrained networks. The proposed routing protocol reduces the number of route request messages by means of delivering the messages only to a master and relay nodes in the cluster. Consequently, the minimized number of route request messages results in less energy utilization and shorter route discovery latency than the conventional on-demand routing protocol. Moreover, we also introduce a recovery scheme for the network failure. The simulation results show that our proposed protocol provides better performances in terms of route discovery latency, energy consumption and the number of route request messages in the power constrained BLE networks.

The main contribution of this paper is to support long distance communication in BLE networks. The proposed multihop routing protocol can be adopted in smart home, smart building system, and various IoT applications as well as the failure locations of network infrastructure such as disaster areas.

REFERENCES

[1] *Bluetooth Core Specification, Version 4.1*, SIG, Newington, NH, USA, Dec. 2013.
[2] K.-H. Chang, "Bluetooth: A viable solution for IoT? [Industry Perspectives]," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 6–7, Dec. 2014.
[3] J. Nieminen *et al.*, "Networking solutions for connecting Bluetooth low energy enabled machines to the Internet of Things," *IEEE Netw.*, vol. 28, no. 6, pp. 83–90, Dec. 2014.
[4] Z. Guo, I. G. Harris, L. Tsaur, and X. Chen, "An on-demand scatternet formation and multi-hop routing protocol for BLE-based wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Sep. 2015, pp. 1590–1595.
[5] C.-M. Yu and Y.-B. Yu, "Joint layer-based formation and self-routing algorithm for Bluetooth multihop networks," *IEEE Syst. J.*, to be published.
[6] (Nov. 2016). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf
[7] S. Sharafeddine, I. Al-Kassem, and Z. Dawy, "A scatternet formation algorithm for Bluetooth networks with a non-uniform distribution of devices," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 644–656, 2012.
[8] O. Al-Jarrah and O. Megdadi, "Enhanced AODV routing protocol for Bluetooth scatternet," *Comput. Electr. Eng.*, vol. 35, no. 1, pp. 197–208, 2009.
[9] G. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees scatternet formation to enable Bluetooth-based ad hoc networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 1. Sep. 2001, pp. 273–277.

[10] C. Petrioli, S. Basagni, and M. Chlamtac, "Configuring BlueStars: Multihop scatternet formation for Bluetooth networks," *IEEE Trans. Comput.*, vol. 52, no. 6, pp. 779–790, Jun. 2003.
[11] Y. Liu, M. J. Lee, and T. N. Saadawi, "A Bluetooth scatternet-route structure for multihop ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 229–239, Feb. 2003.
[12] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*, document RFC 3561, Jul. 2003, pp. 1–13. [Online]. Available: http://www.rfc-editor.org/info/rfc3561
[13] C. M. Yu and J. H. Lin, "Enhanced bluetree: A mesh topology approach forming Bluetooth scatternet," *IET Wireless Sensor Syst.*, vol. 2, no. 4, pp. 409–415, Dec. 2012.
[14] C. Petrioli, S. Basagni, and I. Chlamtac, "BlueMesh: Degree-constrained multi-hop scatternet formation for Bluetooth networks," *J. Mobile Netw. Appl.*, vol. 9, no. 1, pp. 33–47, 2004.
[15] X. Zhang and G. F. Riley, "Energy-aware on-demand scatternet formation and routing for Bluetooth-based wireless sensor networks," *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 126–133, Jul. 2005.
[16] D. W. Kum, A. N. Le, Y. Z. Cho, C. K. Toh, and I. S. Lee, "An efficient on-demand routing approach with directional flooding for wireless mesh networks," *J. Commun. Netw.*, vol. 12, no. 1, pp. 67–73, Feb. 2010.
[17] S. Kamath and J. Lindh, "Measuring Bluetooth low energy power consumption," Texas Instruments, Dallas, TX, USA, Appl. Note AN092. [Online]. Available: http://www.ti.com.cn/cn/lit/an/swra347a/swra347a.pdf
[18] *NetworkX Website*, accessed on Oct. 10, 2016. [Online]. Available: http://networkx.github.io/index.html
[19] A. M. E. Ejmaa, S. Subramaniam, and Z. A. Zukarnain, "Neighbor-based dynamic connectivity factor routing protocol for mobile ad hoc network," *IEEE Access*, vol. 4, pp. 8053–8064, Nov. 2016.
[20] C. Yu and Y. Yu, "Reconfigurable algorithm for Bluetooth sensor networks," *IEEE Sensors J.*, vol. 14, no. 10, pp. 3506–3507, Oct. 2014.
[21] A. Jedda and H. T. Mouftah, "Forming MS-free and outdegree-limited Bluetooth scatternets in pessimistic environments," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 538–550, Jun. 2015.

CHANGSU JUNG received the M.S. degree in communication systems from the Royal Institute of Technology, Stockholm, Sweden, in 2012. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea. His research interests include Bluetooth low energy, architecture of IoT, and routing protocols in short range communication.

KYUNGJUN KIM received the B.S. degree in computer engineering from Kyungil University, Deagu, South Korea, in 1996, and the M.S. and Ph.D. degrees in computer engineering from Kyungpook National University, Deagu, in 1999 and 2005, respectively. He is currently a Research Associate Professor with Information Research Laboratories, Pohang University of Science and Technology, Pohang, South Korea. His current research theme is related to emergency service and application in for smart push notification services on mobile application system.

JIHUN SEO received the M.S. degree in computer engineering from the School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. His area of expertise includes Bluetooth low energy, and routing protocols in ad hoc and wireless networks.

BHAGYA NATHALI SILVA received the B.S. and M.S. degrees in information technology from the Sri Lanka Institute of Information Technology, Colombo, in 2011. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea. Her area of expertise includes architecture designing for Internet of Things, machine-to machine communication, cyber physical systems, and communication protocols.

KIJUN HAN received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 1979, the M.S. degree in electrical engineering from KAIST, South Korea, in 1981, and the M.S. and Ph.D. degrees in computer engineering from the University of Arizona in 1985 and 1987, respectively. Since 1988, he has been a Professor with the School of Computer Science and Engineering, Kyungpook National University, South Korea.