Data-Collecting Protocol Based on a Rooted Tree for Wireless Ad Hoc Networks Constructed Using Data Transmission Modems

SATORU OHTA
Department of Information Systems Engineering, Faculty of Engineering
Toyama Prefectural University
5180 Kurokawa, Imizu-shi, Toyama 939-0398
JAPAN
ohta@pu-toyama.ac.jp https://www.researchgate.net/profile/Satoru_Ohta

Abstract: - This paper presents a simple communication protocol that effectively enables a base station to collect data from nodes in wireless ad hoc networks. The proposed protocol assumes that the network is constructed using a data transmission modem that uses a lower frequency radio wave than that employed by the frequently used systems such as wireless LANs or ZigBee. Since a lower frequency radio wave is advantageous because of less signal attenuation, the modem can provide a more reliable wireless link. Unfortunately, the modem does not support network functions. To construct a network with the modem and collect data from nodes, an appropriate communication protocol is required. The protocol must avoid packet collision because radio waves transmitted from nodes may generate interference. In addition, since the network is multihop and has mesh topology, a routing mechanism is necessary. To satisfy these requirements, the proposed method employs a token-passing approach that utilizes a rooted tree structure, which is extracted from a given network. By utilizing the tree structure, the routing mechanism is significantly simplified. Furthermore, packet collisions are eliminated by the token circulated in a tree. The algorithm that extracts a rooted tree from a given network is also shown. The feasibility and performance of the proposed protocol was examined on a prototype network.

Key Words: - ad hoc network, tree, routing, token passing, communication protocol, wireless network

1. Introduction

Wireless ad hoc networks are extensively used for various applications [1–3]. The technical requirements for ad hoc network implementation can differ greatly depending on the application. Thus, the wireless transmission system and communication protocol must be selected optimally according to the target application requirements.

Some wireless systems such as the IEEE 802.11 series (wireless LANs) and IEEE 802.15 (ZigBee) are standardized and can be used to build ad hoc networks without modification. However, these methods are not always optimal for a particular application. As an alternative system, wireless data transmission modems, which are attractive as data links for some applications because they provide robust transmission over long distances, are commercially available.

Here, we consider an ad hoc network for a climber surveillance system that is being developed to ensure safety of mountain climbers. The network connects mountain huts and enables base stations to collect climber position data. For this application, a network protocol is required to send data stored at the huts to the base station. We plan to use a 429-MHz wireless data transmission modem [4] for data links. The theory of radio wave propagation indicates [5] that signal attenuation is smaller for longer wavelengths. The loss due to rainfall is also less for low frequencies [6]. Moreover, signal attenuation due to obstacles is less at low frequencies because the diffraction of radio waves is high [5]. Thus, the 429-MHz band provides a more reliable wireless link than the 2.4 or 5 GHz band.

Unfortunately, the 429 MHz wireless modem does not provide network functions. Thus, to use this modem in the mountain hut network, we must develop a network protocol that avoids packet collision, provides routing, and efficiently collects data distributed among mountain huts.

A data-collecting protocol featuring data flows from different source nodes to a sink node is often referred to as convergecast [7]. In a previously proposed method [7], Time Division Multiple Access (TDMA) was used to avoid packet collision. However, the TDMA method cannot efficiently handle bursty or multirate traffic because it relies on fixed time slots within a periodic frame. This is inadequate for our application.

This paper proposes a simple data-collecting protocol. The proposed protocol enables a base station to collect data from every node in a mesh multihop network by executing a simple algorithm. The algorithm does not require each node to manage the global network information; moreover, a node only has to know very partial neighbour node information.

To avoid packet collision, the proposed method employs a token-passing approach [8–10] rather than TDMA or contention methods. Another feature of the proposed method is that all packets are transmitted over a rooted tree embedded in the mesh topology network. With this approach, packet collision is eliminated and the routing mechanism is simplified significantly. The feasibility of the protocol was confirmed using a prototype network with nodes implemented using a small computer board and a wireless data transmission modem. The performance of the prototype system was evaluated.

Although this paper assumes the employment of a 429-MHz wireless modem, the proposed protocol works for different types of networks as well. In the market, various data transmission modems are commercially available. These modems use frequencies such as 150 MHz, 900 MHz, 1.2 GHz or 2.4 GHz. Obviously, the protocol is applicable to networks built with any of these modems. Independently of the employed wireless system, the proposed method is advantageous in the easiness of implementation and node setting.

The remainder of this paper is organized as follows. In Section 2, the paper explores the target application and describe technical problems. In Section 3, the proposed protocol is explained. In Section 4, the prototype implementation is presented. Experimental results are reported in Section 5, and related work is reviewed in Section 6. Finally, conclusions are presented in Section 7.

2. Problem Description

The target application is the ad hoc network for the climber surveillance system, which is shown in Fig. 1. The system collects and manages position information of climbers in mountainous terrains. The climber position is identified using a Global Positioning System (GPS) terminal carried by each climber. The terminal has a wireless data transmitter that sends position information to neighbouring mountain huts. A communication node is placed at each hut to form a network and position data collected at each node are transmitted to a base station via this network. Consolidated position data facilitates the discovery and rescue of missing or injured climbers. The scope of this study is to develop a network that connects mountain huts.

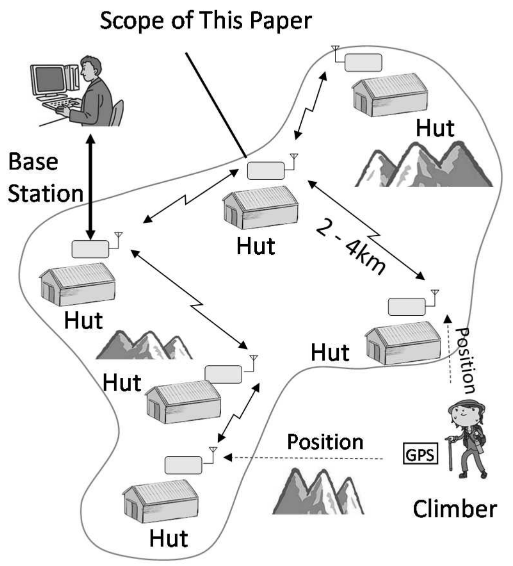

Fig. 1. Target climber surveillance system.

For the mountain hut network, building telecommunication infrastructure, such as cables, is not feasible because of construction difficulties, economic constraints, and environmental considerations. Thus, communication should be provided by a wireless ad hoc network.

To design the ad hoc network, an appropriate wireless transmission system must be selected. The system must provide reliable data transmission for the distance and environment between two adjacent mountain huts. The system that satisfies this requirement is found by assessing the characteristics of wave propagation loss.

Let P_r denote the received power of a radio signal. According to the free space model, P_r is expressed through transmitted power P_t, distance d, wavelength λ, and antenna gains G_r and G_t as follows [5]:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2} \tag{1}$$

Reference [5] derives the free space loss, *FSL* (dB), using (1) and assuming $G_t = G_r = 1$ as follows.

$$FSL(\text{dB}) = 92.4 + 20\log_{10} f(\text{GHz}) + 20\log_{10} d(\text{km}) \tag{2}$$

In (2), f is frequency in GHz. Clearly from (2), the propagation loss is greater for radio signals with higher frequencies. In addition, the diffraction of the radio signal is higher for longer wavelengths. Thus, signal attenuation caused by obstacles is also less for low frequencies. A model for signal attenuation caused by rain has been reported [6]. This model suggests that using a low frequency is also advantageous from the viewpoint of attenuation by rain.

Because of the above propagation characteristics, the 2.4 or 5 GHz radio waves used in IEEE 802.11 or 802.15 wireless systems demonstrate greater signal loss than radio waves of frequency less than 1 GHz. Considering the expected radio wave attenuation for the distance between the huts, the 2.4 or 5 GHz band is inadequate to build a reliable link. Instead, this study examines a data transmission modem with a low frequency, i.e., 429 MHz [4], for a transmission link.

A disadvantage of employing a lower frequency is the difficulty of making small sized antennas. Fortunately, a considerably large antenna is allowed for the considered application, where a node is fixed at a hut. Thus, the disadvantage in the antenna size is not significant.

Although the data transmission modem is advantageous from the viewpoint of signal attenuation, the modem does not provide network functions. Thus, it is necessary to develop a protocol that can transmit data from the huts to the base station. The data-collecting protocol must satisfy the following technical requirements:

- Multihop packet transmission is mandatory because the area is too large to be covered by the wireless transmission range of the modem.
- A link can be established between two nodes if the distance is not greater than the wireless transmission range. Thus, the network has a general mesh topology.
- Radio waves generated from different sender nodes may interfere with each other if nodes are within the wireless interference range. Thus, simultaneously transmitted packets may collide.
- Although the modem affords multiple transmission channels [4], the same fixed channel is used for all nodes in the network for simple implementation and operation.

For the first and second requirements, the protocol must provide a routing mechanism. Packet collision must also be avoided for the third and fourth requirements. The aim is to establish a network protocol that provides these mechanisms with point-to-point, half-duplex data transmission using a modem.

3. Proposed Protocol

Routing over a network becomes simple when the network topology is a tree. If a base station is placed in the tree network, the rooted tree topology (Fig. 2) can be constructed by assigning the root to the base station.

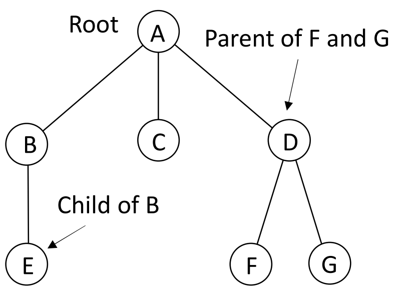

Fig. 2. Rooted tree.

In the rooted tree topology, it is very easy for the root to collect data from other nodes. By sending data from each node to its parent repeatedly, data will arrive at the root node. To request data collection from the root to other nodes, each node only needs to send request packets to its children. Thus, to collect data, it is sufficient for a node to know its parent and child nodes and global network information is not required. Because of this simplicity, the proposed protocol utilizes a rooted tree topology.

To avoid packet collision, some media access control is necessary. For media access control, previous studies on convergecast [7, 11] have employed TDMA. However, TDMA is inefficient against bursty or multirate traffic. In addition, optimal scheduling of time slots for TDMA may unacceptably complicate the operation.

Another approach is the contention method, such as CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). However, this method is not efficient because retransmission of collided packets is required. In addition, implementation is difficult to handle the hidden terminal problem [1, 12].

To avoid packet collision, this study investigates a protocol based on the token-passing concept [8–10]. In this method, only a node with a token can transmit packets; therefore, packet collision never occurs. Since this method does not rely on fixed time slots or frames, it can efficiently handle bursty or multirate data transmission.

The token-passing method has been applied to bus or ring network topologies. To implement this mechanism in other network topologies, the order in which the token is passed must be determined. If the network topology is a rooted tree, the token-passing order is determined easily using a tree search algorithm, such as a depth-first search [13]. Figure 3 shows how the token is circulated in depth-first search order.

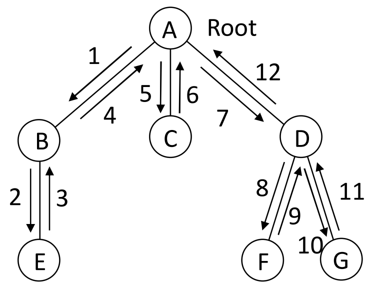

Fig. 3. Token passing in depth-first search order.

As described above, the rooted tree topology simplifies routing for data collection and token passing; however, the network topology in the target application is a mesh. Fortunately, constructing a rooted tree for a mesh network is straightforward. As described later, this is performed using a simple algorithm.

On the basis of the above considerations, this study proposes a tree-based data-collecting protocol. The protocol relies on three types of messages, i.e., *request*, *response*, and *token* messages. The *request* message is transmitted from a node to its children. This message is used to order each child to send data. The *response* message is sent from a node to its parent. The data gathered at the node and its descendants is transmitted by this message. The *token* is passed after the transmission of *request* or *response* messages is completed.

The protocol initiates when the root node sends a *request* message. Then, if a node other than the root receives a *request* with the *token* from its parent, the node executes an algorithm to send the data gathered by itself and its descendants. Here, let *n* denote a node in the network. The proposed algorithm is described in Algorithm 1.

Algorithm 1 *Data_Collection*

```
Given: list $L_n$, the children of node $n$;
1.  if $n$ is not the root
      then
2.      $p$:= the source of request; // parent of $n$
3.      Send response including the data gathered
        at $n$ to $p$;
      else
4.      Send the data gathered at $n$ to the
        application;
    end if
5.  for each $c$ in $L_n$ do
6.    Forward the request and token to $c$;
7.    Receive the responses from $c$ and put them into
      the buffer;
8.    Wait for the token to be returned from $c$;
    end for
    if $n$ is not the root
      then
9.      Forward the responses in the buffer to $p$;
10.     Return the token to $p$;
      else
11.     Send the responses in the buffer to the
        application, and wait for the next command
        from the application;
    end if
```

If a node receives a *request*, its local data is returned to the parent (Algorithm 1; line 3). Furthermore, data gathered by its descendants are collected (Algorithm 1; lines 6–7) and forwarded to the parent node (Algorithm 1; line 9). By performing these steps at each node recursively, the root can collect the data gathered by all nodes.

As described previously, each node only has to maintain a list of its child nodes, i.e., very partial information about the rooted tree.

To determine a rooted tree in a mesh topology network and obtain the children list L_n for node *n*, the proposed method employs a "*tree maker*" message. The message is issued by the root and forwarded from a node to its child node. The *tree maker* message includes a list of nodes that have not been included in the tree. Here, *U* denotes this list. If a node, denoted *n*, receives the *tree maker* message, it deletes itself from *U*. This occurs because *n* is now connected to the tree. Then, if some nodes in *U* are found to be reachable from *n*, they are added to the children of *n*. Then, *n* sends the *tree maker* to its newly discovered children, and the process is executed recursively. Finally, *n* returns the updated *U* to its parent.

This procedure adds reachable nodes to the tree in a greedy manner until all nodes are included. When a node is added to a tree, it is deleted from *U*. The deleted node will not be checked again as a child candidate of other nodes; thus, a cycle will not occur in the resulting topology. In other words, the obtained topology is assured to be a tree. This computation is summarized in Algorithm 2.

Algorithm 2 *Build_Tree*

```
1.  $L_n$:= {}; // Children of $n$
2.  if $n$ is the root
3.    then $U$:= {all nodes}
    else
4.      $U$:= node list given by the tree maker;
5.      $p$:= the source of the tree maker; // Parent
    end if
6.  $U$:= $U - \{n\}$;
7.  for each $c$ in $U$ do
8.    if $c$ is reachable from $n$ then
9.      $L_n$:= $L_n \cup \{c\}$;
```

```
10.         U:= U − {c};
        end if
    end for
11. for each c in L_n do
12.     Send tree maker to c with node list U;
13.     U:= node list replied from c;
    end for
14. if n is not the root then reply U to p;
```

Figure 4 shows an example of a tree constructed by the above algorithm. The thick lines are the links used for the tree, and the dotted lines are the links that are not used for the tree. The algorithm starts at the root node A. Initially, U is set to {A, B, C, D, E, F, G} (Algorithm 2; line 3). First, A is deleted from U (Algorithm 2; line 6). Then, lines 7–10 find that nodes B and C are reachable from A. Thus, children list L_A is set to {B, C}. Next, the algorithm is executed at B and C (Algorithm 2; lines 11–13). Since D, E, and F are reachable from B, B sets L_B to {D, E, F} and returns U = {C, G}. This list is sent to C. Then, G, which is reachable from C, is added to the tree. Node C will reply with an idle list to A. After this procedure is completed, the children lists L_A,…, L_G are set up correctly.

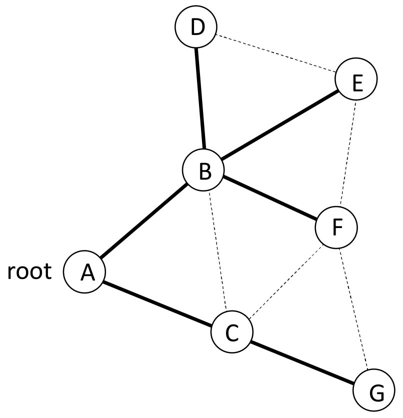

Fig. 4. Tree discovered by Build_Tree algorithm.

4. Prototype Implementation

A prototype ad hoc network was implemented to confirm the feasibility of the proposed protocol. This section describes the hardware and software of the implemented prototype.

4.1. Hardware

Each node of the network was constructed by connecting a wireless data transmission modem and a Raspberry Pi 2 Model B computer board [14], which has a BCM2836 processor and 1 GB RAM. The Linux Raspbian distribution was installed as the operating system. The computer exchanges commands and data with the modem through a universal asynchronous receiver transmitter (UART) serial interface.

This prototype included a Circuit Design, Inc. MU-1N wireless modem [4], which utilizes a 429 MHz radio wave. The proposed protocol was implemented using the following modem commands:

- Data transmission
- Packet transmission check
- Received signal strength indication (RSSI) measurement
- Node identifier setting
- Carrier sense

Each modem command is performed by sending a particular character sequence, e.g., @DT for data transmission, from the computer through the UART interface. The data size for the modem command is limited to 255 bytes. The bit rate of the wireless link is 3500 bps for user data; thus, the modem link is relatively narrow band. However, the current target application does not require a higher bit rate, and the modem specifications are adequate for this application.

A prototype node is shown in Fig. 5

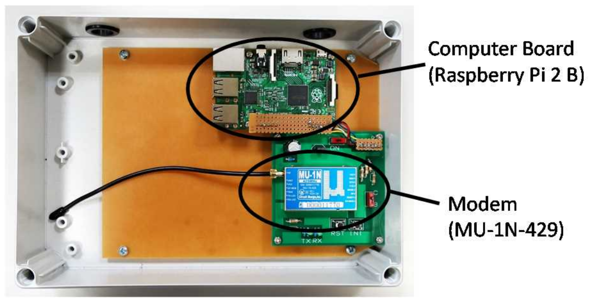

Fig. 5. Prototype node.

4.2. Software

The proposed protocol was implemented as two programs that run on the computer board. One program is run on the root node. This program waits for the command input by the operator. If the program receives a command, the data-collecting algorithm is initiated. The other program is run on nodes other than the root node. This program starts the algorithm if the node receives a *request* message. These programs were written in Perl with optional packages to control serial communication and the BCM2836 processor.

The *request*, *response*, and *token* messages are exchanged by the data transmission command provided by the modem. The packet structure of these messages is shown in Fig. 6. The first field of the packet is the message type, i.e., *request*, *response*, or *token*. The next field is the length of the option, followed by the option. Currently, the option is used to indicate the identifier of the source node. The fourth and fifth fields are used for the application. The fourth field identifies the application level command, whereas the fifth field includes the arguments given to the command.

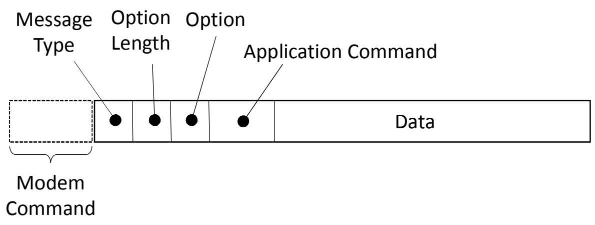

Fig. 6. Packet format.

The application level commands were developed for two purposes. The first purpose is to demonstrate the feasibility of the proposed protocol. From this perspective, we implemented several user application commands that allow the root node to collect the data gathered by each node. Second, to operate the network in real-world applications, some basic management commands were developed.

The user application commands include *get*, *watch*, *update*, and *copy*, while the management commands include *ping*, *rssi*, *build*, and *showtree*. Figure 7 illustrates the relationship among the data-collecting protocol, user applications, and management commands.

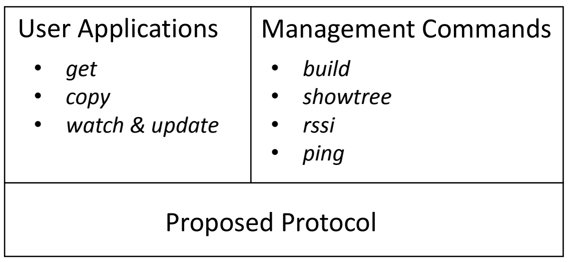

Fig. 7. Software configuration.

Among user applications, the *get* command collects and shows the content of a specified text file stored at each node. The *watch* and *update* commands are used together. The *update* command dumps the latest addition to the file being monitored at each node. The monitored file is specified through the *watch* command. The *copy* command fetches a file from a remote node and stores a copy at the root node.

Other commands are provided for management purposes. The *ping* command confirms that all nodes in the network are alive and reachable. The wireless link performance is checked using the *rssi* command. When this command is initiated, each node measures the signal and noise levels for its parent and children nodes using the modem's RSSI function. The *build* command constructs the tree structure by executing the *Build_Tree* algorithm at each node. The current tree structure can be traced using the *showtree* command, which lists the child nodes of each node in the network.

5. Experimental Results

The proposed method was examined through experiments using a small network that consisted of six prototype nodes. First, the tree construction algorithm was examined using three different node placements in our laboratory. These placements are shown in Fig. 8.

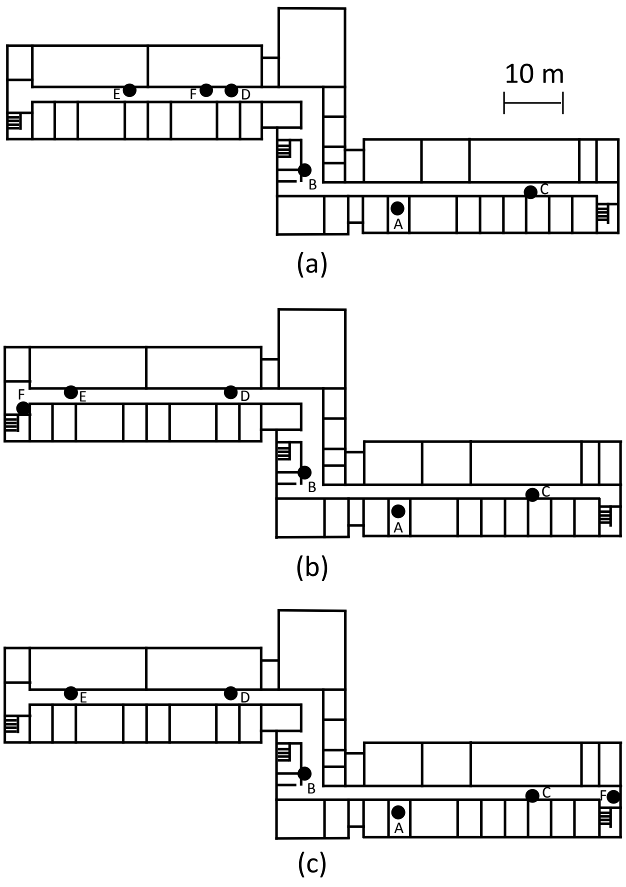

Fig. 8. Floor map and experimental node placement.

In Fig. 8, node A is the root of the tree. In every case shown in Fig. 8, only nodes B and C exist within the transmission range of the A. From B, D is reachable but E and F are not reachable. In Fig. 8(a), E and F are reachable from D, and only E is within the transmission range of D in Figs. 8(b) and 8(c). Node F is reachable from E in Fig. 8(b) and from C in Fig. 8(c).

The *build* command was executed for these node placements, and the obtained tree was extracted by the *showtree* command. Consequently, the trees shown in Fig. 9 were discovered. Figures 9(a), 9(b), and 9(c) show the trees found for the placements shown in Figs. 8(a), 8(b), and 8(c), respectively. For each case, the tree was constructed adequately. Thus, it is confirmed that the tree construction algorithm works successfully.

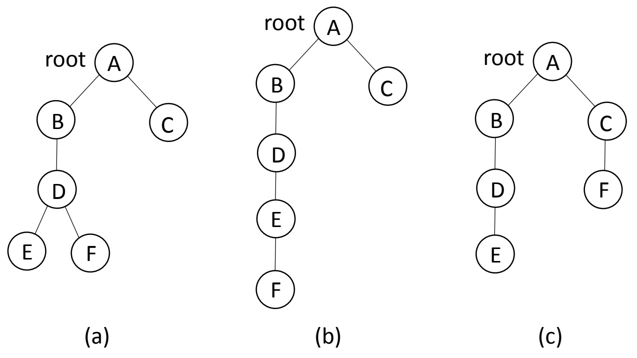

Fig. 9. Obtained trees for node placements shown in Figs. 8(a), 8(b), and 8(c).

For the trees constructed using the *build* command, all application level commands worked correctly. For example, the output of the *rssi* command is shown in Fig. 10.

Note that the response time varies depending on the command. For the configuration shown in Fig. 8(a), the response for the *showtree* command is completed within 4.2 s. The *ping* command takes slightly longer, i.e., 4.5 s because each node issues test packets to its children and waits for the replies. Note that the *build* command took 17.2 s. The response of the *build* command is slower because a time-out mechanism is used to verify node reachability.

The response time taken to complete the *copy* command was measured to evaluate the performance of the protocol. For this purpose, files of different sizes were stored on a micro-SD card of the computer. Then, the *copy* command was executed to transmit these files from a node to the root, and the elapsed time from the command input to the end of command execution was measured. The speed of reading data from the micro-SD card is much higher than the modem's bit rate; thus, its effect on performance is negligible. The file sizes were 1×10^2, 3×10^2, 1×10^3, 3×10^3, 1×10^4, 3×10^4, and 1×10^5 bytes. The tree configuration was set as shown in Fig. 11. As shown in the figure, the response time can be tested by varying the hop count from 1 to 5.

```
Node 32 -> Node 33
S: -70 dBm, N: -120 dBm

Node 32 -> Node 47
S: -71 dBm, N: -122 dBm

Node 33 -> Node 32
S: -65 dBm, N: -115 dBm

Node 47 -> Node 32
S: -72 dBm, N: -117 dBm

Node 47 -> Node 48
S: -74 dBm, N: -118 dBm

Node 48 -> Node 47
S: -73 dBm, N: -121 dBm

Node 48 -> Node 31
S: -56 dBm, N: -115 dBm

Node 48 -> Node 45
S: -56 dBm, N: -117 dBm

Node 31 -> Node 48
S: -55 dBm, N: -119 dBm

Node 45 -> Node 48
S: -56 dBm, N: -120 dBm

-- response completed --
```

Fig. 10. Command results for rssi.

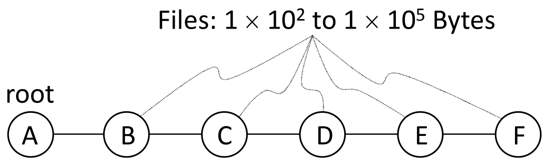

Fig. 11 Tree configuration for performance evaluation.

Figure 12 shows the response time of the *copy* command against the file size for a single hop transmission. The measurement was repeated four times, and the response time is the average. As shown in Fig. 12, the response time increases nearly proportionally with increased file size if the file is larger than 1×10^3 bytes. This suggests that the response time is limited by the transmission bit rate of the modem for a large file size. From the increased rate against transmitted data size, the estimated bit rate is approximately 3500 bps. This bit rate coincides with the actual speed noted in the modem specifications [4]. For smaller file sizes, the response time is not proportional to the file size because of the constant time required to circulate the token throughout the network.

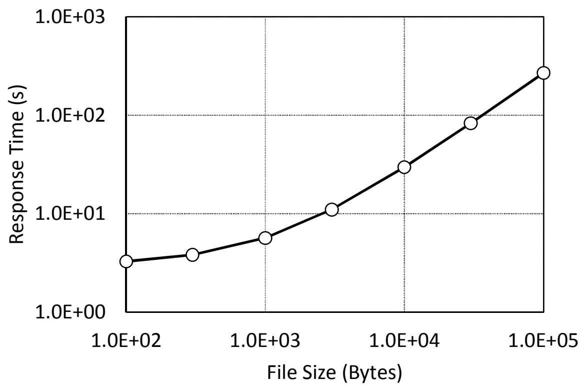

Fig. 12. Response time versus file size for single hop transmission.

Figure 13 shows the relationship between the hop count and the response time of the *copy* command for a 1×10^4 byte file. Figure 13 shows that the response time increases linearly with an increased hop count. This characteristic is obtained because data cannot be transmitted simultaneously from two or more nodes. Suppose that it takes T seconds for a node to send data for a single hop. Then, for N-hop ($N > 1$) transmission, each of the origin and $N - 1$ transit nodes requires T seconds to send the data, and the transmissions cannot be performed simultaneously. Obviously, this requires a data transfer time of NT seconds. This characteristic may be a limitation for delay-sensitive applications.

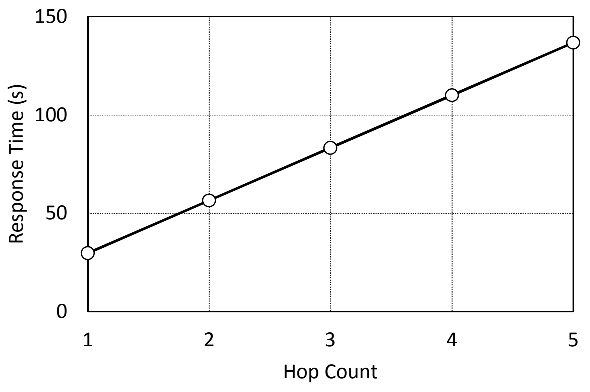

Fig. 13 Relationship between response time and hop count.

6. Related Work

A token-passing approach for data collection has been reported previously [10]. That method differs from the proposed method in that the token is circulated in order of the node identifier. This is possible because that system employs a single hop network. In the case of this study, that method is not applicable because a multihop network is used, and the routing mechanism associated with the node identifier is not available.

The convergecast technique has also been studied [7, 11, 15]. One method [7, 11] uses TDMA to avoid packet collision. Thus, the method does not efficiently handle bursty or multirate data generation because it relies on fixed time slots within a periodical frame. In addition, it is complicated to reschedule the time slot assignment against topology changes caused by node addition, deletion, or link failure.

The other convergecast method [15] employs a data-collecting protocol with emphasis on energy saving. This protocol relies on a periodical frame with fixed length, similar to TDMA. Thus, the method is inadequate for bursty or multirate data generation.

7. Conclusion

A simple data-collecting protocol has been proposed for a wireless ad hoc network. The proposed protocol provides packet collision avoidance and routing functions based on a token-passing approach and a rooted tree topology. This paper has demonstrated how the rooted tree topology is extracted from a mesh network with a simple algorithm. The feasibility of the proposed method was confirmed with a prototype that included a computer board and a data transmission modem.

The proposed protocol was developed assuming the network constructed with a 429-MHz data transmission modem. However, it is obvious that the protocol will work for any mesh networks built with other wireless transmission systems. Independently of the employed wireless system, it is expected that the proposed method has the following advantages:

- Easy implementation,
- No setting required at nodes, and
- Efficient data collecting for bursty or multirate data generation.

Therefore, the proposed protocol enables us to efficiently develop a practical data collecting system over a wireless network in a short development period.

Acknowledgment
The authors would like to thank Enago (www.enago.jp) for the English language review.

References:

[1] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice Hall, 2002.

[2] S. Misra, I. Woungang, and S. C. Misra, *Guide to Wireless Ad Hoc Networks*, Springer, 2010.

[3] M. Frodigh, P, Johansson, and P. Larsson, "Wireless ad hoc networking - the art of networking without a network," *Ericsson Review*, Vol.77, 4, pp. 248–263, April 2000.

[4] Circuit Design Inc., *MU-1N-429 Manual*, http://www.circuitdesign.jp/jp/products/products2/doc/MU-1N-429.pdf (in Japanese), 2015.

[5] A. Ghasemi, A. Abedi, and F. Ghasemi, *Propagation Engineering in Wireless Communications*, Springer, 2012.

[6] *Specific attenuation model for rain for use in prediction methods*, ITU-R Recommendation P.838-3, 2005.

[7] Ö.D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, Vol.11, 1, pp.86-99, Jan. 2012.

[8] W. Stallings, "Local network performance," *IEEE Communications Magazine*, Vol.22, 2, pp.27–36, Feb. 1984.

[9] *Token-Passing Bus Access Method*, ANSI/IEEE Standard 802.4, 1985.

[10] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," *in proc. the 2nd European Workshop on Wireless Sensor Networks*, pp. 108–120, Istanbul, Turkey, Jan. 2005.

[11] V. Annamalai, S.K.S. Gupta, and L. Schwiebert, "On tree-based convergecasting in wireless sensor networks," *in proc. IEEE Wireless Communications and Networking Conference (WCNC 2003)*, pp. 1942-1947, New Orleans, LA, March 2003.

[12] A. Tsertou and D.I. Laurenson, "Revisiting the hidden terminal problem in a CSMA/CA wireless network," *IEEE Transactions on Mobile Computing*, Vol. 7, 7, pp.817-831, July 2008.

[13] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, MIT Press, 2009.

[14] *Raspberry Pi*, https://www.raspberrypi.org/, 2015.

[15] G. Lu, B. Krishnamachari and C.S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," *Wireless Communications and Mobile Computing*, Vol.7, 7, pp. 863-875, Sept. 2007.