Fan et al. EURASIP Journal on Wireless Communications and Networking (2020) 2020:161
https://doi.org/10.1186/s13638-020-01768-4

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Bluetooth-based device-to-device routing protocol for self-organized mobile-phone mesh network

Aihua Fan[1], Zhiqiang Tang[2], Weiwei Wu[2], Yu Tang[2*] and Di Lin[2]

* Correspondence: yutang@uestc.edu.cn
[2]University of Electronic Science and Technology of China, Chengdu 610054, China
Full list of author information is available at the end of the article

Abstract

The bluetooth technology provides a point-to-point communication interface to mobile phones and other electronic devices, which can be used to form a self-organized ad hoc mesh network consists of mobile phones. In such a self-organized mesh network, the data packets are not transferred by traditional telecommunication backbone network but through bluetooth-based hop-by-hop routing among mesh nodes in the network. Such an ad hoc mesh network may find a wide span of application scenarios where access to backbone network infrastructure is not available, yet it needs an efficient routing protocol to provide a timely transmission mechanism for data packets to reach their destination even under situations when network topology is in constant changes due to mobile node's movement. In the bluetooth-based device-to-device routing protocol (BDRP) presented in this paper, we propose several key designs as processed data table, node exclusion by packet and source control routing to provide an end-to-end packet transmission solution in a dynamically changing topology with constraints on network bandwidth, battery power, and storage space on a mobile device. The processed data table and node exclusion by packet mechanisms contribute to eliminating a great number of non-useful duplicate packets in routing path, and the source control routing algorithm implemented with a cross-layer packet retransmission process provides a fast and low-overhead routing policy and an assurance to packet reaching destination. In the simulation experiment conducted BDRP demonstrates its effectiveness and efficiency in eliminating undesirable duplicate packets and adaptability to constant topological changes. The performance comparison experiment indicates that the BDRP outperforms the optimized link state routing (OLSR) protocol in packet arrival rate, particularly under situations when mobile nodes are in fast moving mode, yet manages to maintain the same level of transmission delay as OLSR's even with the packet retransmission process. The work presented in this paper provides a key technology for the applications of Bluetooth mobile phone mesh network in a larger geographic area and a broader application domain.

Keywords: Mesh network, Routing protocol, Packet transmission, Bluetooth, Mobile phone

1 Introduction

Today's wireless users are enjoying a wide span of mobile applications including text, voice, video, instant message, and social network on their smart mobile phones, most of which are equipped with a bluetooth device that allows two mobile phones directly communicate with each other within a limited range. The applications and services on mobile phones are provided through traditional backbone networks such as GPRS, CDMA, WCDMA, or TD-SCDMA by telecommunication providers, i.e., a mobile phone has to be connected and registered with one of above backbone networks prior to customer usage. However, under the circumstances when natural disasters such as earthquake, flood, hurricane, or storm disable the backbone network, or network communication is not available due to geographic limitations or poor work conditions, providing wireless communication presents a big challenge to traditional wireless solutions. Satellite network can be considered as an alternative solution yet comes with high cost and limited transmission bandwidth. Wireless ad-hoc network is considered to be a promising cost-effective solution [1–3], in which constantly moving devices constitute a self-organized mesh network without a centralized registering node. Each node in such a network serves as a transmission end and a mediate routing point as well. For such a self-organized network topology, a dynamic routing mechanism adapted to the topology plays a key role in supporting data transmission in this type of networks.

Bluetooth emerged as a low-energy wireless technology for data exchange over short distance in 1994 and since then has been widely used in consumer electronics and mobile device based applications [4–6]. Bluetooth 4.0 is capable of transferring data at a rate of 150 kbps and to a range of 100 m, powered by button cell that may last for 1 year [7, 8]. New bluetooth 5.0 can reach a distance of 300 m at a transmission speed of 2 Mbps [9], and support multi-point connectivity that is suitable for mesh networks. As a consequence, bluetooth loaded mobile phones are considered to be appropriate candidates to establish a large-scale mesh network (Fig. 1) that can provide data connection among a large number of mobile users without backbone network services. Designing routing algorithm in the bluetooth-based mobile phone mesh network turns out to be a big challenge for two reasons: first, each node is in a constantly moving state, causing a constantly changing network topology for any routing paradigm to work with; second, the very limited computing power and storage space on a mobile phone prevent any global lookup table design at a center node that may simplify the routing structure. Timely responsiveness is another critical issue since human beings are the end users and fast response to user operations is a key in deciding if these networks are acceptable or not.

Targeting on ad hoc mesh networks consist of bluetooth mobile phones, in this paper, we propose a bluetooth-based device-to-device routing protocol (BDRP) that can support many-to-many data links across a multi-hop path and provide data transmission control to upper applications, with the following design goals:

- Short transmission delay
- High packet arriving rate
- Adapt to moving node location
- Decentralized routing structure

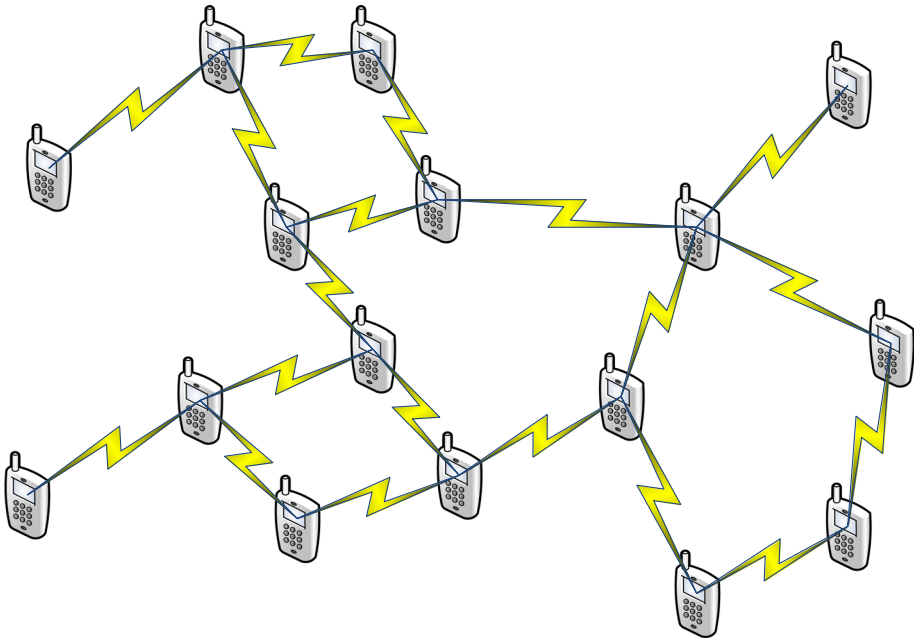


Fig. 1 Bluetooth-equipped large-scale mobile-phone mesh network

The rest of the paper is organized as follows: chapter II describes the routing architecture and two major types of routing approach for ad hoc wireless mesh network; chapter III presents the design of bluetooth-based device-to-device routing protocol; in chapter IV, a series of simulation experiments are performed to verify the effectiveness of the proposed BDRP and to compare its performance with an existing approach; chapter V summarizes what we find in this work and the open issues for future researches.

2 Related work

With bluetooth technology providing data exchange interface between any two adjacent mobile phones within a limited range, a routing protocol is demanded to explore an end-to-end multi-hop transmission from source to destination. The routing protocol for mobile ad hoc network is also required to provide data transmission control function such as ACK/NACK packet or retransmission process in case a packet is lost in transit or does not arrive prior to its deadline. Existing researches on routing protocols can be mainly categorized into three categories, namely, proactive routing [10–12], reactive routing [13–15], and hybrid routing [16, 17]. Proactive routing is also called the table-driven routing [18], by which each node in the mesh needs to maintain a routing look-up table. When network topology changes, each node in the topology needs to be informed of the changes and updates its routing table accordingly. The major advantage of the proactive routing is that a routing path from source to destination is mostly ready when a data packet needs to be sent, which provides a certain level of quality of service and a shorter delivery time since it does not need to spend time to figure out a routing path prior to transmission. Yet, the proactive routing approach requires each node store routing information of any other nodes, and any changes at one node need to multicast to entire topology to keep routing table at each node up to date, which

greatly consumes transmission bandwidth and battery energy. So proactive routing is not considered a scalable solution to large-scale mesh networks.

As a representative of proactive routing protocols, OLSR (optimized link state routing) [19] uses message flooding manner to multicast topological changes among the mesh nodes. To avoid a poor performance of flooding multicast, OLSR uses the MPR (multipoint relay) mechanism [20] to eliminate duplicate transmissions in topological information exchanging process. Compared to traditional flood multicasting in which a

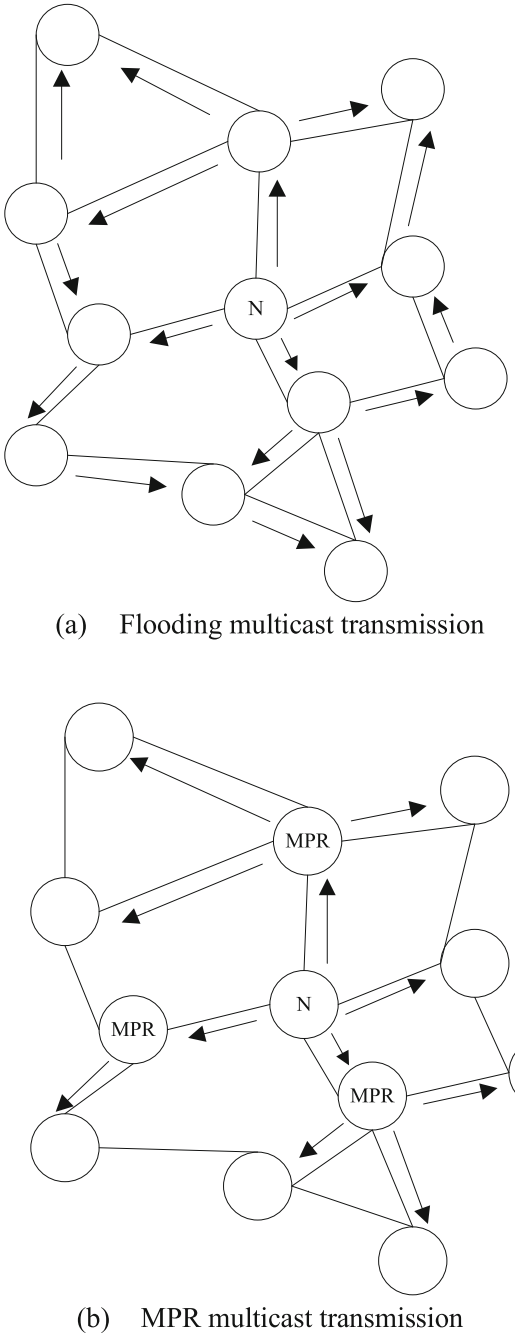


Fig. 2 Topological control data broadcast in OLSR. **a** Flooding multicast transmission. **b** MPR multicast transmission

node (node N in Fig. 2a) forwards every message it receives to all of its adjacent nodes, the MPR transmission requires (i) each node selects a fraction of adjacent nodes as its MPR nodes. One node may be picked up as an MPR node by a group of other nodes, and this group is called the MPR selector set for this node; (ii) when receiving a message, each node only relays the messages to the nodes belong to its MPR selector set. In Fig. 2b, among the four adjacent nodes of node N, only three MPR nodes belonging to the MPR set of node N are picked up as the next hop to forward the messages.

Reactive routing, also called on-demand routing or source routing, takes a different approach in which the end-to-end transmission path between any two nodes is not established or updated until one node starts its transmission and needs to acquire a routing path to the destination. Specifically, reactive routing protocols such as DSR (dynamic source routing) [21] and AODV (ad hoc on-demand distance vector routing) [15, 22] implement three steps to meet the routing requirements initiated by the packet sender, i.e., route discovery, route maintenance, and route removal. The route discovery process triggers construction of dynamic routing tables at intermediate nodes and AODV provides a route maintenance procedure to refresh the routing table at each node when topology change or node failure happens. Yet, the route discovery process significantly increases the transmission delay of reactive routing since every end-to-end message delivery is added a route discovery overhead and, when the topology scale becomes large or in dramatic changes, this routing control overhead grows fast.

Above observations indicate that the proactive routing can provide a timely data transmission in ad hoc wireless mesh networks but suffers high overhead and low bandwidth utilization, while the reactive routing uses network bandwidth more efficiently by implementing an on-demand routing process, yet yields a longer delay time in the design presented in this paper, we use techniques such as processed data table, node exclusion by packet, and source control routing to filter out duplicate data packets, to exclude certain adjacent nodes from packet forwarding, and to implement intermediate fast routing by embedding source information into data packets. These designs greatly help BDRP to achieve the goals of higher packet arrival rate, lower delay time, and reduced network load, which are used as the major performance metrics in the simulation experiments to assess the feasibility of proposed routing protocol in an ad hoc mesh network environment.

3 Methods/experimental

3.1 Cross-layer protocol model

Traditional network protocol OSI model [23] consists of five or seven layers, including data link layer (MAC) , network routing layer (IP protocol), and transmission control layer (TCP protocol), which together provide a reliable end-to-end data transmission across a routing path. In the situation of ad hoc wireless mesh network constrained by limited computing power and storage space on a mobile phone, we suggest a simplified cross-layer protocol model consists of three layers (Fig. 3): the bluetooth transmitting layer to provide the MAC function between two adjacent mobile phones, the BDRP layer to provide a cross-layer function of packet routing and transmission control, and the upper application layer to support end user applications. Each mobile node (phone) joining the mesh network has to install a complete set of above 3-layer protocols and a

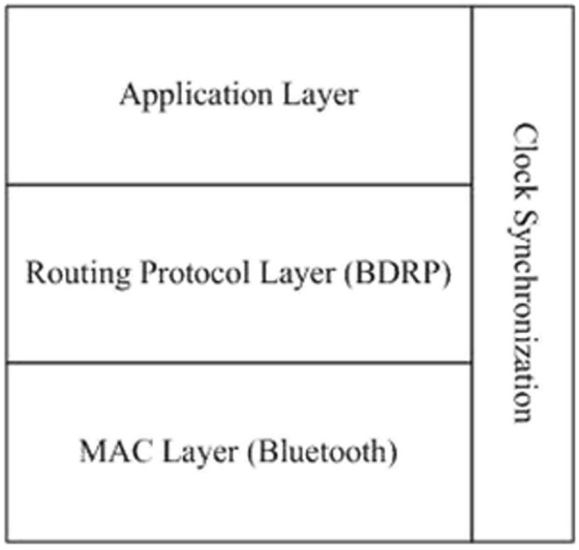


Fig. 3 Cross-layer protocol model for BDRP

mobile application that provides the functions of network registration, hand-shake, and topology management. To simplify the problem, we assume all mobile nodes in the network are working under the same global clock counter, i.e., clock synchronization is assumed for all nodes at each protocol layer.

The data flow of the BDRP-based application is shown in Fig. 4, in which, when a message is transferred from source to destination, only at the two ends (node A and node C) that all three protocol layers are involved in the delivery. At the intermediate node (node B), only the lower two layers are used to forward data packets to next hop.

3.2 Protocol design

3.2.1 Neighbor table and flood multicasting

Neighbor table at a node contains the list of adjacent nodes that can be reached by one-hop transmission, i.e., can be directly reached by a node's bluetooth device. The

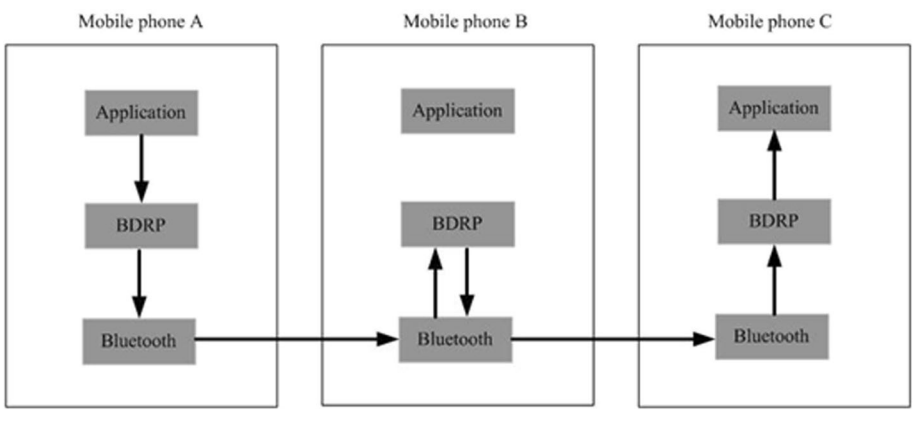


Fig. 4 Data flow process for BDRP

bluetooth device uses a periodically multicasting manner to scan for adjacent bluetooth devices and, if a new node found, it tries to establish a connection to the new node and adds to its own neighbor table. At each intermediate node on a routing path, the neighbor table provides a list of candidate nodes to be selected as the next hop to forward data packet to. Since the storage space on a mobile node is very limited, we intend to trim down the size of neighbor table as much as we can. This is particularly important when the neighbor table size expands fast, along with the increase of network topology scale. Yet multicasting simply generates too many duplicate packets in the network, consuming network bandwidth quickly for no gain and causing unnecessary high workload. Therefore, reducing size of neighbor table at each node and eliminating duplicate packets in network become initial routing design considerations.

3.2.2 Processed data table

Under multicasting mode, setting TTL (time to live) [24] for each transmitted packet is an effective way to control the number of packets circulating in the network. In the BDRP design, the TTL value is set in number of hops when a packet is launched at the source node. Though TTL may prevent a packet be infinitely forwarded in network, yet may not avoid duplicate packets being received and processed at nodes, which is a waste of computing resource. If the first time a packet arrives at a node it is processed and recorded in a processed data table (PDT) that is stored at the node, the next time when the duplicate copy of same packet arrives, the node can check the PDT and drop the duplicate packet if it is verified processed. Table 1 shows the data entry stored in the PDT, and a packet can be uniquely identified by using a combination of the first three attributes of an entry in the table.

Even with a simple PDT design, the fast growth of network topology size may lead to an extra-large table that cannot be efficiently processed. So an entry elimination algorithm is needed to remove those out-of-date records from the table and keep the PDT to a reasonable size. The PDT entry elimination algorithm we proposed here is based on the TTL value, i.e., when a record entry associated with a processed packet is added to the PDT, the expiration time of the packet is also stored in the table as an attribute of the record. The PDT managing thread keeps periodically check the PDT, and, if a record is found reach its expiration time, it will be removed from the table. In this entry elimination algorithm, the TTL of a record is calculated as

$$\mathrm{T}(h) = \mathrm{h} \times (t + p) \tag{3.1}$$

where h is the TTL value of packet set at the source, t the average transmission time between two adjacent nodes, p the average processing time for one packet, and $T(h)$ the TTL value for a record entry in PDT.

Table 1 Processed data table design

Field name	Type	Description
Originating_node_addr	Char string	Source node address
Destination_node_addr	Char string	Destination node address
Data_serial_number	Integer	Serial number for a packet assigned by source node
TTL_value	Float	Expiration time, i.e., TTL value for the PDT entry

Obviously, for most data packets, $T(h)$ is the maximum live time their PDT entry needs to exist. Beyond this value, most packets shall already be eliminated from network due to the packet's TTL settings. Above $T(h)$ design ensures the PDT be trimmed to an accepted table size and avoid removing PDT entries while associated packets are still alive.

Traffic congestion scenarios shall also be considered in the TTL/PDT design. Assuming that a packet arrives at a node, with its record entry already existed in the node's PDT, i.e., the packet is a processed one for this node, thus shall be dropped upon the PDT check. Due to a high workload or traffic jam, this arrived packet may wait in the line for processing for a long time, during which the record entry of this packet in the node's PDT was eliminated since it reached expiration time. In such a scenario, the PDT check would fail and an already processed packet will be undesirably forwarded again or sent to upper application for re-processing. To deal with it, we suggest at each node to add two checks, i.e., the packet's TTL check and the $T(h)$ check, prior to the PDT scan. This will ensure all packets that expire its TTL or $T(h)$ values will not be processed even if their record entries in PDT have been removed from the table.

3.3 Node exclusion by packet

When topology size reaches a certain scale, particularly for those dense networks, quite often, we observe such scenarios: (i) two adjacent nodes share some other adjacent nodes (in Fig. 5, node A and node F share two other adjacent nodes D and G); (ii) two nodes are not adjacent to each other but share some adjacent nodes (node A and node

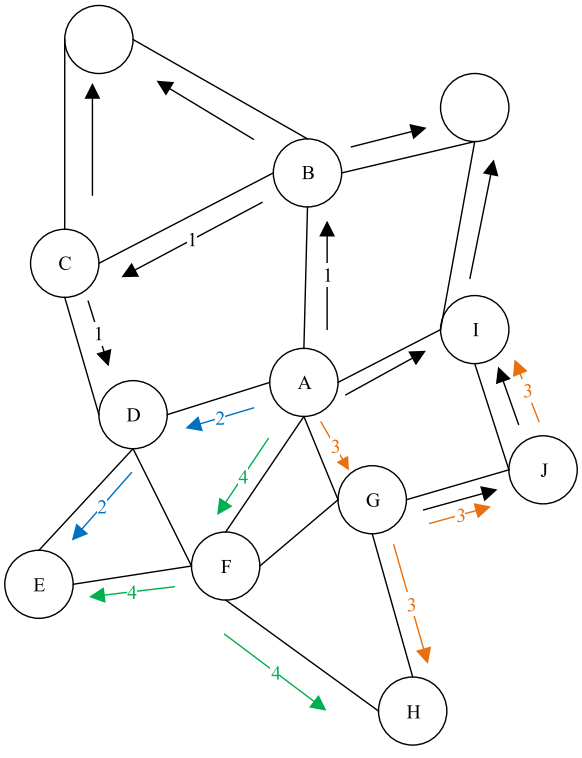


Fig. 5 Shared adjacent nodes in a dense network

H share two adjacent nodes F and G). In either case, when node A broadcasts packets to its adjacent nodes, those packets will go through nodes D, F, G, and then be forwarded to node H. When node F receives the packets, assuming it excludes node A for it is the source node, but still forwards the packets to its adjacent nodes, including D and G. So nodes D and G will undesirably receive the packets from node A at least twice. Same thing happens to node H, in which, the shared adjacent nodes F and G will receive packets launched by node A at least twice. Put these together, we observe a fact that the shared adjacent nodes will receive quite a large number of duplicate packets from various source nodes. Even the PDT design may prevent those duplicate packets be re-processed or re-forwarded, already they may circulate in the network for a while which noticeably adds workload burden and consumes network bandwidth.

To eliminate duplicate packet circulation caused by shared adjacent nodes, the neighbor table can be re-designed or the table-based technique can be applied to prevent duplicate packet forwarding. However, the table-based implementation not only pays the cost of maintaining a growing table, but has the drawback that it does not work when the two nodes (nodes A and H in Fig. 5) are not adjacent but share a set of adjacent nodes (nodes F and G) as well, since A or H would not appear in other's neighbor table. Thus, we suggest a packet-based approach that adds two fields, i.e., Processed_node_list and Processed_node_count, into the packet data segment as shown in Table 2.

With the list of processed nodes embedded in the packet's data segment, a node that receiving the packet may exclude those nodes in the arriving packet's list from the node's neighbor table; hence, preventing duplicate packets be forwarded to any node that has already processed the packet. Since the list of processed nodes contains the packet going through nodes and their one-hop forwarding nodes (also their adjacent nodes), which eventually contains the shared adjacent nodes described above, so most of duplicate packet circulation can be filtered out. Considering the size of a data packet cannot grow infinitely, we place a bound PN_Max_Num on the number of nodes the list can contain and, when a new node needs to be added to a list that reaches its bound, a refreshing algorithm will be deployed to keep the list up to date. Observing the fact that the bluetooth technology on majority of android phones in market can support a maximum number of 15 connection points, we suggest to set the PN_Max_Num value to 30, which covers a full set of adjacent nodes and leaves enough vacancy for others.

3.4 Source backtrack mechanism

When the upper application sends an image or audio/video file to destination even with a compressed version, the file size may still reach a range of hundreds of KB to a few MBs. If sent in one or a few data packets, these packets are too large to be efficiently transmitted in an ad hoc wireless mesh network. A common technique is to partition a

Table 2 Two fields added to packet data

Field name	Range	Description
Processed_node_list	A node set of up to PN_Max_Num nodes	A node list containing experiencing nodes in routing path and its one-hop forwarding nodes
Processed_node_count	0 ~ PN_Max_Num	Number of nodes in current processed node list

large-size file into hundreds or thousands of small-size packets, each assigned with a packet serial number. The destination node receives all these packets and assembly them to the original file by using packet serial numbers. One scenario may be observed; in a short period, a large number of data packets originated from same source are routed toward same destination. So, we propose a source control mechanism, together with designs of tailored retransmission and revised PDT, to provide an efficient and fast routing algorithm to ensure most of packets reach destination in time.

3.5 Tailored retransmission

Under the constraint on network bandwidth and battery power, we propose a cross-layer approach that implements packet retransmission control into network routing protocol, described as below:

i) when the destination receives the packet, an ACK packet is replied to the origin, yet without retransmission assurance on the ACK. The TTL of ACK is set in the following equation to ensure most of ACKs may reach the source node

$$\mathrm{L}(h) = \left\lceil \frac{h}{3} \right\rceil + h \tag{3.2}$$

where h is the original packet's TTL set at the source node and $L(h)$ is the ACK's TTL set by the destination node. The portion of $h/3$ in $L(h)$ is added in the consideration that there is no retransmission on ACK

ii) if the source node does not receive returned ACK within timeout, the source will assume the packet is lost in transmit and then command a retransmission on the same packet
iii) at most the source node will retransmit the same packet for three times. If no ACK returned after the 3rd retransmission, the source node will stop retransmission and report a failure to upper application.

Different from traditional TCP/IP implementation in which the packet expiration time is calculated on the measurement of network jam condition and workload level, which are not applicable in ad hoc wireless mesh network, the BDRP sets the packet expiration time or the time deadline for an ACK based on the TTL set at source node, which is calculated as:

$$\mathrm{R}(h) = \mathrm{T}(h) + \mathrm{L}(h) \times (t + p) \tag{3.3}$$

where $R(h)$ is the time deadline for expected ACK, $T(h)$ is the maximum live time for a data packet in theory (Equation 3.1) , $L(h)$ is the TTL set for an ACK (Equation 3.2), h is the data packet's TTL set by source, t and p are defined in Equation 3.1.

Substituting $T(h)$ and $L(h)$ with Equations 3.1 and 3.2, we obtain the following:

$$\mathrm{R}(h) = \left(\left\lceil \frac{h}{3} \right\rceil + 2h \right) \times (t + p) \tag{3.4}$$

in which it clearly indicates that the expected ACK return time is directly dependent to the value of packet's TTL set at the source node, with the values of t and p being constant. Thus, the packet retransmission window can be dynamically adjusted by the upper application if it is awarded the authority to change the setting of packet's TTL.

3.6 Revised PDT entry

With above retransmission process in place, the PDT design needs to be revised to correctly handle duplicate copies of packets generated not by distinct routing paths but by retransmissions. Actually the PDT design needs to differentiate two cases: the node is an intermediate node for the retransmitted packet, or the destination of the retransmitted packet. If the node is just a hop on the routing path for the retransmitted packet, we need to revise the PDT by adding a retransmission count to the table entry to distinguish the retransmitted copy from others. Without such a retransmission count, a node receiving a retransmitted copy of packet will drop the packet for the record entry associated with the packet (generated by original copy or other retransmitted copies) is already existed in the PDT, which is not correct. The revised PDT entry design is shown in Fig. 6, in which the field *Retransmission_count* is added to uniquely identify a retransmitted packet. The value range of *Retransmission_count* is 0-2, with 0 represents the original copy of data packet, and maximum value 2 means the packet can be transmitted 3 times in maximum. At the destination node, when receiving a retransmitted packet, the node can determine it reaches the destination by checking the destination setting in the packet data and drop any retransmitted copies if there is an entry associated with the packet existed in the PDT (Table 3).

Packet retransmission generates an impact on not only the uniqueness of multiple copies of a packet but also the expiration time of an entry in the PDT, i.e., $T(h)$ in Equation 3.1, as well. At the destination node, let us assume the node processes a packet, saves the record entry to the PDT, and sets the expiration time of the entry to $T(h)$. Also assuming that the ACK sent by destination somehow does not return to the source and the waiting time reaches $R(h)$ (calculated by Equation 3.4), thus, the source sends out a retransmitted copy of the packet. Since $R(h) > T(h)$ (see Equation 3.3), when the retransmitted copy launched, the respective record entry in the PDT most likely is expired and removed from the PDT according to the PDT entry elimination algorithm. So, when the retransmitted copy arrives destination, it will be incorrectly processed again as a new packet, which is a waste of system resource. This reminds us that, when data packets reach the destination, the corresponding PDT entry's $T(h)$ value shall be adjusted to match the $R(h)$ value so that the PDT entry may live long enough to be used for the possible retransmitted copy arriving at the destination. We suggest that the $T(h)$ value of a PDT entry associated with a packet reaching destination is calculated as bellow

$$\mathrm{T}(h) = R(h) + \mathrm{R}(1.2h) + T(1.44h) \tag{3.5}$$

where h is the initial value of packet's TTL (not retransmitted copy's), $R(h)$ is calculated by Equation 3.4, R (1.2 h) is counted as the extra live time for first retransmission and T (1.44 h) is the maximum live time for the second retransmitted packet. Using Equations 3.1-3.4, Equation 3.5 can be simplified as

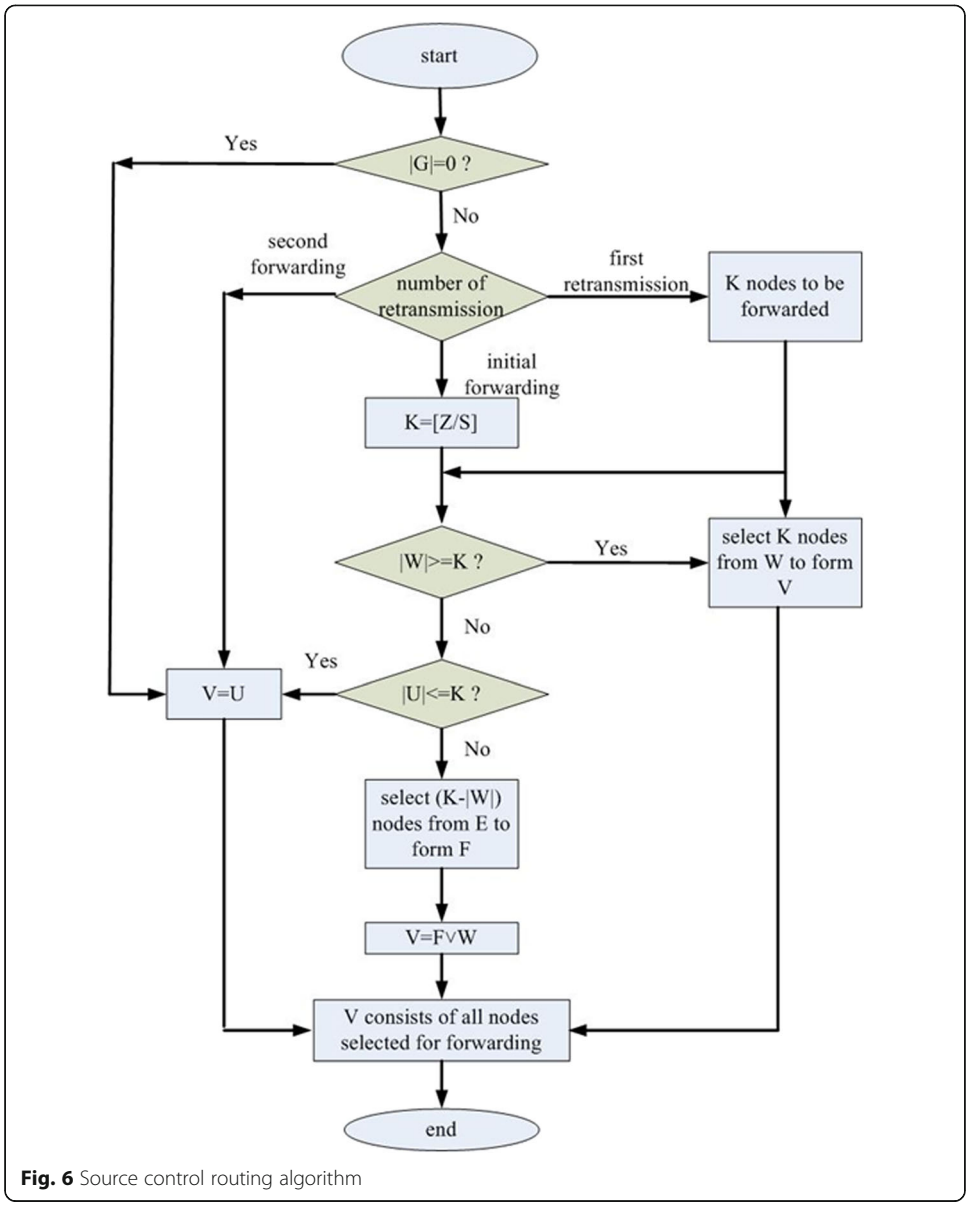


Fig. 6 Source control routing algorithm

Table 3 Revised processed data table

Field name	Type	Description
Originating_node_addr	Char string	Source node address
Destination_node_addr	Char string	Destination node address
Data_serial_number	Integer	Serial number for a packet assigned by source node
TTL_value	Float	TTL value for this PDT entry
Retransmission_count	Integer	Count number attached to different copies of a data packet in retransmission

$$\mathrm{T}(h) = \frac{91}{25} \times \left(\left\lceil \frac{h}{3} \right\rceil + 2h \right) \times (t + p) \tag{3.6}$$

If we use *caddr* to represent the address of current node's address and *daddr* to represent the destination node's address, a generalized calculation for a PDT entry's maximum live time for a packet arriving at a node, either it is its destination or not, can be presented as

$$\mathrm{T}(h) = \begin{cases} \frac{91}{25} \times \left(\left\lceil \frac{h}{3} \right\rceil + 2h \right) \times (t + p), caddr = daddr \\ h \times (t + p), caddr \neq daddr \end{cases} \tag{3.7}$$

3.6.1 Source routing record table

When an ACK is returned from the destination node A across a returning path, at a node C on the path, C can obtain such information from the ACK packet: ACK's origin node A and the last-hop node B before ACK reaches C (obviously B is also a C's adjacent node). If we save such information as origin A and last-hop B in a look-up table at each node in network, next time at node C when it needs to route a packet to node A, can use this look-up table to find node B as the next-hop to forward packet to. Such a look-up table built up by using the ACK information is called the source routing record table (SRRT). When building the SRRT, it shall be noticed that, first, there might be more than one routing path to the origin node, so multiple records pointing to the same origin node may exist in the table; second, due to dramatic topology changes caused by constantly moving mobile nodes, the record entries in SRRT require to be updated dynamically (for instance, at one moment, node B is adjacent to node C, but in other second B moves out of one-hop range of C and is not adjacent to C anymore). We design such an SRRT record updating algorithm as below:

i) when an ACK reaches a node, if no respective record existed in the SRRT, save the record entry into the table and set the TTL of record to (ta + 4) seconds, where *ta* is the ACK arriving time, and the 4 s for consideration of the moving speed of a mobile node and the bluetooth one-hop transmission time. This value can be adjusted in protocol implementation according to upper application's demand
ii) when an ACK reaches a node with a respective record already existed in the SRRT (same origin node and same last-hop node), then update record TTL by (ta′ + 4) where *ta′* is the new arriving time of ACK
iii) under high workload situations, ACK maybe wait for processing at node C for too long, during which time the last-hop node B may move out of one-hop range of node C and becomes non-adjacent, so C needs to check its neighbor table to make sure node B is still there
iv) each node periodically scans SRRT and eliminates those records that expire their TTL
v) if an adjacent node's record in the neighbor table is removed due to topology changes, the corresponding record in the SRRT shall also be eliminated

3.7 Source control routing

With the SRRT established at each node, if the destination node information has been stored in the SRRT table at each node through the first few ACKs returned from the destination, a large number of data packets with the same destination can be transferred in a much more efficient way, by means of routing information provided by SRRT. We use the following symbol set to describe the source control routing algorithm:

U: the node set containing all the candidate nodes in the source routing algorithm, from which the next-hop will be selected for packet forwarding. When the application starts, the set **U** is initialized to contain all the adjacent nodes of the hop

N: size of set **U** in number of nodes

A: node set containing all the nodes in the processed node list embedded in an arriving packet

Z: number of adjacent nodes of host node

G: node set containing all last-hop nodes of those records with the same destination in the SRRT

W: node set consists of the nodes both contained in **U** and **G**

V: node set consists of nodes selected by the source control routing algorithm for next-hop forwarding

The data flow process of source control routing algorithm is described in Fig. 6, in which the routing decision at each node is based on the following steps:

i) if **G** is empty, the process stops
ii) retrieve an arriving packet's retransmission count and calculate the number of candidate nodes (**K**) for forwarding: for Retransmission_count = 0 (initial copy of packet), **K** = $\mathbf{Z}/5$; Retransmission_count = 1 (1st time retransmitted copy), **K** = $3\mathbf{Z}/5$; Retransmission_count = 2 (2nd time retransmitted copy), **K** = **Z**
iii) selection policy: if Retransmission_count = 2 (second retransmission), let **V**=**U**, i.e., forward the packet to all nodes in **U**; else if $\mathbf{K} \le |\mathbf{W}|$, randomly select **K** nodes from **W**, or if $\mathbf{K} > |\mathbf{W}|$ and $\mathbf{K} \ge |\mathbf{U}|$, let **V**=**U**; else $\mathbf{V}=\mathbf{W} \cap \mathbf{X}$, $|\mathbf{X}| = \mathbf{K} - |\mathbf{W}|$ and the nodes in **X** are randomly selected from **U** and different from those in **W**
iv) **V** is formed and forwarding can be started

The above source control routing approach will route the initial copy (not transmitted) of most data packets to the destination, and the retransmission procedure (retransmit two times in most) will most likely make the packets that do not reach their destination eventually reach. We have to mention that the source control routing can only be applied to data packets but not ACK packets since the latter are transmitted without retransmission mechanism.

In this chapter, we present a cross-layer design of the bluetooth device-to-device routing protocol (BDRP) in which the proactive routing paradigm is improved to provide a low delay and high arriving rate performance in a dynamically changed topology. The packet's time to live (TTL) and node's processed data table (PDT) are implemented to filter out undesirable duplicate packet circulation in the network, and the node exclusion by packet (NEP) mechanism is used to effectively prevent duplicate packets bouncing between nodes and wasting network resources. By the cross-layer

design, a tailored retransmission process is added to the routing protocol layer to ensure most of packets can reach their destination even under a constantly changing topology and a high workload. The source routing record table (SRRT) is used to support the Source Control Routing (SRC) algorithm. When a data packet needs to be transferred to the destination, each node on the path can use this information to quickly find an efficient route to forward the packet.

4 Results and discussion

In order to evaluate the effectiveness and performance of the proposed designs in the BDRP approach, we develop a simulation program to emulate a mobile phone mesh network with varied configuration parameters such as node dense, moving speed, and topology size and with a graphic view to show an animation of a packet going through a routing path in a mesh network (Table 4). The simulation program is coded with the following key features:

- A mobile node in the mesh network is emulated as a thread in the simulation, with its own message queues and routing functions
- A delay time queue is used to emulate the transmission times between various nodes, table entry expiration times, and packet retransmission control
- A timeout setting on the delay time queue to emulate the movement of nodes
- A global thread if responsible for maintaining and updating the neighbor table of each node
- The emulated node moving speed is randomly distributed between 0 to 5 m/s with an orientation randomly selected between 0 to 360°.

The system specification of the simulation platform is shown in Table 1, the test parameters are set as below:

Topology: the mobile nodes are located and moved within a square area with fixed side length

Bluetooth transmission range: 25 m

Device-to-device on-hop transmission time: 50 ms

Node processing time for one packet: 10 ms

Test case: two nodes are randomly selected, with one as the source and the other destination, and in one transmission 5000 packets will be transferred from source to destination with a transmission speed at 10 ~ 100 packets/s; in each test case, this transmission action will be repeated 3000 times and measured in its average values

The experiments are designed in two types: BDRP verification experiment and BDRP vs. OLSR performance comparison experiment, with the former examining the effectiveness of designed routing functions of BDRRP and the latter comparing the performance of BDRP and OLSR.

Table 4 Simulation system configuration

O/S	OS X El Captain
CPU	2.6GHz Intel Core i5
Memory	8GB 1600 MHz DDR3
SDK	JDK 1.8.0_121-b13, Java HotSpot (TM) 64-bit Server VM

4.1 BDRP verification experiment

4.1.1 Node exclusion by packet test

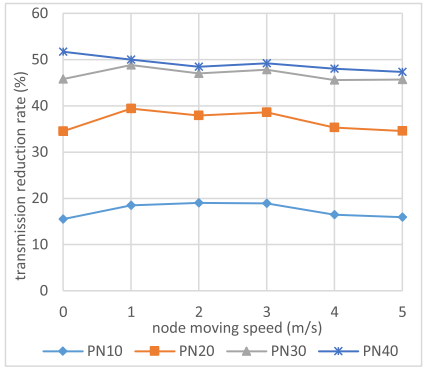
The text is designed to examine the node exclusion by packet (NEP) mechanism's effectiveness in eliminating duplicate packets circulating among shared adjacent nodes, which can be measured by the packet transmission reduction rate. In the NEP design, we put a bound PN_Max_Num on the number of nodes that can be contained in the processed node list of a data packet to prevent the packet become too large to process. Theoretically, the larger value of PN_Max_Num would help exclude undesirable nodes in packet forwarding, yet the smaller value would make packets smaller and easier for processing. In this experiment, we intend to inspect the PN_Max_Num's impact on packet transmission reduction rate under varied conditions such as node moving speed, node density, and size of covered area, and to find an appropriate value of PN_Max_Num that leverages transmission reduction efficiency and packet data size.

The test includes 6 test cases, with each case's PN_Max_Num set to 10, 20, 30, 40, respectively. In each test, 5000 packets are transferred within an 80 m × 80 m emulated area containing 100 mobile nodes with each node may have up to 15 bluetooth connections to other nodes simultaneously. In each test, the node moving speed is randomly distributed into 6 ranges between 0 to 5 m/s with an orientation randomly selected between 0 to 360°. The simulation experiment results for NEP is presented in Fig. 7, in which PN 10 is for test case PN_Max_Num = 10, PN20 for PN_Max_Num = 20, …, etc..

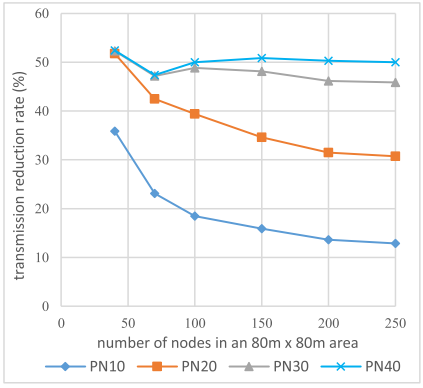
Figure 7a shows the test results under varied node moving speed, which indicate that the reduction rate of packet transmission is not sensitive to node moving speed, but significantly impacted by PN_Max_Num, with a larger value yields a better reduction result. In Fig. 7a, we also observe that, though the highest PN_Max_Num value (PN50) gives the largest reduction rate across various node moving speed spans, yet PN30 yields a reduction rate of 35 ~ 40%, which reaches around 80% of the PN40's number. Above PN30, the larger PN_Max_Num value still gives better reduction number, but in a very narrow space. So, we consider there is an optimized value for PN_Max_Num which, in this test, falls between 30 to 35.

Figure 7b displays test results under varied node density in a covered area 80 m × 80 m, with a node number of 40, 70, 100, 150, 200, 250, respectively. The node moving speed is randomly between 0 ~ 1 m/s and each node is allowed to connect to up to 15 nodes simultaneously. From the results, we find the packet transmission reduction rate decreases along with the increase of node density for all PNs. Yet at a high node density (node number = 150 or higher), PN30 maintains a considerably higher reduction rate of 45-48%, compared to PN10's and PN20's 13-34% on the same span, which demonstrates that PN_Max_Num = 30 yields a stable performance even under high node density situations.

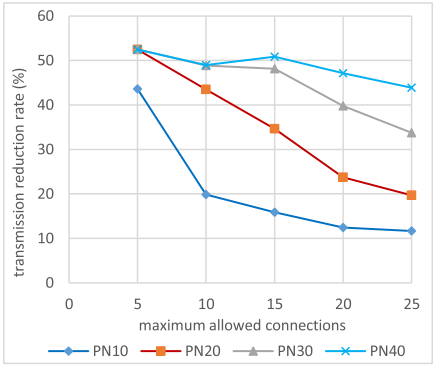
Figure 7c is the measurement on NEP's performance under varied maximum number of adjacent nodes. In this test, 150 nodes are distributed in the same 80 m × 80 m area with a moving speed between 0-1 m/s, and the maximum number of adjacent nodes are set to 5, 10, 15, 20, and 25, respectively. The results in Fig. 7c show that, under all PNs, the reduction rate dramatically drops along with the increasing maximum number



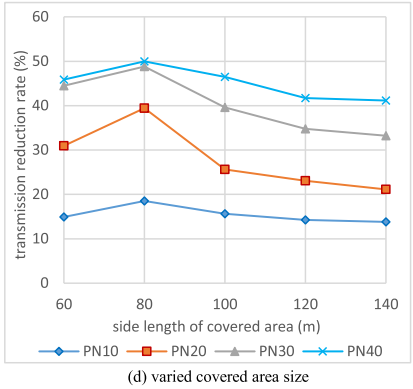
(a) varied moving speed



(b) varied node density



(c) varied maximum allowed connections



(d) varied covered area size

Fig. 7 PN_Max_Num's impact under various conditions. **a** Varied moving speed. **b** Varied node density. **c** Varied maximum allowed connections. **d** Varied covered area size

of adjacent nodes allowed, particularly when the number exceeds 15. One thing we have noticed is that, prior to maximum number of adjacent nodes = 15, PN30 manages to maintain a high reduction rate close to 50%. We need to clarify that the maximum number of adjacent nodes is a design number we assign to the node neighbor table, which is normally larger than the number of physical connections a bluetooth phone allows (currently is 15). The test indicates that PN_Max_Num = 30 may be the better choice in the NEP design in terms of reducing duplicate transmission in network.

In Fig. 7d, we also test the performance of NEP against increasing area sizes from 80 m × 80 m, 100 m × 100 m, 120 m × 120 m, to 140 m × 140 m, with each size keeps the same node density and the nodes move in a speed between 0-1 m/s in random directions. Not surprisingly, the reduction rate decreases slowly along with the increasing size of covered area, in which the largest PN_Max_Num (PN40) gives the best performance under current settings.

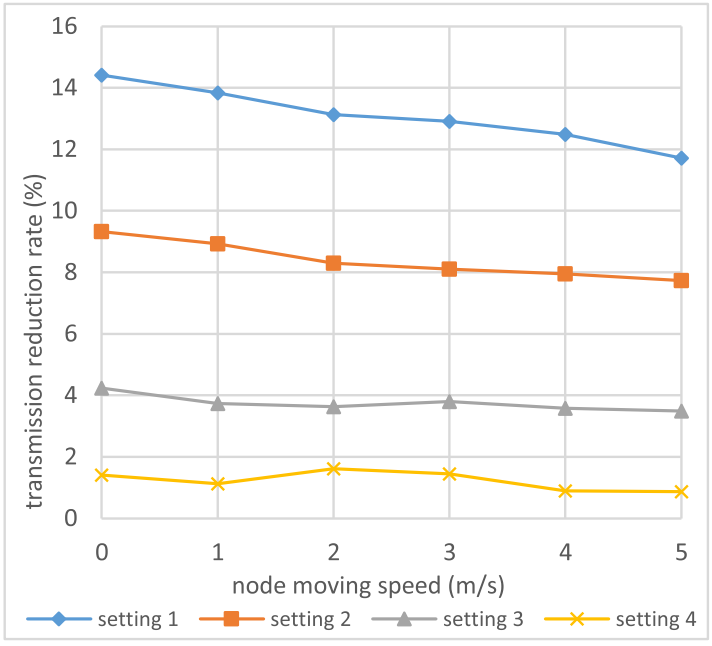
4.1.2 Source control routing test

Packet retransmission process is designed in the source control routing (SCR) to ensure most of packets will eventually arrive their destinations, even retransmissions may add to the total of packet transmissions, which is what we are trying to trim down. Simulation tests on SCR verify that, with the retransmission process, in most cases packet retransmission rate appears under 3.2% while packet non-arriving rate is controlled under 1.0%. In the SCR algorithm, let X represent the packet retransmission rate for 1st round retransmission and Y for 2nd round retransmission, which are two key factors in determining the algorithm's outcome. We conduct an SCR transmission reduction rate test in an 80 m × 80 m square area with each node can have 15 connections in maximum at one time. In the varied node moving speed test, we calculate packet transmission reduction numbers against various speed spans, while in the varied node density test, the number of nodes in the covered area is increased from 40, 70, 100, 150, to 180, with a moving speed span of 0-1 m/s. We use the following settings of combined X and Y values in the test:

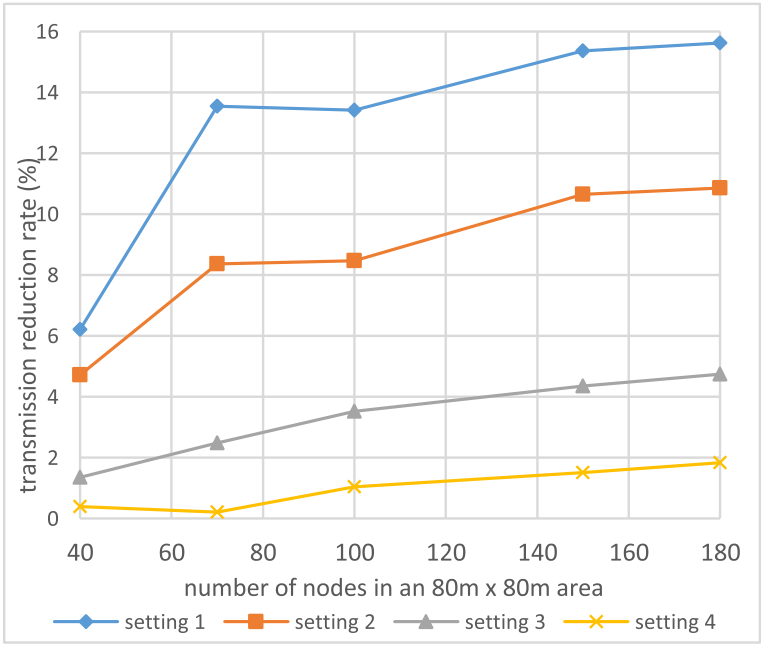
Setting 1: X = 1/5, Y = 3/5
Setting 2: X = 1/3, Y = 2/3
Setting 3: X = 1/2, Y = 3/4
Setting 4: X = 2/3, Y = 5/6

Test results displayed in Fig. 8 under varied node moving speed or varied node density clearly indicate that SCR's performance in packet filtering: (a) slowly decreases along with increasing node moving speed; (b) slowly increases along with increasing number of nodes in the area. In both cases, the setting 1 (X = 1/5, Y = 3/5) yields a better outcome with a reduction rate between 12 and 16%.

The simulation program opens a graphical window to display an overview of tested area, with a group of moving nodes among which black dots represent mobile nodes, red dot the source node, blue dot the destination node, and the black circle shows the one-hop transmission range for a node. Figure 9a displays a full routing path from the source (red dot) to the destination (blue dot) under the BDRP paradigm in a static topology (all node are drawn at their initial positions), while Fig. 9b draws the same path in a dynamic topology where each node's movement (distance and orientation) is displayed by a red colored bar started from its initial position.



(a) varied moving speed

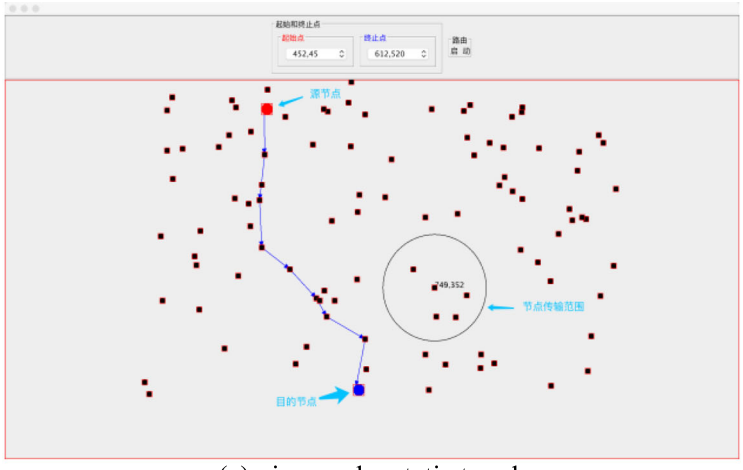


(b) varied node density

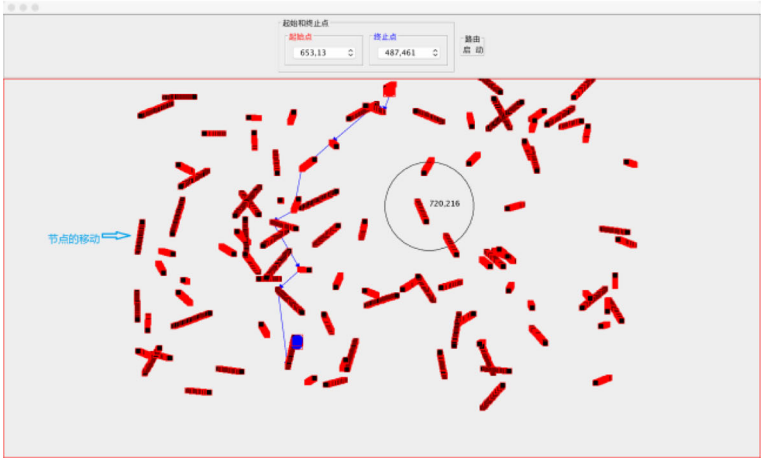
Fig. 8 SCR Performance under various conditions. **a** Varied moving speed. **b** Varied node density

4.2 Performance comparison experiment

We implement the OLSR algorithm in the simulation to use it as a reference in the performance comparison experiment between BDRP and OLSR. The test is conducted in an 80 m × 80 m area with 100 mobile nodes moving at a speed between 0-5 m/s. The maximum number of simultaneous connections, a node is allowed by the bluetooth device is 15, and the time interval for the transmission control (TC) message of OLSR is set to 2 s. Still in each test case, 5000 packets are transferred and each test is



(a) view under static topology



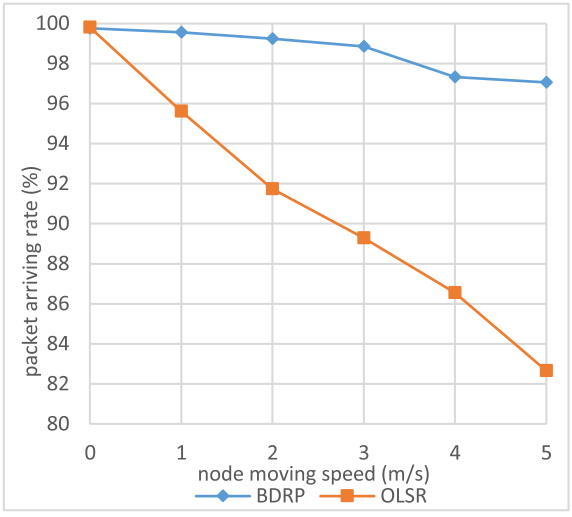
(b) view under dynamic topology

Fig. 9 A graphic view of simulation program. **a** View under static topology. **b** View under dynamic topology

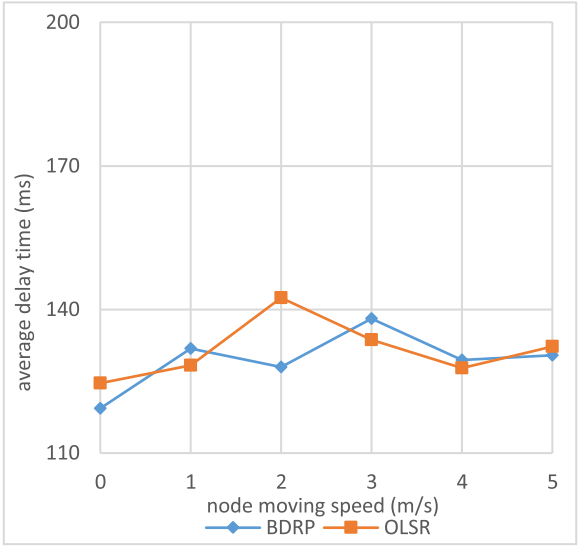
repeated a few thousand times to obtain statistical results. In the comparison experiment, we use three performance metrics as packet arriving rate, average delay time, and network workload to compare the two routing algorithms' outcome under same test configurations.

The two approaches' packet arriving rates under varied node moving speed are shown in Fig. 10a, in which OLSR's arriving rate drops quickly (> 15%) along with the increase of node moving speed while BDRP manages to maintain the decrease to a small margin (< 3%). This clearly indicates that BDRP has a better ability to dynamically adjust to the topological changes to ensure most packets reach their destinations, most likely contributing to BDRP's packet retransmission process implemented in the source control routing algorithm. As for the average delay time, Fig. 10b indicates that both BDRP and OLSR are not sensitive to varied node moving speeds, which diminishes our concerns that the packet retransmission mechanism BDRP deploys may increase transmission delay as the cost of lifting packet arriving rate.

Another comparison experiment we conduct is the workload comparison test between BDRP and OLSR, in which the workload at a node is measured by the number



(a) packet arriving rate vs. node moving speed



(b) average delay time vs. node moving speed

Fig. 10 BDRP vs. OLSR under varied node moving speed. **a** Packet arriving rate vs. node moving speed. **b** Average delay time vs. node moving speed

of unprocessed packet per unit time and the network workload at a moment is the average workload at all nodes in the network. The test is conducted in a configuration as same as previous comparison test except the node moving speed range is changed to 0-2 m/s and OLSR's TC time interval is reset to 0.5 s. In each test case, 5000 packets are launched at the source node under a varied launching speed, and we set a counter at each node in the network to record the number of unprocessed packets at one moment. Test results are drawn in Fig. 11, which shows that, at a low packet launch speed (< 60 packets/s), BDRP yields a better number in network workload than OLSR does, yet when packet launch speed is lifted to a high level (> 60 packets/s), BDRP's workload number is higher than that of OLSR. Higher workload number means more packets waiting in the line for processing at nodes, which in term means longer delay time for packets in transmission process. This may become a disadvantage for BDRP when it is

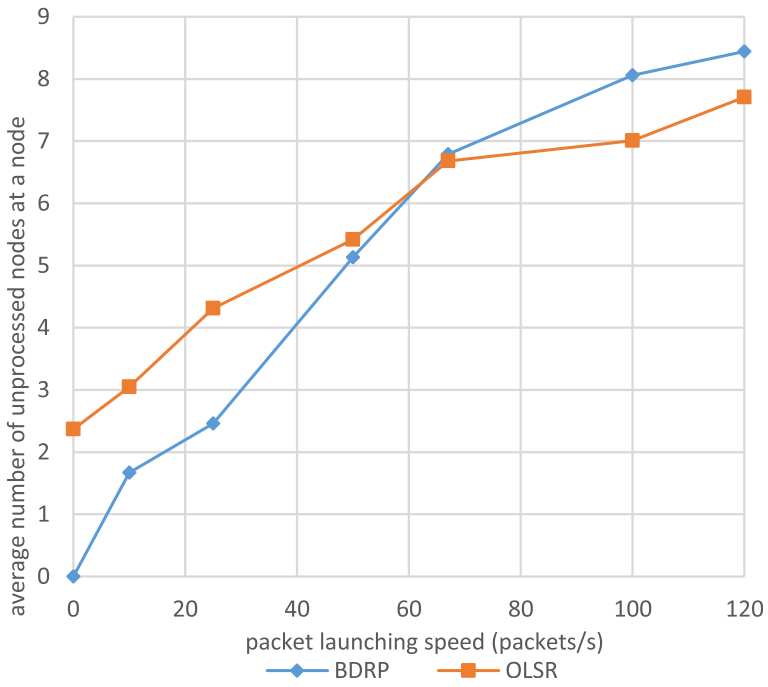


Fig. 11 BDRP vs. OLSR under varied packet launch speed

needed to support the applications that require to send an extra-large file such as video to the destination in a short time. Yet for the text and image files currently supported in the bluetooth-based ad hoc wireless applications, the BDRP approach yields a satisfied performance, particularly for being suitable to a dramatically changed network topology caused by constantly moving nodes.

5 Conclusions

In this paper, a cross-layer proactive type approach for bluetooth mobile-phone mesh network, namely, the bluetooth-based device-to-device routing protocol (BDRP), is presented and simulation experiments are conducted to evaluate the performance of BDRP under varied work conditions. Packet arriving rate, transmission delay and protocol's suitability to dynamically changing topology are major performance requirements on the routing paradigm of mobile phone-based ad hoc mesh network.

Corresponding to above requirements, we propose a node-based processed data table (PDT) design to eliminate duplicate packet forwarding among the intermediate nodes on a routing path, and a packet-based node exclusion by packet (NEP) algorithm to prevent duplicate packet circulation through the shared adjacent node set between two nodes. The PDT and NEP together provide an effective and efficient means in routing layer to filter out a great number of undesirable packet transmissions in ad hoc mesh network, which in turn contributes to a significant reduction in packet transmission time. To ensure most packets arrive their destinations under a dramatically changing topology due to mobile node's constant moving, we implement a tailored packet retransmission mechanism in the routing protocol layer to retransmit the packets that lost in previous transit up to three times. This cross-layer approach provides an assurance to the packet arriving rate that is essential to upper applications. One major

drawback of proactive type routing approach is that, to maintain routing path information up to date, the protocol needs to broadcast an extremely large number of transmission control messages to network to accommodate to a dynamically changed topology, which causes an undesirable high workload to network that consumes scarce network bandwidth and computing power. In our approach, the source record routing table (SRRT) based source control routing (SCR) algorithm implements a high efficiency source backtrack routing mechanism, in which the routing information of an intermediate node to a destination node is recorded and maintained at the intermediate node, through the data packets or ACKs previously went through the node. When a new node arrives at the node for forwarding, the algorithm can quickly identify the next-hop node for transmission. The SRRT/SCR approach provides a highly effective and efficient routing design fully suitable to a dramatically changing topology, yet without multicasting an overwhelming amount of routing control messages in the network. In the simulation experiment conducted, it demonstrates that, even with the retransmission process, BDRP does not generate a high workload in network.

A simple and naive routing solution to all ad hoc wireless mesh network is the flood multicasting for packet transmission at each node, which causes an overwhelming volume of packets, including a great number of duplicate packets, are circulating in the network and consuming scarce network resources. Reducing packet transmission in network by eliminating those unbeneficial duplicate packet circulation is one of the major design goals for any optimized routing protocols. The effectiveness and efficiency for filtering out undesirable packet transmissions can be measured the transmission reduction rate, which can be obtained by comparing the proposed routing approach to the flood multicasting solution. The verification experiment we conduct in this work indicates that, under varied test conditions such as node moving speed, node density, topology size, and maximum number of adjacent nodes, the BDRP's processed data table (PDT) and node exclusion by packet (NEP) designs yield a satisfied transmission reduction rate of 33 to 50% (with PN_Max_Num = 30). A higher PN_Max_Num value may allow the protocol to handle a larger area with more mobile nodes, yet in the cost of a longer packet data segment and longer processing time. Simulation results for the source control routing (SCR) algorithm also demonstrate its effectiveness in reducing packet transmissions even with the packet retransmission process.

The comparison between BDRP and OLSR through simulation experiments shows that BDRP outperforms OLSP in packet arriving rate, particularly, when node moving speed goes high from 0 to 5 m/s, OLSP's packet arriving rate drops sharply from 100 to 83% while BDRP maintains at 97%. Higher node moving speed will cause more dramatic changes in network topology and BDRP demonstrates its better suitability to this character. The comparison test between BDRP and OLSR on transmission delay tells us that, with a clearly higher packet arriving rate, BDRP does not sacrifice its timely responsiveness in packet transmission by keeping its packet delay time at the same level as OLSR's. Our explanation on this is that, even retransmission of a fraction of packets (in our test < 3.2%) adds to the transmission time, the effectiveness of SRC in reducing network workload and the quickness in finding a forwarding path for packet compensate to BDRP's overall performance.

Based on what we have found in the work, we come up with a preliminary conclusion that the combination of node-based PDT and packet-based NEP is an efficient way to

filter out duplicate packet circulation in network, the cross-layer retransmission design provides an effective mechanism to ensure most packets arrive destinations, and the SRC algorithm's source backtracking approach demonstrates to be a quick and precise routing solution, particularly for the scenarios when a quite large number of packets targeted on the same destination. We also observed that there are still a few issues opened for further investigation. The parameter PN_Max_Num plays a key role in the BDRP's NEP algorithm that filters out no-use duplicate packet transmission. The appropriate value of PN_Max_Num seems related to the maximum number of physical connections a bluetooth phone can support (right now is 15), and the setting PN_Max_Num = 30 in the tests seems yield a satisfied outcome. But we are wondering if the value of PN_Max_Num shall be linearly correlated to the number of physical connections the Bluetooth device allows. If not, how shall we select value for PN_Max_Num? Obviously, we need more analytical research and experiment work on this topic. The combination of X and Y is selected to be the minimum values in the BDRP's retransmission process, and yields a satisfied packet arriving rate 97% even at a high node moving speed 5 m/s. If the covered area size is enlarged with a higher node density, it can be foreseen that more packets will be lost in transmit due to dramatic topology changes. Then what values shall we set X and Y in order to maintain the packet arriving rate to an accepted level, yet without a large volume of retransmitted packets to overload the network? In Fig. 11, we noticed that, when packet launching speed at the source reaches a certain level (> 60 packets/s), BDRP's workload becomes higher than OLSR's, which means a possible jitter may happen at the intermediate nodes in routing path. This also needs to be examined in future researches.

Abbreviations

AODV: Ad hoc On-demand Distance Vector Routing; BDRP: Bluetooth-based device-to-device routing protocol; DSR: Dynamic source routing; MPR: Multipoint relay; NEP: Node exclusion by packet; OLSR: Optimized link state routing; PDT: Processed data table; SCR: Source control routing; SRRT: Source routing record table; TC: Transmission control; TTL: Time to live

Authors' contributions

AF carried out the Device-to-device Routing Protocol studies and drafted the manuscript. ZT carried out the simulation experiment, and participated in the Device-to-device Routing Protocol studies. WW participated in the design of the study and performed the statistical analysis. YT participated in the design of simulation and helped to draft the manuscript. DL participated in the statistical analysis. All authors read and approved the final manuscript.

Funding

This work is supported by the Fundamental Research Funds for the Central Universities (ZYGX2019J076).

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

There are no financial and non-financial competing interests.

Author details

[1]Xi'an Polytechnic University, Xi'an 710048, China. [2]University of Electronic Science and Technology of China, Chengdu 610054, China.

Received: 29 December 2019 Accepted: 14 July 2020
Published online: 13 August 2020

References

1. Z. Na, Y. Wang, X. Li, J. Xia, X. Liu, M. Xiong, W. Lu, Subcarrier allocation based simultaneous wireless information and power transfer algorithm in 5G cooperative OFDM communication systems. Physical Communication **29**, 164–170 (2018)
2. Z. Na, J. Wang, C. Liu, M. Guan, Z. Gao, Join trajectory optimization and communication design for UAV-enabled OFDM networks. Ad Hoc Netw. **98**, 1–10 (2020)

3. S. Guo, O. Yang, Energy-aware multicasting in wireless ad-hoc networks: a survey and discussion. Comput. Commun. **30**(9), 2129–2148 (2007)
4. X. Liu, M. Jia, X. Zhang, W. Lu, A novel multichannel internet of things based on dynamic spectrum sharing in 5G communication. IEEE Internet Things J. **6**(4), 5962–5970 (2019)
5. D. Silva, J. Henao, C. Pedraza and F. Vega, Use of bluetooth technology for applications of intelligent transportation system. 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, 28-29 April (2016).
6. A.K. Arumugam, S.M.D. Armour, B.S. Lee, et al., Consumer electronics application and coverage constraints using bluetooth and proposed bluetooth evolution technologies. IEEE Trans. Consum. Electron. **47**(3), 283–293 (2002)
7. X. Liu, X. Zhang, NOMA-based resource allocation for cluster-based cognitive industrial internet of things. IEEE transactions on industrial informatics **16**(8), 5379–5388 (2020)
8. X. Liu, X. Zhai, W. Lu, W. Celimuge, QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA. IEEE Transactions on Industrial Informatics **99**, 1–10 (2019)
9. Di Lin, Su Hu, Yuan Gao, Wanbin Tang. Heuristic-learning-based network architecture for device-to-device user access control. IEEE Commun. Mag.. 57. 96-101(2019).
10. S. Mohseni, R. Hassan, A. Patel and R. Razali. Comparative review study of reactive and proactive routing protocols in MANETs. 4th IEEE International Conference on Digital Ecosystems and Technologies, Dubai, 13-16 April (2010).
11. T. Kunz, R. Alhalimi, Energy-efficient proactive routing in MANET: energy metrics accuracy. Ad Hoc Netw. **8**(7), 755–766 (2010)
12. Mbarushimana C, Shahrabi A. Comparative study of reactive and proactive routing protocols performance in mobile ad hoc networks. International Conference on Advanced Information Networking and Applications Workshops. Niagara Falls, 21-23 May (2007).
13. K. Pallavi, R. Monika, S. Alankar, et al., Performance study of ad-hoc reactive routing protocols. J. Comput. Sci. **6**(10), 1159–1163 (2010)
14. S. Azadm, A. Rahman, F. Anwar, A performance comparison of proactive and reactive routing protocols of mobile ad-hoc network (MANET). J. Eng. Appl. Sci. **5**, 891–896 (2007)
15. B. Karthikeyan, N. Kanimozhi and S. H. Ganesh. Analysis of Reactive AODV Routing Protocol for MANET. 2014 World Congress on Computing and Communication Technologies. Trichirappalli, 27 February-1 March (2014).
16. H. Cheng, J. Cao, A design framework and taxonomy for hybrid routing protocols in mobile ad hoc networks. IEEE Communications Surveys & Tutorials **10**(3), 62–73 (2008)
17. S. Bhat, A performance study of proactive, reactive and hybrid routing protocols using Qualnet simulator. Int. J. Comput. Appl. **28**(5), 10–17 (2011)
18. Sahu S K, Prusty D, Jena S K. Comparative performance analysis of table-driven and on-demand routing protocols in mobile ad-hoc network. International Conference on Futuristic Trends in Computational Analysis and Knowledge Management. Greater Noida, 25-27 February (2015).
19. O. Ogundile, M.O. Oloyede, O.A. Osanaiye, Selective-path clustered based routing protocol for large scale wireless sensor networks. International Journal of Communication Networks and Distributed Systems **25**(3), 284–306 (2020)
20. C. Tausch, P. Konstantiniuk, A. Haid, et al., RFC7187: Routing multipoint relay optimization for the optimized link state routing protocol version 2 (OLSRv2). Acta Chirurgica Austriaca **32**(3), 99–104 (2014)
21. M.S. Haghighi, Z. Aziminejad, Highly anonymous mobility-tolerant location-based onion routing for VANETs. IEEE Internet Things J. **7**(4), 2582–2590 (2020)
22. B. Master, P. Shete, Adaptive energy-efficient on-demand distance vector routing protocol for MANET. Int. J. Comput. Appl. **100**(7), 39–43 (2014)
23. Martin Murhammer and Eamon Murphy. TCP/IP Tutorial & Technical Overview, 6th. edition, Simon & Schuster, Inc., USA, (1998).
24. Comer D E. Internetworking with TCP/IP, volume 1: principles, protocols, and architectures, 4th edition, Prentice Hall PTR, USA, (2000).

Publisher's Note