International Journal of Engineering and Technology Innovation, vol. 8 no. 1, 2018, pp. 01-12

Dual Polling Protocol for Improving Performance in Wireless Ad Hoc Networks

Sunmyeng Kim[1,*]

[1]Department of Computer Software Engineering, Kumoh National Institute of Technology, Gumi, South Korea.

Received 21 March 2017; received in revised form 08 June 2017; accepted 09 June 2017

Abstract

An ad hoc network is based on a distributed coordination function (DCF) to transmit data packets through channel contentions. In DCF, when the number of stations increases, performance degrades considerably because of extremely high collision probability. To improve performance, a distributed point coordination function (DPCF) protocol was proposed. In this protocol, stations operate as in DCF to obtain channel access rights. When a station obtains access rights, it polls all neighboring stations by using point coordination function (PCF). The polled stations then transmit their data packets without channel contentions. However, this protocol aggravates the issue of hidden terminals and causes channel wastage. To solve these problems, we propose a protocol in which a station polls stations in limited areas as opposed to every station in its transmission range. In addition, it polls only stations with data packets. The proposed protocol lowers the probability of collision and improves network performance.

Keywords: collision, dual polling, fairness, hidden terminal

1. Introduction

Recently, the demand for continuous connectivity of users has increased, regardless of their physical locations, and accordingly multi-hop wireless networks have been growing. Therefore, interest in and study of ad hoc networks have increased. An ad hoc network consists of many stations that transmit data packets via a wireless medium and are not centrally controlled by an infrastructure such as access points (APs). Therefore, an ad hoc network is cost-effective and easy to build. Essentially, an ad hoc network should be able to support many stations. However, because of limitations of medium access control (MAC) protocols and interference between stations, the network is not very scalable and restricted in terms of network performance. To improve the scalability and performance of the network, many studies have been conducted. One solution is to use multiple channels [1-2]. However, providing multiple channels is not easy because of a lack of radio frequency resources and high costs. In addition, it is necessary to install additional hardware so that stations can operate on multiple channels.

The IEEE 802.11 MAC protocol was designed using two methods of communication for stations: 1) distributed coordination function (DCF) and 2) point coordination function (PCF) [3]. The DCF was designed for contention-based channel access. It has two data transmission methods: default basic access and optional request-to-send/clear-to-send (RTS/CTS) access. The basic access method uses the two-way handshaking (DATA-ACK) mechanism. The RTS/CTS access method uses the four-way handshaking (RTS-CTS-DATA-ACK) mechanism to reserve the channel before transmitting long data packets. This technique was introduced to avoid the hidden terminal problem. The DCF is best known for asynchronous data transmission (or best effort service). The PCF uses a central controlled polling method to support synchronous data transmission (quality of service for real-time traffic).

---

* Corresponding author. E-mail address: sunmyeng@kumoh.ac.kr

IEEE 802.11 DCF is essentially carrier sense multiple access with collision avoidance (CSMA/CA). Packet collisions on the medium are resolved using a binary exponential backoff algorithm [4]. A station with a packet to transmit must ensure that the medium is idle before attempting to transmit. It selects a random backoff counter that is less than the current contention window based on uniform distribution and then decreases the backoff counter by one at each slot when the medium is idle. If the medium is busy, the station waits until the end of the current transmission. A station transmits a packet when its counter reaches zero.

IEEE 802.11 DCF is widely used in ad hoc networks [5-7]. However, when the number of stations increases, the performance of the MAC protocols based on IEEE 802.11 DCF protocol degrades considerably because of extremely high collision probability. To improve network performance in ad hoc networks, several MAC protocols have been proposed. In [8], the authors proposed a distributed queuing MAC protocol for ad hoc networks (DQMAN) protocol. In this protocol, the stations operate as they do in DCF to obtain channel access rights. After the channel contention, a station becomes the master and forms a cluster. In the cluster, data packets are transmitted based on DQCA protocol proposed in [9]. Adaptive MAC protocol proposed in [10] alters data transmission methods dynamically based on the traffic load of the network. When the traffic load reaches the threshold, the stations exchange DCF and dynamic time division multiple access (D-TDMA) protocols to operate. In [11], the authors used Padovan sequence to reduce the size of contention window. The reason that doubling the contention window is not always the best or optimal solution to deal with the packets collision problem. In [12], the authors redesign the message exchange process of the MAC protocol. By using a signal with shorter length, the proposed protocol can reduce the protocol overhead and thus improve the transmission reliability. In [13], the authors use a parameter, persistent probability (*x*), to decide whether a station doubles its contention window size or not after a successful transmission. After a successful transmission, a station doubles its contention window size with the probability (*x*), and resets its contention window size to the minimum value with the probability (1-*x*). In [14], the authors proposed a retransmission scheme to reduce the retransmission time, while once data transmission is fail, only the relays which have received the data frame will help for retransmission instead of repeating all retransmission cycle. In [15], the authors address the challenge of designing a multi user MAC protocol for single hop Mobile Ad Hoc Networks with the aim of allowing multiple users to transmit concurrently to increase the network spectral efficiency. To make use of the network resources more efficiently, adaptive MAC schemes, combining CSMA/CA with TDMA in a hybrid MAC frame pattern, are proposed in literature, which intend to switch between the two MAC frame structures either periodically [16, 17] or via an adaptability to a changing network traffic load [18]. The distributed point coordination function (DPCF) protocol combines the functionality of both DCF and PCF to improve network performance [19]. In the DPCF protocol, a station obtains channel access rights by using the rules of the regular DCF. When a station obtains access rights, it transmits a ready-to-send (RTS) packet to a destination station. After receiving the RTS packet, the destination station becomes the master and responds to the RTS packet with a beacon. As in the PCF, the master polls all neighboring stations. These stations then transmit their data packets without any collisions or by undergoing the channel access process.

In the DPCF protocol, a destination station, which becomes the master, operates for all neighboring stations within its transmission range. This aggravates the hidden terminal issue and further deteriorates performance. In addition, the protocol does not consider whether neighboring stations have data packets to send. It may poll neighboring stations without data packets and thus cause channel wastage.

In this paper, we propose a dual polling protocol. The proposed protocol has a similar concept to that of DPCF. In the DPCF, the destination station becomes the master and polls all neighboring stations. The polled stations then transmit their data packets without any collisions. However, in the dual polling protocol, both the source and destination stations become the master and polls all neighboring stations in the limited areas.

The main contributions of the proposed protocol are summarized as follows:

- In the DPCF protocol, the destination station becomes the master and polls its neighboring stations. And, the source station does not become the master and its neighboring stations do not possess the opportunity for data packet transmission. However, in the dual polling protocol, both the source and destination stations become the master and then poll their neighboring stations. Therefore, the dual polling protocol can solve the issue of fairness.

- In the DPCF protocol, the master polls all neighboring stations within its transmission range. Stations outside the transmission range continue the procedure of sending its own data packets. Therefore, it may further aggravate the hidden terminal problem. In the dual polling protocol, the master limits polling areas based on NAV (network allocation vector) overlap ratio. The dual polling protocol can reduce interference caused by the hidden terminal and work well in multi-hop environments.

The remainder of this paper is organized as follows. Section 2 describes the basic operational principles and issues of the DPCF protocol. Section 3 describes in detail the operations of the proposed dual polling protocol. In Section 4, performance evaluations are conducted through simulations. A conclusion is provided in Section 5.

2. DPCF Protocol

2.1. Basic operation

In the DPCF protocol, the stations operate as they do in DCF to obtain channel access rights. After its backoff counter becomes 0 and it acquires channel access rights, a source station sends an RTS packet to a destination station. Unlike in DCF, the destination station does not respond by sending a CTS packet. Instead, it sends a beacon frame to the source station. In addition, it becomes a master that acts as an AP in PCF to initiate a contention-free period. Neighboring stations receiving the beacon frame become slaves. To provide an opportunity to send data packets of the neighboring stations, the master begins polling. First, it polls the source station. After receiving the polling, the source station sends a data packet. After receiving the data packet, the master sends an ACK packet, which contains polling information about a station among slaves. Therefore, a slave that has received the polling sends its own data packet. After receiving the polling from the master, any slave that does not have any data packets to transmit, sends a NULL packet. The master repeats the aforementioned process with all slaves within its own transmission range. If no slave remains to poll, it sends a CE (CFP End) packet to all slaves to notify them of the end of the contention-free period. A station receiving the CE packet operates again in DCF.

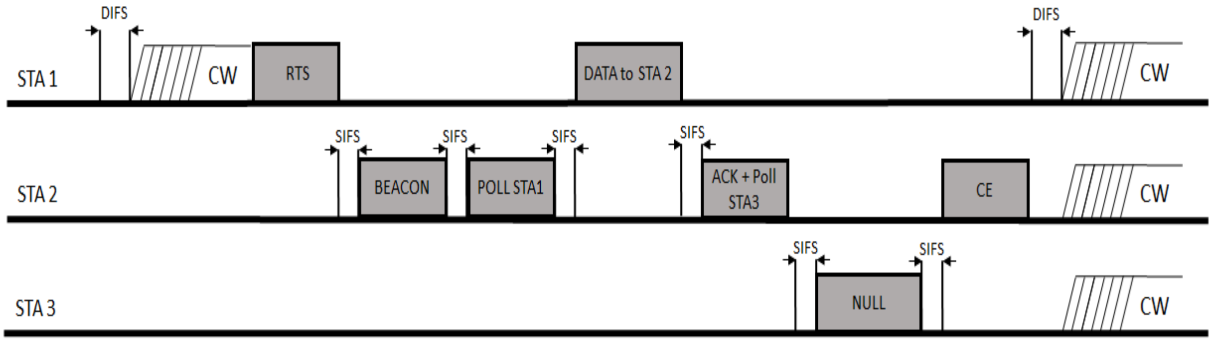


Fig. 1 Example of transmission process in the DPCF

Fig. 1 shows an example of the transmission process in the DPCF protocol. Three stations (STA1, STA2, and STA3) exist. STA1 occupies a channel in advance according to DCF and sends an RTS packet to STA2. STA2, receiving the RTS packet, becomes a master and sends a beacon frame. STA1 and STA3, receiving the beacon frame, become slaves. Initially, STA2 polls STA1, which is the source station. STA1 then sends its own data packet. After receiving the data packet, STA2 sends an ACK packet that includes polling data for STA3 as well as the reply to the data packet. Because STA3 has no data to transmit, it sends a NULL packet, and because no slave remains to poll, the master sends a CE packet to terminate the contention-free period. Neighboring slave stations receiving the CE packet re-enter the contention period.

2.2. Issues

The DPCF protocol has three issues. First, the master must regard all neighboring stations within its transmission range as its slaves and tries to poll them. Stations outside the transmission range of the master do not wait for the polling and continue the procedure of sending its own data packets. When a slave is polled and sends data packets, it may affect the outside stations that are not slaves. In other words, it may further aggravate the hidden terminal issue.

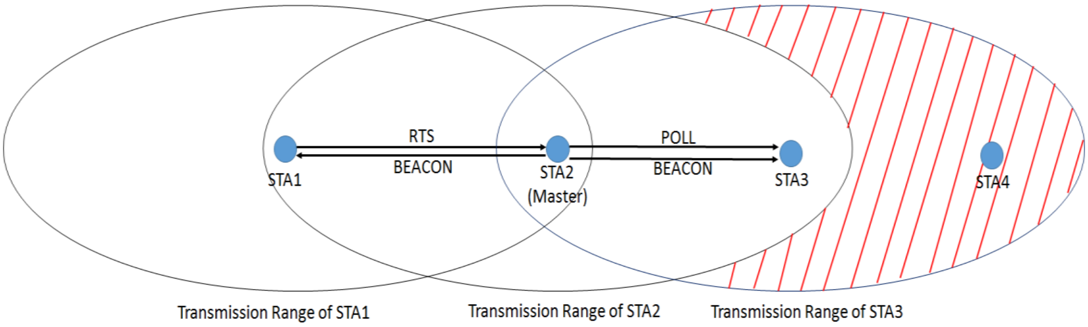


Fig. 2 Example of the hidden terminal issue in DPCF

Fig. 2 shows an example of the hidden terminal issue in DPCF. Four stations (STA1, STA2, STA3, and STA 4) exist. STA1 sends an RTS to STA2 through the contention for a channel. STA2 sends a beacon frame, and STA1 and STA3 become slaves and wait for polling. However, because STA4 is outside the transmission range of STA2, it is not a slave. If STA3 is polled and sends a data packet, it affects STA4. If STA4 communicates with another station, then it causes a collision. Likewise, although it is within the transmission range of STA3, it could affect communications in areas outside the transmission range of STA2 (e.g., the dashed area in Fig. 2).

Second, because the master does not consider whether any data packet exists to send in the queue of a slave, it may poll a slave without data packets, which may waste the channel.

Finally, in DPCF, a station receiving an RTS packet becomes the master and performs polling to stations located only within its transmission range. In this case, the stations located outside the transmission range of the master, but within the transmission range of a station sending the RTS packet do not obtain the polling. Therefore, they do not possess the opportunity for transmission. Accordingly, depending on the location of a station, the opportunities for transmission become different, which generates a concern for fairness. In other words, if bi-directional communications occur equitably and the probability of being a master is the same, then neighboring stations obtain fair opportunity for transmission. However, in the case of unidirectional communications, stations that neighbor the destination station obtain many polling opportunities, whereas other stations neighboring the source station do not have such opportunities, which produces a serious concern for fairness.

3. Proposed Dual Polling Protocol

This paper proposes a dual polling protocol, which can reduce interference caused by the hidden terminal issue and work well in multi-hop environments.

The proposed protocol has a similar concept to that of DPCF. To solve the issue of fairness in DPCF, in the dual polling protocol, the source station that sends an RTS packet as well as the destination station of that RTS packet become masters and poll their slave stations in sequence to send data packets. In this manner, it can solve the issue of fairness generated according to the direction of communications. First, a source station becomes a master. After completing polling, it switches to a destination station. Henceforth, for the purpose of classification, we use the terms source station master and destination station master.

To acquire channel access rights, a station performs the backoff process, and when the backoff counter becomes 0, an RTS packet is then sent to a destination station. The destination station sends a beacon frame to confirm and respond to the RTS packet. Neighboring stations receiving the beacon frame become slaves of the destination station and wait for polling. After receiving the beacon frame, the source station becomes a source station master and works in a similar manner to a master in the DPCF protocol. First, it sends its data packet to the destination station. The destination station replies with an ACK packet. When sending the data packet, it uses a reserved bit not used in the IEEE 802.11 data packet format to notify neighboring stations of its entry into a contention-free period. Neighboring stations receiving the RTS packet and the data packet containing the reserved bit become slaves of the source station and wait for polling. Subsequently, the source station master performs polling against its slaves and has them send and receive data packets. When the source station master completes polling against all slaves, it sends a master switching (MS) packet to the destination station to exchange to the role of master. After receiving the MS packet, the destination station becomes a destination station master and performs polling against slave stations. If no slaves remain to poll, the destination station master sends a CFP end (CE) packet to announce the termination of its contention-free period. The source station, which receives the CE packet, resends another CE packet. With the CE packet, all contention-free periods are terminated, and then the contention period restarts. After receiving RTS packets, beacon frame, MS packets, and so on, non-slave stations set their own network allocation vector (NAV) and do not participate in the communication. The duration value contained in each packet to set NAV is explained in a later section.

In the proposed dual polling protocol, a station located between the two masters is polled by each master. Therefore, it acquires more opportunities to transmit compared to other stations. To prevent this, the destination station master does not poll the stations which are polled by the source station master. The destination station master overhears polling, data, or ACK packets transmitted between the source station master and slaves, and easily knows which slaves are polled by the source station master. In this manner, each slave is polled only once.

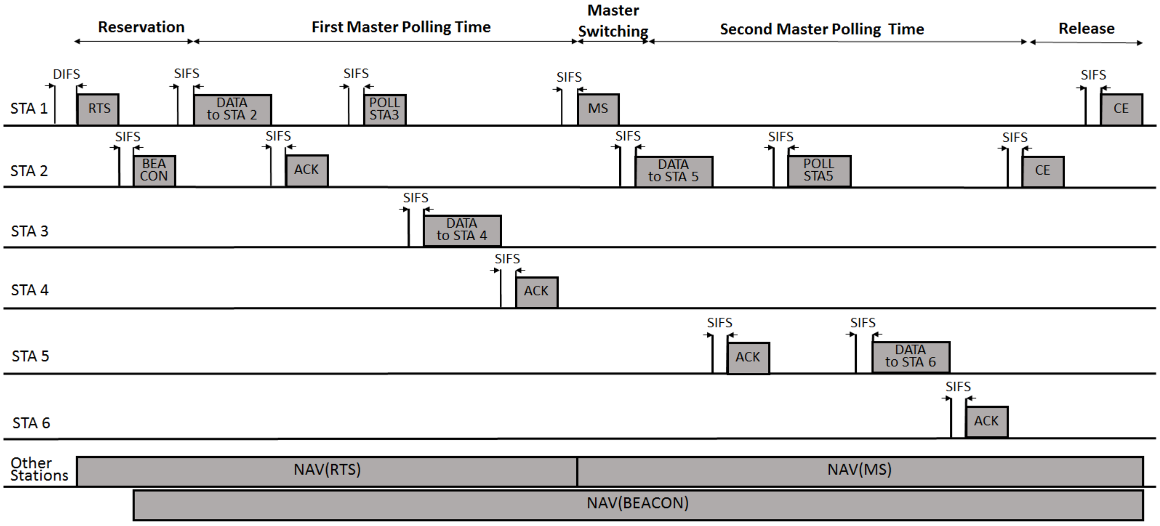


Fig. 3 Example of dual polling protocol

Fig. 3 shows the overall operations of the proposed dual polling protocol. STA1 performs contention for a channel according to DCF operations. After acquiring channel access rights, it sends an RTS packet to STA2. In addition, STA2 sends a beacon frame. Subsequently, all stations perform in the contention-free period. In other words, they send and receive data packets by means of polling. Both the source and destination stations become masters, enter the contention-free period, and perform polling to slave stations. After receiving a beacon frame, the source station master STA1 first sends its data packet to

STA2. STA2 replies with an ACK packet. Then, STA1 polls STA3. STA3 transmits a data packet to STA4, and STA4 sends an ACK packet. After completing polling to all slaves, the source station master sends an MS packet to the destination station master STA2 to switch to the role of master. STA2 polls its slaves, and when no station remains to poll, it sends a CE packet. The source station master receiving the CE packet resends the CE packet.

To reduce the influence of the hidden terminal issue, in the proposed dual polling protocol, a master station only polls the neighboring stations in the limited area. Fig. 4 shows the polling area of the DPCF and dual polling protocols. In other words, a master considers stations within the limited area as slaves. To do this, every station maintains a neighboring station table. Fig. 5 shows the table format for neighboring stations.

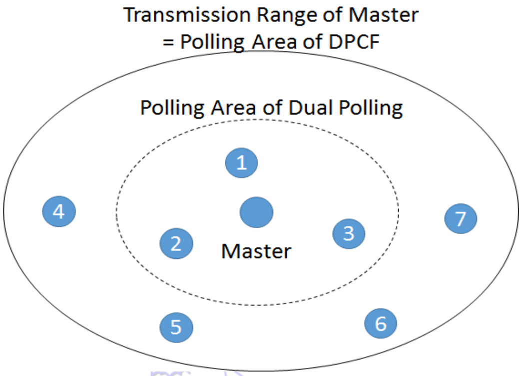
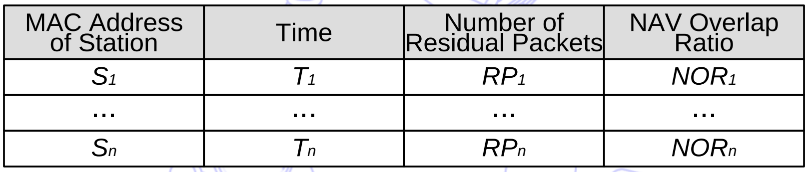


Fig. 4 Polling area of DPCF and dual polling



MAC Address of Station	Time	Number of Residual Packets	NAV Overlap Ratio
S_1	T_1	RP_1	NOR_1
…	…	…	…
S_n	T_n	RP_n	NOR_n

Fig. 5 Table format for neighbouring stations

When neighboring stations send packets such as RTS, DATA, or ACK, each station takes them and updates the neighboring station table. A neighboring station table contains four data fields. The first field contains the MAC address of a station sending packets. In the "Time" field, the time of the last packet received from each neighboring station is recorded. If no new packet is received from a station for a certain period, then the data on the station is deleted. The number of residual packets means the number of data packets contained in the queue of a neighboring station. The neighboring station must send a packet that includes the number of residual packets in the station. The NAV overlap ratio (NOR) in the last field shows the amount of NAV overlap time for each station when it becomes a slave of another master engaged in a hidden relation. This value is obtained by dividing the overlapped NAV time by the total NAV time. In this paper, we add two bytes to the packet format defined in IEEE 802.11. One byte is for the number of residual packets and the other is for the NAV overlap ratio. Stations must send packets that include these information.

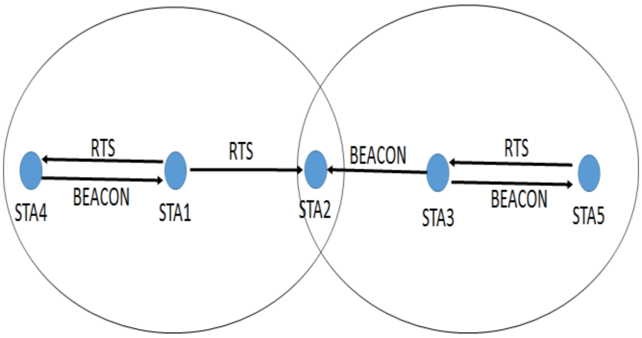


Fig. 6 Topology used to calculate the NAV overlap ratio

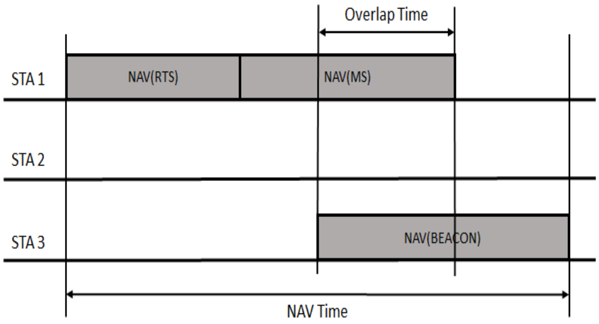


Fig. 7 Example of calculation of the NAV overlap rati

To explain the manner in which to compute NAV overlap ratio, let us consider the topology shown in Fig. 6. In the figure, both STA1 and STA3 are located in a hidden environment. STA2 is located within the transmission range of the two stations. STA1 and STA5 are the source station masters, whereas STA3 and STA4 are the destination station masters. Both STA1 and STA3 consider STA2 in their transmission range as their slave simultaneously. STA2 obtains the overlap ratio produced by STA1 and STA3 by computing the NAV time (T_N) and overlap time (T_O). The NAV of the source station master STA1 is the sum of the duration of an RTS packet and that of an MS packet ($NAV(RTS) + NAV(MS)$); whereas the NAV of the destination station master STA3 is the duration of a beacon frame ($NAV(BEACON)$) (see Fig. 7). The overlap time can be easily calculated by using the amount of time that overlaps with other NAV as shown in Fig. 7. Accordingly, T_N and NOR are represented as follows:

$$T_N = NAV(RTS) + NAV(MS) + NAV(BEACON) - T_O \tag{1}$$

$$NOR = \frac{T_O}{T_N} \tag{2}$$

To reduce the effect of the hidden environment, a master considers stations that meet the following three criteria as its slaves:

- Station i is located within the transmission range of the master.
- The number of residual packets of station i is more than one.
- $NOR_i < NOR_m + M$, where $NORm$ and $NORi$ are the NAV overlap ratio of master station m and station i, respectively. In addition, M is margin.

Hereafter, we explain the process of calculating the duration of packets such as RTS, BEACON, and MS used to set NAV.

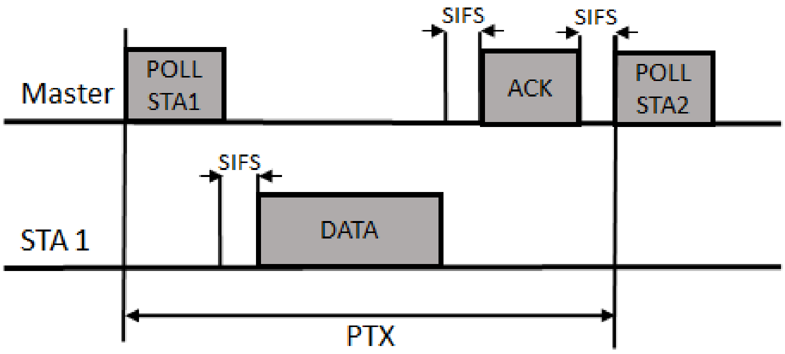


Fig. 8 Definition of PTX

The RTS duration (D_{RTS}) is the sum of transmission time of the RTS packet (T_{RTS}), beacon transmission time (T_{BEACON}), transmission time of a data packet of the source station master (T_{DATA}), ACK transmission time (T_{ACK}), SIFS time ($SIFS$), and packet transmission time of the slaves (T_{SLAVE}) (see Fig. 3). First, the packet transmission time (PTX) of a slave is the sum of the polling packet transmission time (T_{POLL}), data packet transmission time, ACK transmission time, and SIFS time (see Fig. 8). PTX is defined as follows:

$$PTX = T_{POLL} + T_{DATA} + T_{ACK} + 3 \cdot SIFS \tag{3}$$

The transmission time of all slaves is defined as follows:

$$T_{SLAVE} = n \cdot PTX \tag{4}$$

where n is the number of slave stations determined by the master with the three slave criteria.

The duration of RTS is defined as follows:

$$D_{RTS} = T_{RTS} + T_{BEACON} + T_{DATA} + T_{ACK} + 4 \cdot SIFS + T_{SLAVE} \tag{5}$$

The beacon duration (D_{BEACON}) is the sum of the RTS duration (excepting T_{RTS}), the MS packet transmission time (T_{MS}), the transmission time of slaves to be polled by the destination station master (T_{SLAVE}), and the CE packet transmission time (T_{CE}).

$$D_{BEACON} = D_{RTS} - T_{RTS} + T_{MS} + T_{SLAVE} + 2 \cdot T_{CE} + SIFS \tag{6}$$

The duration of an MS packet (D_{MS}) is defined as follows:

$$D_{MS} = D_{BEACON} - D_{RTS} + T_{RTS} + SIFS \tag{7}$$

4. Simulation Results

In this section, we analyze the performance of the proposed dual polling and the DPCF protocols based on simulation results. We have implemented them in C++. The parameters used in the simulation are listed in Table 1. The simulation was conducted based on IEEE 802.11g in which the transmission rate of data packets was 54 Mbps, and that of control packets such as RTS, beacon, polling, and ACK was 6 Mbps. In the simulation, the length of a data packet was 1000 bytes.

Table 1 Simulation parameters

Parameter	Value
Data Rate	54 Mbps
Control Rate	6 Mbps
Slot Time	9 us
SIFS	16 us
DIFS	34 us
Propagation Delay	1 us
MAC Header	26 Bytes
FCS	4 Bytes
ACK	14 Bytes
CWmin	31
CWmax	1023

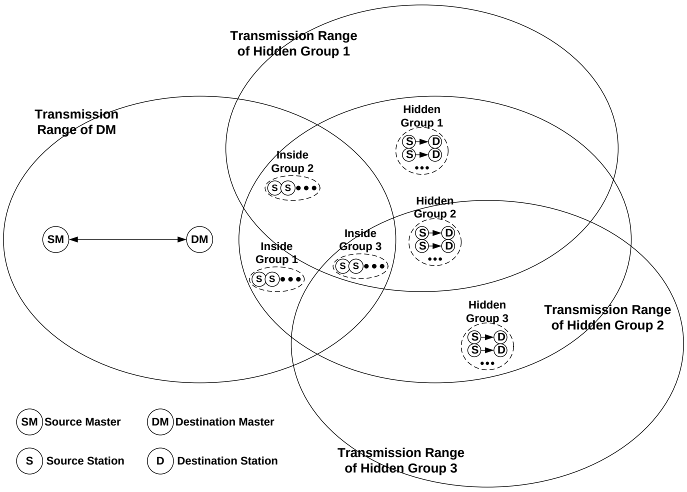


Fig. 9 Simulation topology

Fig. 9 shows the topology used in the simulation. As shown in the figure, the simulation was performed by focusing on the source master (SM) and destination master (DM) stations. SM became a source station master by acquiring channel access rights through the backoff process, and the DM became a destination station master. Stations within the transmission range of the DM were divided into three groups (Inside Groups 1, 2, and 3). Stations outside the transmission range were also divided into three groups (Hidden Groups 1, 2, and 3). Stations in an inside group sent data packets to the DM only when they received polling packets from the DM. Stations in a hidden group sent data packets to their own destination station. Inside Group 1 was within the transmission range of Hidden Group 2, and Inside Group 2 was within those of Hidden Groups 1 and 2, whereas Inside Group 3 was within those of Hidden Groups 1, 2, and 3.

Figs. 10-12 show the results simulated in an environment in which each inside group have four stations and the hidden groups have the same number of stations. In the figures, when the number of stations on the X axis is six, this means that six source and six destination stations exist, as well as six flows in the hidden groups. Therefore, in each hidden group, two flows exist.

Fig. 10 shows the normalized throughput of stations in the inside groups. We could confirm that regardless of the number of stations in a hidden environment, the throughput of both the dual polling and DPCF protocols was constant. The DM was not affected by stations in the hidden environment. When the DM became the destination station master, it performed polling on stations in the inside groups according to its own schedule. A station in an inside group sent a data packet when it was polled. In addition, the transmitted data packets were transferred to the DM without any collision occurring. Therefore, even if the number of stations in the hidden environment increased, the throughput of an inside group was not affected and remained constant. In our study, we observed that the throughput of the DPCF protocol was slightly better than that of the dual polling protocol. In the DPCF protocol, every station in an inside group was polled, but in the proposed dual polling protocol, stations only within limited areas of an inside group were polled. Thus, performance of the DPCF protocol was better. However, the difference in performance was not considerable.

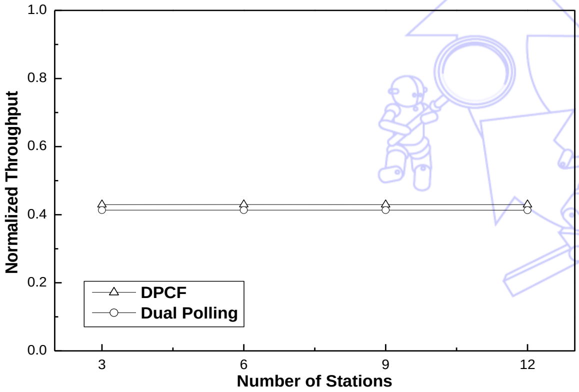


Fig. 10 Throughput of stations in inside groups based on the number of stations in the hidden environment

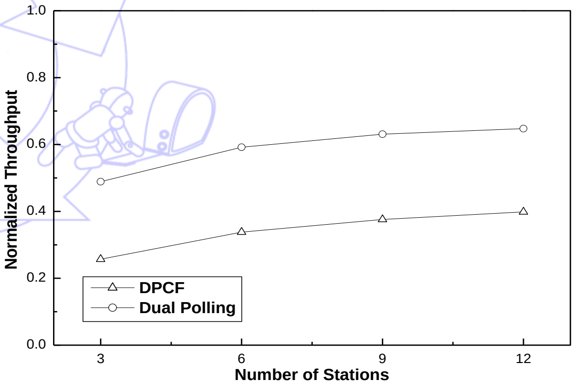


Fig. 11 Throughput of stations in hidden groups based on the number of stations in the hidden environment

Fig. 11 shows the normalized throughput of stations in the hidden groups. Unlike the results shown in Fig. 10, the throughput, improved with an increasing number of stations. When the number of stations in the hidden groups increased, the number of data packets to send also increased, which yielded improved performance. This reveals that the throughput of the proposed dual polling protocol is far superior to that of DPCF. In the proposed dual polling protocol, the DM polled stations only in the limited areas of inside groups. Accordingly, the proposed protocol's influence on stations in the hidden environment was not considerable. However, in the DPCF protocol, the DM performed polling against all stations in the inside groups, and corresponding stations sent data packets. Thus, the effect on the hidden environment was sufficiently large. Although the proposed dual polling protocol caused a slight decrease of throughput for the inside groups, as shown Fig. 10, it improved considerably the throughput of those in the hidden environment.

Fig. 12 shows the probability that stations in the hidden groups collide. Because collisions occurred essentially because of the data transmission of stations in hidden groups, the probability of collision increased in both the DPCF protocol and dual polling protocol following an increase in the number of hidden stations. In addition, when stations in the inside groups sent data packets, collisions occurred. In the proposed dual polling protocol, polling operations were performed only on stations in the limited areas of the inside groups. Therefore, it minimized collisions with data packets of hidden stations. However, in the DPCF protocol, every station in the inside groups was polled; thus, collisions with hidden stations are more likely. Therefore, the proposed dual polling protocol shows much less probability of collision than does the DPCF protocol.

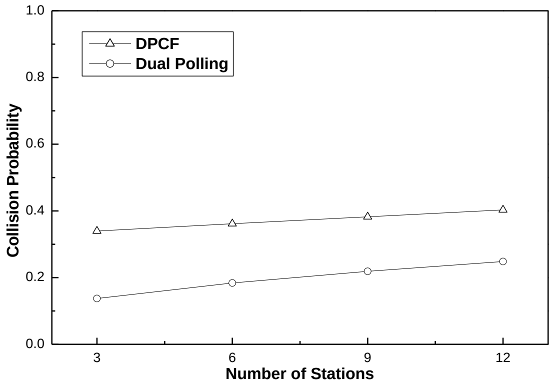


Fig. 12 Probability of stations colliding in the hidden groups based on the number of stations in the hidden environment

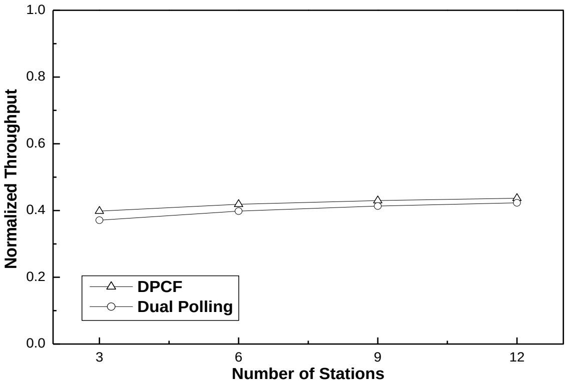


Fig. 13 Throughput of stations in inside groups based on the number of stations in inside groups

Figs. 13-16 show the results from a simulated environment in which each hidden group had two pairs of source and destination stations. In other words, each group had two flows. In addition, each inside group had the same number of stations. In the figure, when the number of stations on the X axis is six, this means two stations are present in each inside group.

Fig. 13 shows the normalized throughput of stations in inside groups. In both the dual polling and DPCF protocols, we confirmed that as the number of stations in inside groups increased, the throughput improved in small increments. In addition, the throughput of the DPCF protocol was slightly better than that of the dual polling protocol. In the DPCF protocol, every station in the inside groups was polled. However, in the proposed dual polling protocol, only stations within limited areas of the inside groups were polled. Thus, performance of the DPCF protocol was better. However, the difference in performance was not considerable.

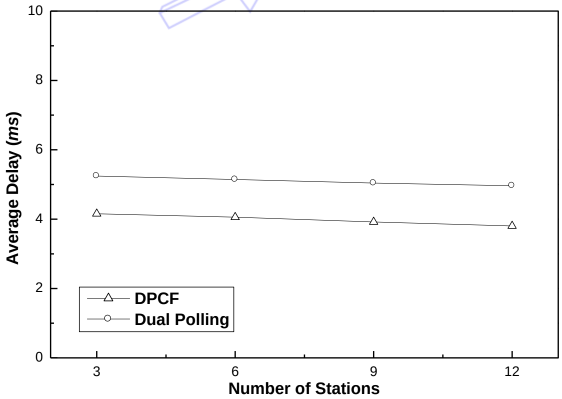


Fig. 14 Average delay of stations in inside groups based on the number of stations in inside groups

Fig. 14 shows the average delay of stations in inside groups. The proposed dual polling protocol shows the average delay always higher than that of the DPCF regardless of the number of stations. In the proposed dual polling protocol, the DM polled stations only in the limited areas of inside groups. In the DPCF protocol, the DM performed polling on all stations in the inside groups. Therefore, stations in the DPCF can send their data packets faster than those in the dual polling. Even though the proposed protocol has higher delay than the DPCF protocol, it does not matter since the delay is very low and can meet quality of service (QoS) requirements.

Fig. 15 shows the normalized throughput of stations in hidden groups. Unlike the results shown in Fig. 13, throughput deteriorated progressively with an increasing number of stations. This reveals that the throughput of the proposed dual polling protocol is far superior to that of DPCF. In the proposed dual polling protocol, the DM polled stations only in the limited areas of inside groups. Accordingly, its influence on stations in the hidden environment was not considerable. However, in the DPCF protocol, the DM performed polling on all stations in the inside groups, and corresponding stations sent data packets. Thus, the effect on the hidden environment was sufficiently large. Although the proposed dual polling protocol caused a slight decrease of throughput for the inside groups, it improved the throughput of those in the hidden environment considerably.

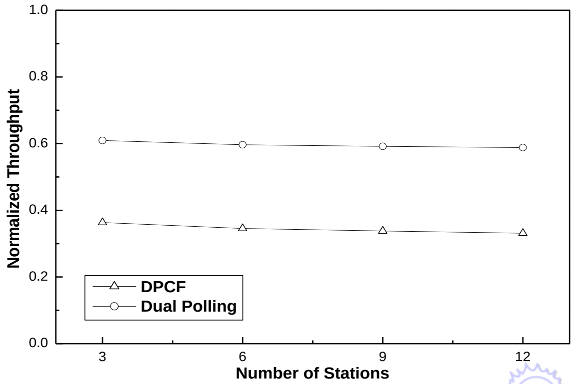


Fig. 15 Throughput of stations in hidden groups based on the number of stations in inside groups

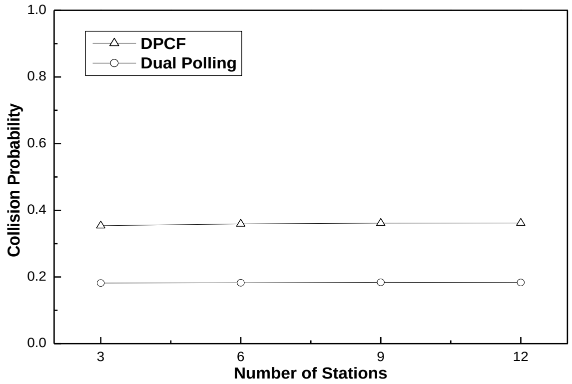


Fig. 16 Probability of stations colliding in hidden groups based on the number of stations in inside groups

Fig. 16 shows the probability of stations colliding in hidden groups. We confirmed that as the number of stations in inside groups increased, the probability of collision also increased, but only slightly. In the proposed dual polling protocol, polling operations were performed only on stations in the limited areas of inside groups. Therefore, it minimized collisions with data packets from hidden stations. However, in the DPCF protocol, every station in the inside groups was polled. Therefore, it increased collisions with hidden stations. Thus, the proposed dual polling protocol showed much less probability of collision than did the DPCF protocol.

5. Conclusions

In a wireless ad hoc network, as the number of stations increases, the probability of collision also increases. This causes performance to deteriorate rapidly. To improve performance, the DPCF protocol was proposed. In this protocol, the destination station becomes the master and polls its neighboring stations by using PCF. However, the DPCF protocol has three issues: fairness problem, hidden terminal problem, and channel wastage. To solve the three issues of the DPCF protocol and improve performance, in this paper, we proposed a dual polling protocol to improve performance. In the proposed protocol, a station basically runs in DCF. When it then acquires channel access rights, it performs polling operations on some stations within limited areas instead of on all stations in the transmission range. In addition, polling is performed on stations with data packets. The polled stations, then transmit their data packets without channel contentions. Through these operations, the proposed protocol lowers the probability of collision and improves performance in a multi-hop ad hoc network with a hidden environment. In future works, we would like to study the effect of coexisting legacy DCF stations and dual polling stations on performance. And then, we seek how they can coexist smoothly.

Acknowledgments

This paper was supported by Kumoh National Institute of Technology.

References

[1] K. H. Almotairi and X. Shen, “A distributed multi-channel MAC protocol for ad hoc wireless networks,” IEEE Transactions on Mobile Computing, vol. 14, no. 1, pp. 1-13, January 2015.

[2] K. Hafeez, L. Zhao, J. Mark, X. Shen, and Z. Niu, “Distributed multichannel and mobility aware cluster-based MAC protocol for vehicular ad-hoc networks,” IEEE transactions on vehicular technology, vol. 62, no. 8, pp. 3886-3902, October 2013.

[3] IEEE 802.11 working group, “Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” IEEE September. 802.11, 2012.

[4] D. Basu, G. S. Gupta, G. Moretti, and X. Gui, “Performance comparison of a new non-RSSI based wireless transmission power control protocol with RSSI based methods: experimentation with real world data,” International Journal of Engineering and Technology Innovation, vol. 6, no. 1, pp. 30-54, January 2016.

[5] P. C. Ng and S. C. Liew, “Throughput analysis of IEEE802.11 multi-hop ad hoc networks,” IEEE transactions on networking, vol. 15, no. 2, pp. 309-322, April 2007.

[6] H. Xu and L. Lei, “A MAC protocol based on busy tone for ad-hoc networks with beamforming antennas,” Proc. international conf. systems and informatics (ICSAI 2016), IEEE Press, January 2016.

[7] Q. Ye, W. Zhuang, L. Li, and P. Vigneron, “Traffic-load-adaptive medium access control for fully connected mobile ad hoc networks,” IEEE transactions on vehicular technology, vol. 65, no. 11, pp. 9358-9371, November 2016.

[8] J. Alonso-Zarate, E. Kartsakli, L. Alonso, and C. Verikoukis, “Performance analysis of a cluster-based MAC protocol for wireless ad hoc networks,” EURASIP journal on wireless communications and networking, vol. 2010, pp. 1-16, March 2010.

[9] J. Alonso-Zarate, E. Kartsakli, A. Cateura, C. Verikoukis, and L. Alonso, “A near-optimum cross-layered distributed queuing protocol for wireless LAN,” IEEE wireless communications, vol. 15, no. 1, pp. 48-55, February 2008.

[10] Q. Ye, W. Zhuang, L. Li, and P. Vigneron, “Traffic load adaptive medium access control for fully-connected mobile ad hoc networks,” IEEE transactions on vehicular technology, vol. 65, no. 11, pp. 9358-9371, January 2016.

[11] D. Moad, S. Djahel, and F. Nait-Abdesselam, “Padovan sequence based backoff algorithm for improved wireless medium access in MANETs,” Proc. global information infrastructure and networking symposium (GIIS), September 2014, pp. 1-6.

[12] Q. T. Hoang and X. N. Tran, “Improved cross-layer cooperative MAC protocol for wireless ad hoc networks,” Proc. Signal and Information Processing Association Annual Summit and Conference (APSIPA), December 2014, pp. 1-6.

[13] H. H. Chin, C. C. Lin, and D. J. Deng, “Contention resolution algorithm for MAC protocol in wireless ad-hoc networks,” Proc. Wireless Communications and Mobile Computing Conference (IWCMC), August 2012, pp. 856-859.

[14] J. P. Sheu, J. T. Chang, C. Ma, and C. P. Leong, “A cooperative MAC protocol based on 802.11 in wireless ad hoc networks,” Proc. Wireless Communications and Networking Conference (WCNC), April 2013, pp. 416-421.

[15] H. Saggar, Y. Jiang, B. Daneshrad, and G. Pottie, “A concurrent CSMA MAC protocol for mobile ad hoc networks using beamnulling,” Proceedings of Military Communications Conference (MILCOM), October 2015, pp. 97-102.

[16] R. Zhang, L. Cai, and J. Pan, “Performance study of hybrid MAC using soft reservation for wireless networks,” 2011 IEEE International Conf. Communications (ICC), IEEE Press, July 2011, pp. 1-5.

[17] R. Zhang, L. Cai, and J. Pan, “Performance analysis of reservation and contention-based hybrid MAC for wireless networks,” 2011 IEEE International Conf. Communications (ICC), IEEE Press, July 2010, pp. 1-5.

[18] W. Hu, H. Yousefi’zadeh, and X. Li, “Load adaptive MAC: a hybrid MAC protocol for MIMO SDR MANETs,” IEEE transactions on wireless communications, vol. 10, no. 11, pp. 3924-3933, November. 2011.

[19] J. Alonso-Zarate, C. Crespo, C. Skianis, L. Alonso, and C. Verikoukis, “Distributed point coordination function for IEEE 802.11 wireless ad hoc networks,” Ad hoc networks, vol. 10, no. 3, pp. 536-551, May 2012.