Learning based relay selection for reliable content distribution in smart class application

Taehong Kim[1]
[1]Electronics and Telecommunications Research Institute (ETRI)
Daejeon, Republic of Korea
[e-mail: taehong.kim@etri.re.kr]
*Corresponding author: Taehong Kim

Received February 10, 2015; revised May 19, 2015; revised June 4, 2015; accepted June 24, 2015; published August 31, 2015

Abstract

As the number of mobile devices such as smart phones and tablets explodes, the need for new services or applications is also rapidly increasing. Smart class application is one of the emerging applications, in which most of contents are distributed to all members of a class simultaneously. It is highly required to select relay nodes to cover shadow area of radio as well as extend coverage, but existing algorithms in a smart class environment suffer from high control packet overhead and delay for exchanging topology information among all pairs of nodes to select relay nodes. In addition, the relay selection procedure should be repeated in order to adapt to the dynamic topology changes caused by link status changes or device's movement. This paper proposes the learning based relay selection algorithm to overcome aforementioned problems. The key idea is that every node keeps track of its relay quality in a fully distributed manner, where RQI (Relay Quality Indicator) is newly defined to measure both the ability of receiving packets from content source and the ability of successfully relaying them to successors. The RQI of each node is updated whenever it receives or relays broadcast packet, and the node having the higher RQI is selected as a relay node in a distributed and run-time manner. Thus, the proposed algorithm not only removes the overhead for obtaining prior knowledge to select relay nodes, but also provides the adaptability to the dynamic topology changes. The network simulation and experimental results prove that the proposed algorithm provides efficient and reliable content distribution to all members in a smart class as well adaptability against network dynamics.

Keywords: broadcasting, content distribution, relay node, relay selection, wireless ad-hoc networks, smart class

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP). (B0101-15-233, Smart Networking Core Technology Development)

http://dx.doi.org/10.3837/tiis.2015.08.008 **ISSN : 1976-7277**

1. Introduction

As the number of mobile devices such as smart phones and tablets are explosively produced recently, many kinds of applications are also rapidly increased. One domain of emerging applications is content sharing among smart devices, for example, user freely shares contents such as documents, music, picture, and videos with other group members through any kinds of network interfaces such as 3G/4G cellular network, Wi-Fi, and Bluetooth. It is also possible that smart device can discover a local printer connected wirelessly with internet and transfer images to the printer.

As well as above individual use of content sharing, some kinds of public applications, such as smart class [1, 2] and e-meeting [3], have recently realized. Smart class application assumes that all the students as well as teacher have equipped with smart devices like tablet. The educational contents such as the lecture file, audio-video materials, and notes on white board are shared to the students in a class. In addition, teacher can check attendance of students, control and monitor students' devices, and support real-time questions and answers. Existing commercialized solutions have developed with wireless AP (Access Point), in other words, AP has a role of receiving educational contents from a teacher and redistribute to all members in a class.

However, considering that most of classes have not equipped with wireless AP yet, such content sharing can be supported by wireless ad-hoc networking. [4] For example, teacher's device can distribute its contents by broadcasting without any help from wireless AP, and most of students' devices can receive them. To cover the devices located outside of transmission range of the teacher's device, it is required for some students' devices to relay the received contents.

Even though above mechanism seems very similar with traditional broadcasting algorithms in wireless multi-hop networks, existing algorithms [5-15] cannot be applied due to several characteristics of smart class environment. First, existing algorithms require topology information about at least 2-hop neighbors in order to select relay nodes covering the whole topology. Since the time and control packet overhead increase proportionally to the network density, it causes a long waiting time to start smart class application which composed of dense network devices. Second, existing algorithms need to repeat topology information gathering and reselection of relay nodes in order to adapt to the dynamic topology changes caused by link status change or device movement. Same with the initial phase, the reselection procedure also requires a long time to collect prior topology knowledge, and it may interfere with seamless content distribution as well as lecture itself. Thus, a novel broadcast relay selection algorithm is required to overcome aforementioned problems by considering the characteristics of the smart class application. The followings are the summary of three requirements (R) and two characteristics (C) of the smart class application.

- ***R1 - Fast and reliable content sharing:*** To support streaming contents, it is required to design fast and reliable broadcast algorithm. Since the control packet overhead consumes the bandwidth, the number of control packet during content distribution should be minimized.
- ***R2 - No delay for initial setup:*** Application users may not willing to wait several minutes until relay nodes are selected and ready to forward contents. However, it is inevitable that

existing algorithms require some time for obtaining topology information to select relay nodes covering the whole topology.

- ***R3 - Need adaptability to network dynamics:*** Teacher and students can move with smart devices, and thus, the relay selection algorithm should be adaptable to network dynamics such as noise traffic and topology changes. In addition, the delay to reselect the relay nodes should be minimized in order to allow seamless content distribution during lecture.
- ***C1 - Maximum 2-hop coverage***: Even though the content source's transmission range is enough to cover most of small classroom, it is highly required for relay nodes to cover border area that is not covered by the content source as well as to extend the coverage for large area such as lecture room and auditorium. However, the maximum number of hops is limited by 2-hop by considering the high quality of service like content streaming, because it is already proved by [19] that the performance of multi-hop communication significantly decreases from 3-hops.
- ***C2 - Dense network:*** The smart class application supports up to several hundreds of devices considering large lecture room. Note that most of devices located within the content source's transmission range can be relay candidates. However, it is not realistic to apply the existing algorithms exchanging connectivity information among all pairs of relay candidates into the smart class application, since it may cause severe control packet among the relay candidates. Therefore, the new relay selection algorithm should minimize the prior knowledge for selecting relay nodes by considering the dense network topology.

To satisfy above requirements as well as provide efficient and reliable content distribution in smart class application, this paper proposes the learning based relay selection algorithm. The key idea is that every node keeps track of its relay quality in a fully distributed manner, where RQI (Relay Quality Indicator) is defined as a metric to measure both the ability of receiving packets from content source and the ability of successfully relaying them to successors. Since RQI of each node is learned from real data traffic, it does not require any time or control packet overhead for selecting relay nodes. Moreover, the proposed algorithm allows nodes to compare with the current relay nodes' RQI contained in the relayed packet in a distributed manner, and the nodes having the highest RQI are selected as relay nodes. Since RQI is designed to be increased as much as relaying the packets successfully, and on the contrary, it is decreases as amount of the failure of relays. Therefore, the relay node can be replaced promptly with the promising node having higher RQI in order to adapt to the network dynamics such as noise traffic, node removal, and movement.

This paper is organized as follows. Section 2 summarizes the related works on the broadcasting algorithms in wireless ad hoc networks, and section 3 describes the proposed learning based relay selection algorithm. Section 4 and 5 evaluate the performance of the proposed algorithm through network simulation and testbed implementation, respectively, and section 6 concludes this paper.

2. Related Work

There have been many kinds of researches on broadcasting algorithms in wireless multi-hop networks during several decades. The broadcasting algorithms can be divided into three major topics such as redundancy, reliability, and latency of broadcasting. [5] Among these three topics, the redundancy problem indicates that there occur throughput degradation, packet collision and loss from many times of duplicate rebroadcasting. This problem can be alleviated by selecting a small set of relay nodes set.

The random based relay selection algorithm [5, 6] allows each node to rebroadcast the received packet with the given priority *p*. If *p* is 1, this scheme is same with the simple flooding algorithm in which all nodes rebroadcast a received packet once, when they receive a broadcast packet for the first time. Note that even though it cannot guarantee that rebroadcasted packets cover all the network coverage, it effectively alleviates the redundancy of rebroadcasting packets without any prior knowledge to select relay nodes. By focusing on that probabilistic broadcasting gives benefit such as low overhead and robustness against failures and mobility of nodes, Reina et al. [6] has classified and analyzed the probabilistic broadcasting schemes suggested during last decade. The authors has lessened that there is no global optimal solution, since each algorithm depends on many topological parameters such as density, mobility, and scenario. Most of all, the probabilistic schemes cannot provide reliability of broadcasting due to the lack of consideration for coverage problem.

One of the representative relay selection algorithms is MPR (Multipoint Relays) [7, 8], which is standardized for IETF OLSR on ad hoc network. In MPR, each node selects the relay nodes within its 1-hop neighbors that can cover 2-hop neighbor nodes, requiring additional time and control packets to obtain topology information about at least 2-hop neighbor information. Despite control packet overhead, OLSR protocol is widely used in IEEE 802.11 based mobile ad hoc networks, and the analytical models and performance analysis on QoS, end-to-end throughput, and delay in IEEE 802.11 based multi-hop networks are provided by [20-22]. Recently, a new MPR selection algorithm [8] has proposed to cover all 2-hop neighbor nodes m times for the robustness of broadcasting. The main idea is to select more auxiliary MPR nodes as well as main MPR nodes by considering the redundancy count for 2-hop neighbor nodes. The proposed algorithm has enhanced the reliability of broadcasting, however, it has inherited large control packet overhead from MPR.

Collective flooding [9, 10] proposes to solve both of broadcast redundancy and reliability problems. The main idea of collective flooding algorithm is link correlation. In detail, it allows the sender to infer the success of transmission to a receiver based on the acknowledgement from other neighboring receivers. Moreover, relay nodes are dynamically selected according to the coverage probability updated based on the collective acknowledgement. The link correlation between communication links also used in LBC (Light-weight broadcast) [11]. Instead of adjusting the backoff timing in the collective flooding, a source or an intermediate node in LBC retransmits a packet until every neighbor acknowledges the reception or a maximum number of retries is reached, and selects the best forwarder nodes among every possible combination of neighbors based on measured link qualities. Note that both collective flooding and LBC require exchanging link status messages among all pairs of 1-hop neighbor nodes for obtaining link correlation information used for relay node selection.

Similar with collective flooding, UFlood [12] is designed for each node to dynamically choose the sender by comparing the additional coverage of neighboring nodes. UFlood also exploits the network coding technique [13] and bit rate selection, but it still requires prior knowledge about network connectivity and link status information. Moreover, the prior knowledge for selecting relay nodes both in collective flooding and UFlood cannot be reused and should be obtained again, when there occur link status or topology changes.

It is interesting to note that above relay selection algorithms [7-12] commonly require link status exchange among all pairs of 1-hop neighbors. Especially, theoretical algorithms in [14, 15] require gathering of whole topological information, not 1-hop neighbors, even though they can provide optimal solution for selecting relay nodes. However, as mentioned as one of the characteristics of smart class application, most of devices in a class are within the content source's transmission range. All these device should exchange the link status messages to obtain their 2-hop neighbors' coverage information, and it may results in tremendous number of control packets and high delay until selecting relay nodes. Moreover, this relay selection procedure should be repeated, whenever network topology is changed.

There are also several researches focusing on content dissemination in wireless ad hoc networks. Li et al [16] proposed the relay selection algorithm assuming that not all the nodes in a network are interested in the content. They choose the relay nodes to filter out unnecessary data transmission as well as cover all the nodes interested in the content. The paper [17] has analyzed the capacity enhancement of cache enabled distribution in wireless ad hoc networks by considering two fundamental content access schemes such as nearest caching node scheme and enroute caching scheme, and the paper [18] proposed the popularity based adaptive content delivery with in-network caching scheme in content delivery network. The above in-network caching mechanism is useful for increasing network capacity, but it is not affordable to support fast content dissemination into all the nodes in a network due to its inherent feature.

Different from existing algorithms [7-15], the proposed relay selection algorithm does not require any additional control packets and time to collect topology information, which is necessary for selecting relay nodes. Instead, each relay candidate learns its relay quality in real time and distributed manner, and the node with the highest relay quality is dynamically selected as relay node. Since the relay quality metric measures link quality as well as network coverage by utilizing NACK (Negative Acknowledgement) packet, the proposed algorithm achieves reliability of broadcasting by keeping the number of relay nodes to minimum. Another contribution of the learning based relay selection algorithm is that the relay nodes can be changed dynamically in order to adapt to the network topology changes. To the best of knowledge, there is no learning based relay selection algorithm in wireless ad hoc networks. Note that this would not be possible with the analysis on the characteristics and requirements of the smart class application.

3. Learning based Relay Node Selection Algorithm

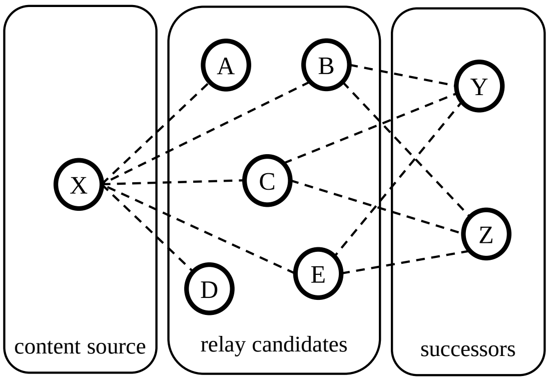


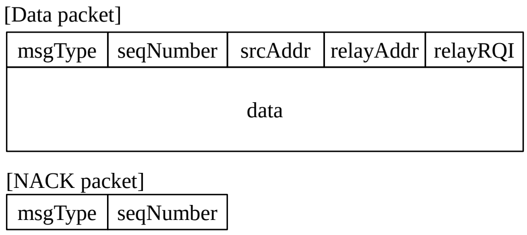
Fig. 1. Network topology of a smart class application

Fig. 1 shows an example of network topology in a smart class application. Once the content source transmits packets, the nodes within the transmission range of the content source, A, B, C, D, and E in **Fig. 1**, receive the packets. These nodes are denoted as relay candidates, and one of relay candidates should be selected to rebroadcast the received packets. Note that node A and D cannot deliver the packets to successsors, since they do not have network connectivity to the successors. Thus, only when one of nodes B, C, and E relays the packets, the packet can be successfully delivered to the successors.

The main goal of this paper is to select relay nodes while satisfying three requirements of smart class application and two characteristics described in section 1. Followings are the brief overview on how to achieve the requirements by using smart class's own characteristics.

- ***R1 - Fast and reliable content sharing:*** The proposed algorithm chooses only one relay node that covers neighboring area in order to maximize the network bandwidth. Moreover, the main criterion of RQI is that how well a relay node successfully delivers the packets to its successor nodes.
- ***R2 - No delay for relay node selection:*** A relay node is selected without collecting prior knowledge. Thus, it does not require any training time to select a relay node.
- ***R3 - Need adaptability to network dynamics:*** RQI is updated based on the unit of a broadcast packet, and a relay node can be reselected whenever a new broadcast packet is transmitted and received. Thus, it is easy to adjust a relay node against the network dynamics, such as noise traffic and topology changes caused by nodes' leaving or movement.
- ***C1 - Maximum 2-hop coverage:*** Instead of obtaining topology information to select relay nodes, the proposed algorithm uses NACK packet to check network connectivity, where NACK packet reaches to most of relay candidate nodes in this application.
- ***C2 - Dense network:*** Instead of exchanging connectivity information among all nodes, the proposed algorithm allows each node to compare its ROI with RQI contained in the currently relayed packets and decide whether to relay packets for the next sequence.

3.1 Frame Format



[Data packet]

msgType	seqNumber	srcAddr	relayAddr	relayRQI
data				

[NACK packet]

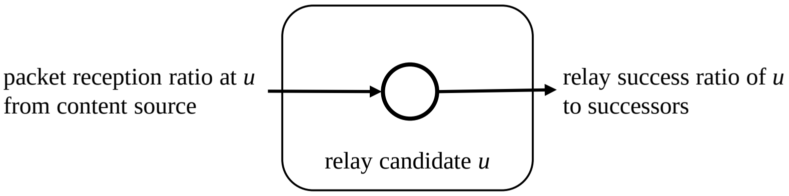
msgType	seqNumber

Fig. 2. Frame format of data and NACK packets

Fig. 2 describes the frame format of data and NACK packets used in the proposed algorithm. The *msgType* and *seqNumber* fields are commonly used in these packets. The *msgType* field is used to identify whether it is data packet or NACK packet, and the *seqNumber* field in the NACK is used to indicate the missed data packet. In the data packet, *srcAddr*, *relayAddr*, and *relayRQI* fields are additionally defined. The *srcAddr* denotes the originator address of broadcast packet, and the *relayAddr* and *relayRQI* fields play an important role to select a relay node. Namely, *relayAddr* and *relayRQI* fields are updated by relay nodes, whenever packets are relayed. Since each node can identify whether its RQI is higher than that of the current relay node by comparing the value in *relayRQI* field, it is possible to select the relay node having the highest RQI in a fully distributed manner.

Moreover, each node can easily identify whether it is relay candidate or successor in **Fig. 1** by comparing the *sourceAddr* and *relayAddr* fields in the received packet. The nodes receiving the packet with identical *sourceAddr* and *relayAddr* fields become relay candidates, and the other nodes with different *sourceAddr* and *relayAddr* fields are the successors. Here, note that relay candidates and successors can be changed whenever the content source broadcasts packets, since actual transmission coverage varies with wireless link condition.

3.2 RQI (Relay Quality Indicator)



$$\mathrm{RQI(u)} = \ (packet\ reception\ ratio\ at\ u) \cdot (relay\ success\ ratio\ of\ u)$$

Fig. 3. Definition of RQI (Relay Quality Indicator)

Fig. 3 shows the metric called RQI (Relay Quality Indicator), which is used by nodes to decide whether it will relay the packets or not. It indicates that the node with higher RQI has high possibility to afford to relay the packets from its content source to successors. As shown in **Fig. 3**, the quality of relaying is composed of packet reception ratio from the content source and relay success ratio to its successors. Note that two conditions should be satisfied at the same time to act as a relay node. For example, even though a node receives one hundred percent of packets from the content source, it cannot be a relay node if there are no neighboring successors to deliver the packets. As an opposite case, even though a node has good network links to the neighboring successors, it cannot act as a relay node if it has no received packets from the content source.

$$\begin{aligned}\mathrm{RQI(u)} &= \ (packet\ reception\ ratio\ at\ u) \cdot (relay\ success\ ratio\ of\ u) \\ &= \left(\frac{\sum_{x=i}^{j} rx(p_x)}{j-i}\right) \cdot \left(\frac{\sum_{x=i}^{j} relay(p_x) - \sum_{x=i}^{j} nack(p_x)}{\sum_{x=i}^{j} relay(p_x)}\right) \\ &, where \begin{cases} rx(p_x) = 1, when\ u\ recievs\ p_x; 0, otherwise \\ relay(p_x) = 1, when\ u\ relays\ p_x; 0, otherwise \\ nack(p_x) = 1, when\ u\ receives\ NACK\ for\ p_x; 0, otherwise \end{cases}\end{aligned} \tag{1}$$

Eq. (1) derives the detail of RQI in the viewpoint of relay candidate node *u*. The p_x is denoted as the broadcast packet with *seqNumber x*, and the value of *i* and *j* indicate the first and the last *seqNumber* of monitoring block of consecutive broadcast packets. If the relay node has enough memory, it can keep track of all broadcast packets by setting *i* to 0 and *j* to the last received *seqNumber*. Otherwise, relay node *u* can restrict the size of monitoring block of broadcast packets by adjusting the value of *i*. The other terminologies *rx(p_x)*, *relay(p_x)*, and *nack(p_x)* are defined to indicate whether *u* receives, relays, or receives NACK for the packet p_x respectively, as described in detail in Eq. (1).

As shown in Eq. (1), RQI of each node *u* is learned whenever it receives or relays broadcast packet. The former term in RQI is the packet reception ratio from the content source, and it can be simply measured by dividing the total number of received packets by the total number of transmitted packets from the content source. The relay success ratio of *u* in the latter term can be calculated by counting the number of relayed packets and the number of NACK packets received from successors. The NACK packet is sent when successors find out missed packets based on the *seqNumber* of received packets. This feedback informs the relay nodes that the previously relayed packets failed to deliver.

For example in **Fig. 1**, let assuming that node B relayed p_1 and p_2, and node E relayed p_x *(for x>3)*, since B has missed p_x. In case that node Z has received only p_x from E, it recognizes that the packets with *seqNumber smaller than x* have missed. Thus, it sends back NACK by broadcasting. On receiving NACK from Z, node B can get feedback that its relaying has failed to deliver to all of successors, and its RQI decreases again by recalculation. Since the relay node is selected with highest RQI among relay candidates, node E continues to relay the next broadcast packets unless it gets another NACK from the successors.

Note that NACK packet sent by broadcasting is reached to most of relay candidate nodes. But it cannot guarantee that NACK is delivered to the actual relay node, since it is possible that they are located outside their communication range. To avoid this case, the reliable feedback can be provided by applying broadcasting based recovery procedure for the missed packet. In other words, if successors request the missed packet by broadcasting and the other relay candidate responses with missed packet, the response packet can be reached at least 2-hop away from the requesting successors. Therefore, NACK packet's reachability problem can be enhanced by considering these request and response packets for missed packet as NACK.

3.3 RQI based Relay Node Selection Algorithm

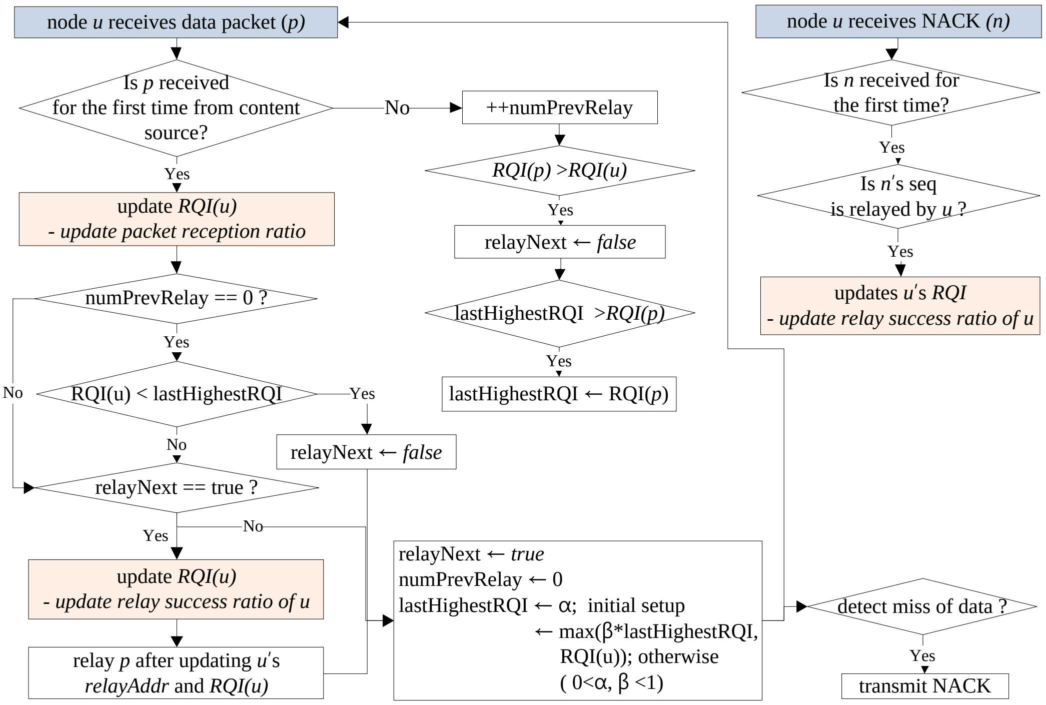

node u receives data packet (p)
Is p received for the first time from content source?
No
++numPrevRelay
RQI(p) >RQI(u)
Yes
relayNext ← false
lastHighestRQI >RQI(p)
Yes
lastHighestRQI ← RQI(p)
Yes
update RQI(u)
- update packet reception ratio
numPrevRelay == 0 ?
Yes
No
RQI(u) < lastHighestRQI
Yes
relayNext ← false
No
relayNext == true ?
No
Yes
update RQI(u)
- update relay success ratio of u
relay p after updating u's relayAddr and RQI(u)
relayNext ← true
numPrevRelay ← 0
lastHighestRQI ← α; initial setup
← max(β*lastHighestRQI, RQI(u)); otherwise
(0<α, β <1)
node u receives NACK (n)
Is n received for the first time?
Yes
Is n's seq is relayed by u ?
Yes
updates u's RQI
- update relay success ratio of u
detect miss of data ?
Yes
transmit NACK

Fig. 4. RQI based relay node selection algorithm

Note that each node updates its RQI whenever it transmits or receives the packets. For example, RQI increases as much as it receives data packets from the content source without collision and it successfully relays the packets to successors. On the contrary, RQI decreases when it has missed packets from the content source or receives a NACK packet indicating the failure of packet relay.

Fig. 4 describes the proposed relay node selection algorithm, in which each node decides whether to relay the received packets in a distributed and dynamic manner. Namely, there is no centralized coordinator to choose relay nodes, and relay nodes are dynamically adjusted according to the change of a network environment. The main idea of the proposed algorithm is that each node keeps updating its RQI, and decides to relay if it has the highest RQI among neighboring nodes. Thus, each node *u* monitors the RQI information relayed in data packets and compares its RQI with *RQI(p)* contained in the relayed data packet *p*. If node *u* has recognized that it has the highest RQI, it relays the received packet. Otherwise, it does not relay the packet but only updates the RQI information. 1.

Followings are the detailed algorithm.

1. Each node *u* keeps three variables such as *relayNext*, *numPrevRelay*, and *lastHighestRQI* in order to update its *RQI(u)* and decide to relay the received packet. The *relayNext* field indicates whether to relay the received packet, when a node *u* receives the broadcast packet for the first time. If this value is set as *true*, node *u* rebroadcasts the packet. The *numPrevRelay* counts the number of duplicate packets, and the *lastHighestRQI* keeps track of the highest RQI received at the previous broadcast packet. The initial settings for *relayNext*, *numPrevRelay*, and *lastHighestRQI* are *true*, *0*, and *α*, respectively, where $0< \alpha<1$.
2. When a node *u* receives the broadcast packet for the first time from the content source, node *u* updates its *RQI(u)* since the number of packets from the content source is increased.
   a. In case that *numPrevRelay* is higher than 0, there were at least one relayed packets at the previous broadcast packet. Note that, among previous neighboring relay nodes, only one node with the highest RQI has a chance to relay the next broadcast packet. (step 3) This node can be identified by *relayNext* field. If *relayNext* is set as *true*, a node *u* broacasts the received packet *p* and updates *RQI(u)* to reflect relay success ratio of *u*. Otherwise, *u* does not relay the received packet *p*.

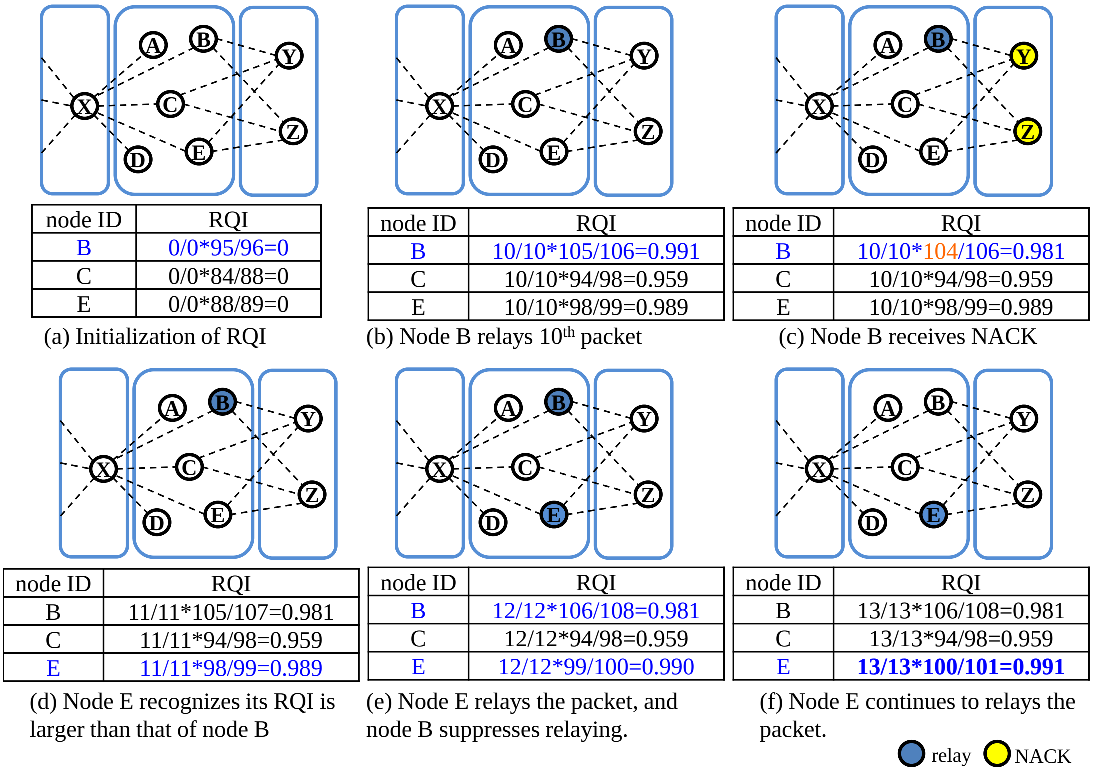
   b. In case that *numPrevRelay* is 0, there were no relay nodes at the previous broadcast packet. It results in that all nodes' *relayNext* field was set to *true*, as described in the next step 2.c. If all nodes participate in relaying the packet, the collision from duplicate relays occurs and it may fail to select the best relay node. Thus, the solution is to allow only selected nodes with higher *RQI* than *lastHighestRQI* to maintain *relayNext* field as *true*.

   However, in the initial stage, it is still possible that several nodes with identical and higher RQI than the initial value of *lastHighestRQI (= α)* relay the received packet. To prevent this problem, the relay success ratio part in RQI is initialized with random values, as shown in **Fig. 5 (a)**. By selecting appropriate value of *α* and relay success ratio part, it is possible to select small number of relay nodes even in the initial stage. The example and detail discussion are given in the next subsection 3.4.

   c. When the broadcast packet handling process finished, node *u* resets *relayNext*, *numPrevRelay, and lastHighestRQI* fields. *RelayNext* is initialized as *true*, and *numPrevRelay* is set with 0. The *lastHighestRQI* is reduced as much as *1-β*, where *β* is defined as reduction ratio considering that every node's RQI is reduced due to packet reception failure, packet relay failure, and any other reasons.

3. When a node *u* receives duplicate broadcast packets, each node *u* keeps monitoring *RQI(p)* relayed by other nodes and counts the number of relay packets using *numPrevRelay* field. If node *u* finds *RQI(p)* higher than its *RQI(u)*, it disables *relayNext* field with *false*. As well as *relayNext* field, it also investigates whether *RQI(p)* is higher than *lastHighestRQI* value, and updates it with *RQI(p)*. Note that this information plays an important role in that they decide whether to relay the received packet in a distributed manner and support dynamic adjustment for network topology changes.
4. When a node *u* receives the NACK packet *n* for the first time, *u* investigate *seqNumber* field at NACK packet *n*. If the data packet with the same *seqNumber* is relayed by *u*, node *u* updates its *RQI(u)* since NACK indicates that the relayed packet by *u* failed to reach to all of successors. This update *RQI(u)* is notified at the next broadcast packet relay and it makes other nodes with higher *RQI* take over a relaying role.

3.4 RQI Initialization and Example of Relay Selection



node ID	RQI
B	0/0*95/96=0
C	0/0*84/88=0
E	0/0*88/89=0

(a) Initialization of RQI

node ID	RQI
B	10/10*105/106=0.991
C	10/10*94/98=0.959
E	10/10*98/99=0.989

(b) Node B relays 10th packet

node ID	RQI
B	10/10*104/106=0.981
C	10/10*94/98=0.959
E	10/10*98/99=0.989

(c) Node B receives NACK

node ID	RQI
B	11/11*105/107=0.981
C	11/11*94/98=0.959
E	11/11*98/99=0.989

(d) Node E recognizes its RQI is larger than that of node B

node ID	RQI
B	12/12*106/108=0.981
C	12/12*94/98=0.959
E	12/12*99/100=0.990

(e) Node E relays the packet, and node B suppresses relaying.

node ID	RQI
B	13/13*106/108=0.981
C	13/13*94/98=0.959
E	**13/13*100/101=0.991**

(f) Node E continues to relays the packet.

Fig. 5. Example of learning based relay node selection

Fig. 5 shows an example of the proposed relay node selection algorithm. In **Fig. 5**, a relay node is selected based on RQI that is learned from the packet reception ratio from the content source and relay success ratio to successor nodes. However, it is clear that nodes need a certain number of broadcast packets to learn RQI, and several relay candidates participate in relaying the packets at the same time in the initial stage. But, it may cause that the collision from duplicate relays interrupts normal procedure to select best relay node.

$$\begin{aligned}\mathrm{RQI(u)} &= (packet\ reception\ ratio\ at\ u)\cdot(relay\ success\ ratio\ of\ u)\\ &= \left(\frac{\sum_{x=i}^{j} rx(p_x)}{j-i}\right)\cdot\left(\frac{\left(\upsilon+\sum_{x=i}^{j} relay(p_x)\right)-\left(\omega+\sum_{x=i}^{j} nack(p_x)\right)}{\upsilon+\sum_{x=i}^{j} relay(p_x)}\right)\\ &,where\ \upsilon\ and\ \omega\ are\ uniform\ random\ value\ and\ \upsilon\geq\omega.\end{aligned} \tag{2}$$

As briefly mentioned at step 2.b in subsection 3.3, the solution to prevent this problem is to apply random values at the relay success ratio part in the RQI as shown in **Fig. 5 (a)**. The relay success ratio part in Eq. (1) was originally composed of the number of relayed packets and the number of NACK packets received, and it can be modified to Eq. (2) by adding uniform random variables *v* and *w*. By applying Eq. (2) instead of Eq. (1), the RQI of the relay candidates can be differentiated even in the initial stage of broadcasting as shown in **Fig. 5**, and the aforementioned problem can be prevented.

It is also noticeable that initial value *α* of *lastHighestRQI*, discussed in algorithm step 2.b, can be chosen to allow at least one relay node in the initial stage by considering the number of nodes in a network and the distribution of two random variables *v* and *w*.

Fig. 5 (b)-(f) shows that how the relay node is dynamically selected according to the feedback from successors. In **Fig. 5 (b)**, node B is selected as a relay node and relays broadcast packets 10 times. If successors Y and Z have missed one packet from B, they transmit NACK packets. On receiving NACK packets, RQI of node B is decreased and it becomes smaller than that of node E, as shown in **Fig. 5 (c)**. Since node E is recognized that it RQI is higher than current relay node by overhearing relay packet from B as shown in **Fig. 5 (d)**, it starts relaying from the next broadcast packet. It allows both B and E to rebroadcast at the next broadcast packet as shown in **Fig. 5 (e)**, but node B suppresses its relaying as shown in **Fig. 5 (f)** because it comes to know that there is the node with higher RQI than B. Like this manner, a relay node is dynamically adjusted by learning RQI and selecting the node with the highest RQI.

4. Network Simulation Results

Table 1. Simulation Parameters

Simulation Parameters	Value
Network size	30m x 30m
Number of nodes	51
Deployment type	Random
Position of source node	Corner
PHY/MAC protocol	IEEE 802.11 a/g
Maximum data rate	54Mbps
Propagation model	Two-ray
Transmission range	25m
Content size	1500bytes x 1000 segments
Background hello interval	No/1000/500msec

In this section, the learning based relay selection algorithm is evaluated through network simulation by dividing two subsections. At the first subsection, it is analyzed whether a relay node can be dynamically adjusted according to network dynamics such as noise traffic and topology changes. To enable noise traffic condition, each node transmits hello messages for a given interval. Thus, it can be said that a network has higher noise traffic for the smaller hello interval. At the second subsection, the performance of the learning based relay selection algorithm is compared with the random based relay selection algorithm and topology based minimum relay selection algorithm in terms of network throughput, packet delivery ratio, and the number of transmitted packet overhead.

In this evaluation, the network simulator NS-2 and IEEE 802.11 a/g PHY/MAC protocols are used. To emulate a smart class environment, 51 nodes are deployed in the terrain size 30m x 30m, and a source node is located at the corner. Other general parameter settings are summarized in **Table 1**. [23]

4.1 Analysis of Dynamic Adjustment Behavior

This subsection evaluates how well the proposed algorithm changes relay nodes against background noise traffic environment. As already mentioned, RQI information of each node is always updated whenever it has successfully received packets from the content source, it relays the packets, and it receives NACK from successors. In case that it successfully receives the packets from the content source or plays as a relay node, node *u*'s RQI increases. On the contrary, if the relay node fails relaying the packets, node *u*'s RQI decreases. Since the proposed algorithm allows only the node with the highest RQI to participate in relaying the packets, the node, which continuously successes to rebroadcast the packet, consistently plays as a relay node. However, when it fails to relay the packet, its RQI is decreased. Therefore, it could be replaced with other node with higher RQI.

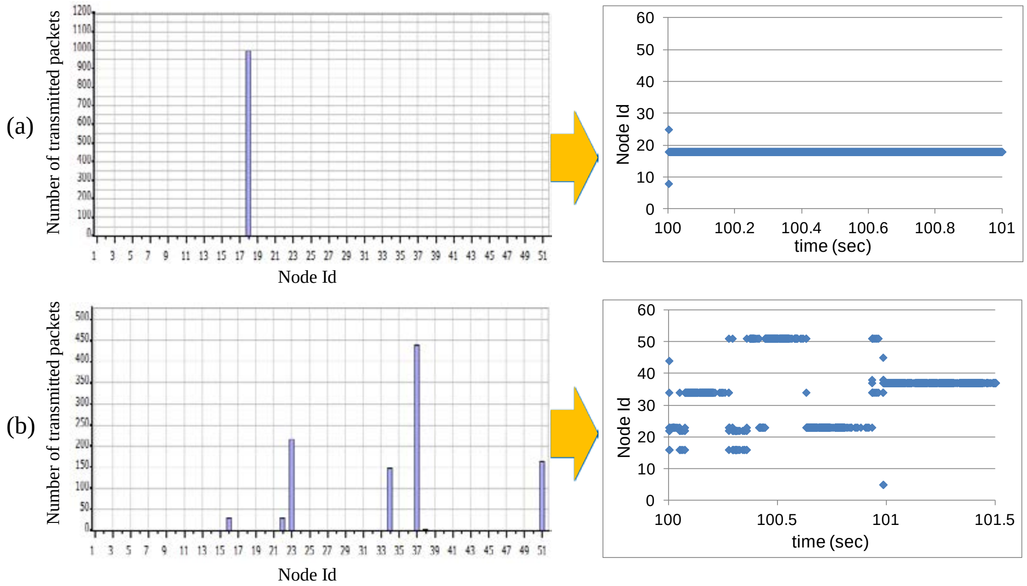


Fig. 6. Analysis on the number of forwarded packets of each node in learning based relay selection algorithm under (a) no background traffic and (b) background traffic

Fig. 6 describes the behavior of the learning based relay node selection algorithm according to the background noise traffic. Note that the left figures show the number of forwarded packets per each node, and the right figures shows how relay nodes change as time passes. When there is no noise traffic like **Fig. 6 (a)**, it only takes 0.1sec to select reliable relay node, and the selected relay node continuously relays the received packets. On the contrary, **Fig. 6 (b)** shows that about six nodes participate in relaying the received broadcast packets, when there is background noise traffic. It is because the noise traffic is occurred randomly both in time and location, and the selected relay node may fail to deliver a packet when noise occurs around it. In this case, another node that has higher RQI takes over relaying role and provides packet delivery to successors. Thus, it can be concluded that the proposed learning based relay node selection algorithm is designed to adapt to the dynamic network environment such as noise traffic, node removal or movement.

4.2 Performance Comparison

This subsection compares the learning based relay selection algorithm with the random relay selection and the topology based minimum relay selection algorithms by varying the noise traffic interval. In the random relay selection algorithm, each node randomly chooses whether to relay the received packet for the given probability, which is set as 4 percent (= 2 relay nodes/50 receives) in this simulation. The minimum relay selection algorithm follows [14], which is a topology based centralized algorithm and proves that its latency is O(1) times of the optimal solution. The evaluation metrics include the number of transmitted packets, packet delivery ratio, and the network throughput, but the control packet overhead and time to select relay nodes in the minimum relay algorithm are not included in order to focus on broadcasting performance.

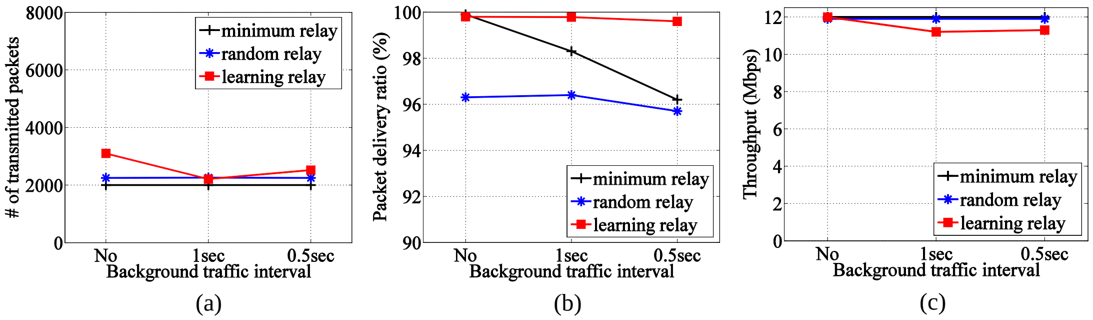


Fig. 7. Comparison of broadcasting performance

As shown in **Fig. 7 (a)**, the minimum relay selection algorithm minimizes the number of relay nodes, and the number of transmitted packets in the minimum relay selection tends to be constant with the increasing noise traffic. The proposed learning based algorithm requires slightly higher number of packets than others, since it allows successors to transmit NACK packet as a feedback to the current relay nodes. However, it is interesting to note that the learning based relay selection algorithm provides near to one hundred percent delivery ratio regardless of the background noise traffic in **Fig. 7 (b)**, while the packet delivery ratio of others gradually decrease for the increasing noise traffic. It is because the relay node in the learning based algorithm is dynamically adjusted with NACK feedback according to the link status changes as shown in **Fig. 6**. On the contrary, both of the random relay and the minimum relay algorithms do not equipped with any mechanism to detect link status or topology changes and adjust relay nodes. Especially, the packet delivery ratio in the minimum relay algorithm is significantly affected by the noise traffic, since it keeps using the selected relay nodes differently from the random relay algorithm that changes the relay node for every broadcast packet. The network throughput in **Fig. 7 (c)** proves the efficiency of broadcast selection algorithm. For example, the random relay and minimum relay selection algorithm achieve the maximum 12Mbps throughput, since they do not require any control packet overhead during the relay selection. The learning based relay selection algorithm consumes up to 1Mbps bandwidth for NACK feedback and adjustment of relay nodes, and this can be considered as the cost to provide reliable broadcasting against dynamic topology environment.

5. Testbed Implementation Results

Fig. 8 shows the experimental testbed, which is composed of one content source and eight receiver devices. Each tablet with 1.4GHz quad core is equipped with three kinds of broadcasting algorithm such as proposed algorithm, simple flooding, and the random based broadcast algorithm. Similar with network simulation, the measured metrics are relay ratio and packet delivery ratio, where the relay ratio is defined as the ratio of the number of relayed packets to 1 originated packet. For example, relay ratio 1.0 indicates that the exactly 1.0 number of packets is relayed for 1 broadcast packet.

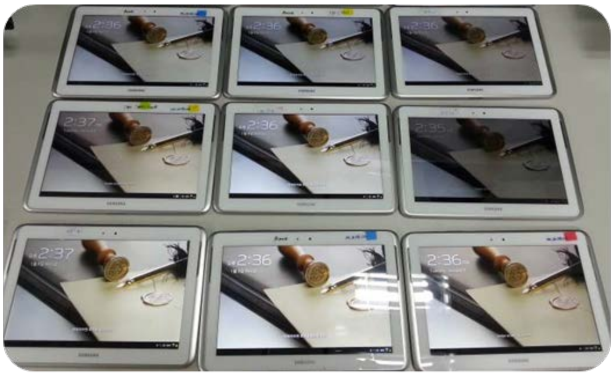


Fig. 8. Evaluation testbed

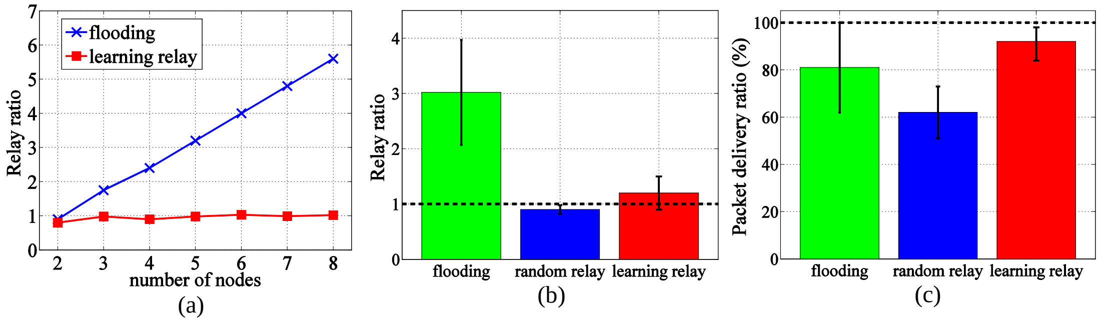


Fig. 9. Performance comparison with flooding and random relay

The main goal of this measurement is to prove that the proposed relay selection algorithm successfully converges to one best relay node regardless of increasing number of nodes in a network. In other words, if the proposed learning based relay selection algorithm does not work well in the real environment, the relay ratio may increase as many as the number of nodes in a network, like simple flooding algorithm in **Fig. 9 (a)**. It is also interesting to note that the relay ratio of the learning relay selection algorithm remained relatively constant with the number of nodes in the network, while the simple flooding algorithm proportionally increases for the number of nodes.

As shown in **Fig. 9 (b)** and **(c)**, the learning relay selection algorithm supports more than 95 percent reliable packet delivery ratio with relay ratio 1.2, whereas the simple flooding and random relay algorithm achieve 81 and 62 percent of packet delivery ratio with 3.0 and 0.9 relay ratio, respectively. Therefore, it can be concluded that the proposed relay selection algorithm successes to choose a best relay node even in a real environment regardless of network density.

6. Conclusion

This paper proposes the learning based relay node selection algorithm for efficient and reliable broadcasting in smart class application. Different from existing algorithms that require separate phase for selecting relay nodes, the proposed algorithm makes each node learn relay quality in a run-time and distributed manner. It has no startup delay to run the proposed algorithm and the selected relay nodes can be easily adjusted against network dynamic changes. The network simulation results have shown the superior performance in terms of packet delivery ratio, bandwidth usage, and adaptability comparing with existing algorithms, and the real testbed experiment also proves the feasibility of the proposed algorithm. Therefore, it is expected that the proposed learning based relay selection algorithm to be utilized for efficient and reliable content delivery in diverse wireless application including smart class.

References

[1] D. Kang, K. Kang, H. Lee, J. Jung, C. Bae and J. Lee, "A Smart Class System based on SoD (System on-Demand) Technology," in *Proc. of IEEE International Symposium on Consumer Electronics*, 2012. Article (CrossRef Link).

[2] S. Lim and J. Lee, "An Immersive Augmented-Reality-Based e-Learning System Based on Dynamic Threshold Marker Method," *ETRI Journal*, vol. 35, no. 6, pp. 1048-1057, Dec, 2013. Article (CrossRef Link).

[3] M. Osamnia, A. J. Berena, S. Chunwijtra, H. Okada and H. Ueno, "An Automated Authoring System by Means of Integrating e-Meeting and e-Learning to Support Higher Education Under the WebELS Platform," in *Proc. of IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, 2014. Article (CrossRef Link).

[4] J. Kuo, C. Shih, C. Ho and Y. Chen, "A Cross-layer Approach for Real-time Multimedia Streaming on Wireless Peer-to-peer Ad Hoc Network," *Ad Hoc Networks*, vol. 11, issue 1, pp. 339-354, Jan, 2013. Article (CrossRef Link).

[5] S. Ni, Y. Tseng, Y. Chen and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Netwok," in *Proc. of ACM Mobicom*, 1999. Article (CrossRef Link).

[6] D. G. Reina, S. L. Toral, P. Johnson and F. Barrero, "A survey on probabilistic broadcast schemes for wireless ad hoc networks," *Ad Hoc Networks*, vol. 25, pp. 263-292, Feb, 2015. Article (CrossRef Link).

[7] A. Qayyum, L. Viennot and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks," in *Proc. of HICSS*, 2002. Article (CrossRef Link).

[8] J. Ahn and T. Lee, "Multipoint relay selection for robust broadcast in ad hoc networks," *Ad Hoc Networks*, vol. 17, pp. 82-97, June, 2014. Article (CrossRef Link).

[9] T. Zhu, Z. Zhong, T. He and Z. Zhang, "Exploring Link Correlation for Efficient Flooding in Wireless Sensor Networks," in *Proc. of NSDI*, 2010. Article (CrossRef Link).

[10] S. Guo, S. Kim, T. Zhu, Y. Gu and T. He, “Correlated Floooding in low-Duty-Cycle Wireless Sensor Networks,” in *Proc. of IEEE ICNP*, 2011. Article (CrossRef Link).

[11] B. Lichtblau and J. Redlich, “Link Quality based Forwarder Selection Strategies for Reliable Network-Wide Broadcasts,” in *Proc. of WoWMoM*, 2013. Article (CrossRef Link).

[12] J. Subramanian, R. Morris and H. Balakrishnan, “UFlood: High-Throughput Flooding over Wireless Mesh Networks,” in *Proc. of IEEE INFOCOM*, 2012. Article (CrossRef Link).

[13] J. Cha and G. Baek, “Novel Selection-Based Joint Network Coding and Scheduling Scheme in WMNs: JNCS,” *ETRI Journal*, vol. 37, no. 2, April, 2015. Article (CrossRef Link).

[14] R. Gandhi, A. Mishra and S. Parthasarathy, “Minimizing broadcast latency and redundancy in ad hoc networks,” *IEEE/ACM Transaction on Networking (TON)*, vol. 16, pp. 80-851, Aug, 2008. Article (CrossRef Link).

[15] W. Wang and B.H. Soong, “Collision-free and low-latency scheduling algorithm for broadcast operation in wireless ad-hoc networks,” *IEEE Communications Letters*, vol. 11, no. 10, pp. 793-795, Oct, 2007. Article (CrossRef Link).

[16] H. Li, K. Bok, K. Chung and J. Yoo, “An Efficient Data Dissemination Method over Wireless Ad-Hoc Networks,” *Wireless Personal Communications*, vol. 79, issue 4, pp. 2531-2550, Dec, 2014. Article (CrossRef Link).

[17] B. Liu, V. Firoiu, J. Kurose, M. Leung and S. Nanda, “Capacity of Cache Enabled Content Distribution Wireless Ad Hoc Network,” in *Proc. of IEEE MASS*, 2014. Article (CrossRef Link).

[18] J. Kim, G. Lee and J. Choi, “Popularity-Based Adaptive Content Delivery Scheme with In-Network Caching,” *ETRI Journal*, vol. 36, no. 5, Oct, 2014. Article (CrossRef Link).

[19] R. Draves, J. Padhye and B. Zill, “Comparison of routing metrics for static multi-hop wireless networks,” in *Proc. of ACM SIGCOMM*, 2004. Article (CrossRef Link).

[20] Z. Shi, C. Beard and K. Mitchell, “Analytical models for understanding misbehavior and MAC friendliness in CSMA networks,” *Performance Evaluation*, vol. 66, issues 9-10, pp. 469-487, Sep, 2009. Article (CrossRef Link).

[21] Z. Shi, C. Beard and K. Mitchell, **“**Analytical Models for Understanding Space, Backoff and Flow Correlation in CSMA Wireless Networks,” *Wireless Networks*, vol. 19, issue 3, pp. 393-409, April, 2013. Article (CrossRef Link).

[22] Z. Shi, C. Beard and K. Mitchell, “Competition, Cooperation, and Optimization in Multi-Hop CSMA Networks with Correlated Traffic,” *International Journal of Next-Generation Computing*, vol. 3, no.3, pp. 228-246, Nov, 2012. Article (CrossRef Link).

[23] T. Kim, P. K. Chong and D. Kim, "One-hop neighbor based broadcast scheduling in wireless sensor networks," *IEICE Transactions on Communications*, vol. E94-B, no. 1, pp. 297-301, January, 2011. Article (CrossRef Link).

Taehong Kim received the BS degree in computer science from Ajou University, Korea, in 2005, the MS degree in information and communication engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2007, and the PhD degree in computer science from KAIST, Korea, in 2012. He has been a research staff member at SAIT (Samsung Advanced Instituted of Technology) and Samsung DMC R&D Center, from May 2012 to March 2014. He currently works for ETRI (Electronics and Telecommunications Research Institute) since April 2014. His research interests include wireless sensor network, content centric network, and software defined network.