논문 12-37B-12-13

한국통신학회논문지 '12-12 Vol.37B No.12
http://dx.doi.org/10.7840/kics.2012.37B.12.1208

CSMA 무선 네트워크에서 안정성 있는 신뢰적 브로드캐스트를 위한 2-폴링 피드백 방법

윤 원 용◆

2-Polling Feedback Scheme for Stable Reliable Broadcast in CSMA Wireless Networks

Wonyong Yoon◆

요 약

IEEE 802.11와 같은 CSMA 무선 네트워크에서 브로드캐스트 전송을 안정성 있게 신뢰적으로 수행하기 위해, 브로드캐스트 송신 노드는 복수의 수신자들에게 충돌 없는 전송을 할 수 있도록 해야 하며 복수 수신자들의 패킷 수신 상태를 유지해야 한다. 본 논문은 무선 네트워크에서 안정적이고 신뢰적인 브로드캐스트를 수행하기 위한 효율적인 피드백 방법인 2-폴링 피드백을 제안한다. 이 방법에서 송신 노드는 두 개의 수신자 노드의 채널 상태를 확인한 후 브로드캐스트 전송을 수행한다. 본 논문은 브로드캐스트 전송을 위한 피드백 방법의 클래스를 분류하고 이들에 대해 성능 분석을 수행한다. 제안한 2-폴링 피드백 방법이 기존 피드백 방법보다 패킷 전송 지연시간과 패킷 안정화 시간 측면에서 우수함을 보인다. 2-폴링 방법은 전체-폴링 방법보다 복잡도가 낮으며, 1-폴링 방법과 비교할 때는 동일한 복잡도에 패킷 안정화 시간이 반 정도 감소함을 알 수 있다.

Key Words : CSMA network, stable reliable broadcast, acknowledgment feedback, polling, numerical analysis

ABSTRACT

Disseminating broadcast information stably and reliably in IEEE 802.11-like CSMA wireless networks requires that a source should seek collision-free transmission to multiple receivers and keep track of the reception state of the multiple receivers. We propose a simple yet efficient feedback scheme for stable reliable broadcast in wireless networks, called 2-polling feedback, where the state of two receivers are checked by a source before its broadcast transmission attempt We present a performance analysis of the class of reliable broadcast feedback schemes in terms of two performance metrics (packet transmission delay and packet stable time). The analysis results show that the proposed 2-polling feedback scheme outperforms the current existing classes of feedback schemes in the literature, i.e., all-polling feedback and 1-polling feedback. The 2-polling feedback scheme has lower asymptotic complexity than the all-polling feedback, and has the same asymptotic complexity as the 1-polling feedback but exhibits almost 50 % reduction in packet stable time.

Ⅰ. Introduction

As wireless network applications increasingly involve multiple receivers for the same

※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2012R1A1A2008478).
◆ 주저자 : 동아대학교 전자공학과, wyyoon@dau.ac.kr, 정회원
논문번호 : KICS2012-07-310, 접수일자 : 2012년 7월 6일, 최종논문접수일자 : 2012년 12월 7일

information, e.g., mobile IPTV, adding reliability and stability to broadcast transmission to these multiple receivers in wireless networks becomes as important as conventional reliable unicast transmission. Efficient feedback schemes are needed for multiple receivers to reserve their neighbor area, e.g., via Clear to Send (CTS) packets in IEEE 802.11-like carrier-sense multiple access (CSMA) protocols, in order to receive broadcast packets without collision with other transmission in that neighbor area. In addition, the sender needs to gather acknowledgments (ACKs) from these multiple receivers without collision with each other. Despite this requirement, the current standard wireless medium access control (MAC) protocols, e.g., IEEE 802.11 [1] and Bluetooth [2] have not yet adopted efficient feedback mechanisms for multicast/broadcast transmission. IEEE 802.11 broadcast mechanism allows for a sender to simply sense the carrier, avoid collision, and transmit a broadcast packet without reserving receivers' neighbor areas and collecting any ACKs from receivers[1]. Bluetooth 1.1 also does not require receivers to respond with ACKs except that a sender can transmit a broadcast packet multiple times for increased reliability. For the first time as a standard protocol, Bluetooth 1.2 adds reliability by simply collecting ACKs from all the receivers.

A packet is considered stable when a packet source knows that all intended receivers receive the packet. The source can notice this by receiving from the receivers either ACK to an individual packet or the reception state information of the receivers. The source can purge its buffer space for the packets acknowledged because they are stable and hence can be discarded from the buffer. The source will keep packets which are not yet acknowledged in the buffer. Some protocols lack such feedback information and thereby cannot guarantee reliability with finite buffer spaces[3,5-8].

Explicit ACK is necessary for packet stability. Smaller packet stable time is desirable because with the less stable time the less buffer space should be occupied. Depending on how many receivers send back ACK to the source, the existing feedback methods for stable reliable broadcast can be classified into two categories. In the all-polling feedback class, all intended receivers send ACK or NACK for each data packet transmission, e.g., Kuri et al.'s work[6]. In the 1-polling feedback class, only one receiver sends ACK for each data broadcast, e.g., BMW protocol[10]. In this article, we propose a new simple yet efficient feedback class where two receivers' ACK status is reported back to the source in one feedback opportunity. The three feedback classes differ in how to reserve receivers' area for collision-free broadcast transmission. More specifically, the all-polling feedback scheme requires all receivers to be ready to receive broadcast packets before the source can transmit, that is, the source waits for CTS from all receivers before transmission. However, the 1-polling feedback scheme requires only one CTS from one receiver [10] while the 2-polling feedback scheme checks the wireless medium status of two receivers.

1.1. Contribution

We highlight the contributions of this paper. First, we propose a new class of stable reliable broadcast feedback schemes in single-hop wireless networks, called 2-polling feedback, that polls explicit ACK status of two receivers in an efficient way to reduce packet stable time. Second, we perform to the best of our knowledge the first formal numerical analysis of stable reliable broadcast protocols in single-hop wireless networks and provide asymptotic complexity and numerical comparison. Most work has used simulation for performance evaluation. Third, we focus especially on stability in addition to reliability that most previous works have dealt with.

1.2. Related Work

Research has been done in two categories: one using in-band control packets [6][9][10] and another using an extra busy tone radio[3,5,8]. Kuri et al. addresses the collision problem of CTS packets and ACK packets by expecting different responses (CTS and ACK from a leader node, and NCTS and NAK from others) in IEEE 802.11 infrastructure mode single-hop wireless networks[6]. A source may figure out when the transmitted packet is stable by extending the protocol to select a new leader at each attempt. Batch Mode Multicast MAC requires a source to collect n CTS packets before transmission and n ACK packets after transmission where n is the number of receivers[9]. Although the protocol itself is stable, it causes too much overhead for packet stability. BMW protocol [10] exploits the broadcast nature of the wireless medium. At one time, the source unicasts a packet to one receiver while others may overhear it.

Lou et al. propose ACK transmission only by some selected receivers called forwarding nodes [7]. Forwarding nodes are selected from the source node's neighbors such that the union of each forwarding node's range covers the source node's two-hop range. Only forwarding nodes send ACKs to the source upon successful receipt of broadcast packets. Non-forwarding nodes do not send ACKs to the source to reduce the number of ACKs at the source. They have two chances of packet receiving, one from the source and the other from other forwarding node. However, the protocol is not stable because the source does not know the ACK status of non-forwarding nodes. Zhu et al. [11] exploits the correlation of reception states among receivers to avoid getting ACKs from all individual receivers. With such correlation in consideration, a source can infer the reception state of a node from ACKs which the other node sent. Our work is different from [7][11] in that it focuses on single-hop MAC-layer broadcast transmission.

Gupta et al. [5] use an additional radio for sending busy tones as NCTS tones and NACK tones, to avoid the problem of CTS and ACK packet collisions. RMAC [8] uses two additional busy-tone radios for the purpose of avoiding hidden-node problems and acknowledging packet reception. Chaporkar et al. [3] also use a busy-tone radio for determining transmission point only, but not for MAC-layer reliability. Since the source does not receive any explicit ACK packets, the busy-tone protocols are not stable. Chaporkar et al. [4] propose the transmission policy that can make an optimal tradeoff between delay and loss under the assumption that receiver readiness is identical and identically distributed (i.i.d.). A sender does not collect ACKs from receivers to consider them in the next retransmission, and hence the proposed policy is not stable.

In cellular networks (e.g., 3GPP LTE, IEEE 802.16m), OOK (On-Off Keying)-based common ACK/NACK feedback channel has been considered. Jung et al. [12] propose optimization of feedback parameters that can reduce feedback overhead for multicast HARQ (Hybrid Automatic Repeat reQuest) transmission. Successful receivers do not send any signal whereas failed receivers send a pre-defined NACK feedback which will be detected by a multicast source. For ad hoc networks, Oliveira et al. [13] propose using a single-stage backoff and determining the optimal medium access probability for broadcast transmission by measuring the channel occupancy.

Ⅱ. 2-Polling Feedback for Stable Reliable Broadcast

We first give the generic description of two existing MAC-layer feedback schemes for stable reliable wireless broadcast called all-polling and 1-polling feedbacks. We propose a new enhanced 2-polling feedback scheme. We assume that a source node maintains the list of broadcast receivers in some ways, e.g., the list of associated nodes (through IEEE 802.11 association procedure), or the list of neighbors it discovers. Since we focus on single-hop MAC layer broadcast, MAC addresses of such neighbors are maintained at MAC layer.

2.1. All-polling Feedback

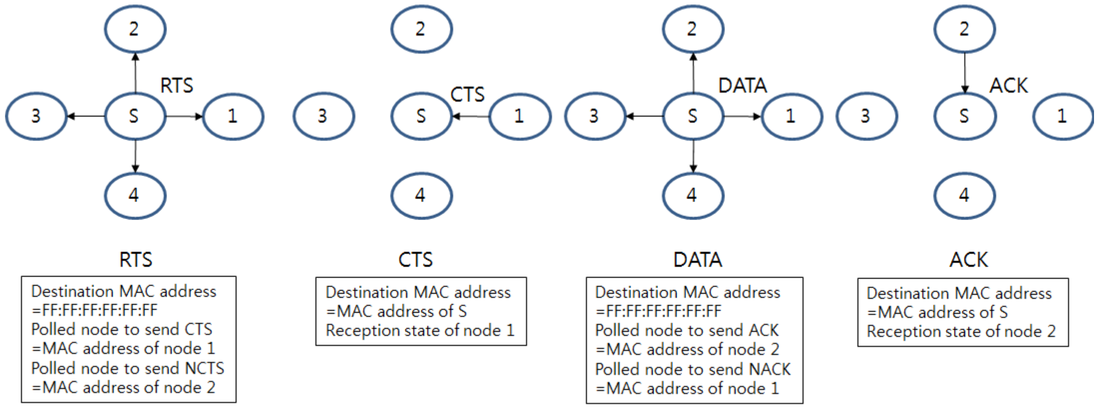

Fig. 1. One phase of RTS-CTS-DATA-ACK and information contained in each packet

A generic all-polling feedback scheme is defined as follows. A source sends RTS to reserve the wireless medium. It polls CTS from one selected receiver and NCTS from all other receivers. Upon receipt of CTS, which means all receivers are ready to receive, the source broadcasts packets. The selected receiver sends ACK if it successfully receives the broadcast packet. All other receivers send NACK if they do not receive the packet in the same slot of ACK. Upon receipt of ACK, which means the selected node successfully received a packet and all the other receivers did not send NACK, the source selects and polls the next receiver to send CTS and repeats the above procedure. An example of this class of protocols is Kuri et al.'s work[6].

2.2. 1-polling Feedback

A generic 1-polling feedback scheme is defined as follows. A source sends RTS. It polls CTS from one selected receiver. Upon receipt of CTS, the source broadcasts a data packet. The selected receiver sends ACK if it successfully receives the broadcast packet. Upon receipt of ACK, the source selects the next receiver to send CTS and repeats the above procedure. An example of this class of protocols is BMW protocol[10].

2.3. 2-polling Feedback

Based on the assumption that the source maintains a list of receivers, it knows n, the number of receivers. A source starts trying broadcast transmission from the first receiver i (=1) in the list. The source first senses the carrier and if the channel is idle, broadcasts RTS packet with a node to send CTS set to i and a node to send NCTS set to (i+1)%n. Then the following cases are possible.

- If the source receives CTS which implies that receiver i is definitely ready, but receiver (i+1)%n may be ready or not, then the source transmits a broadcast packet.
- If the source receives NCTS, receiver i may not be ready and receiver (i+1)%n is not ready, The source retries the same packet to i and (i+1)%n.
- If the source receives nothing, then i is not ready to receive, but (i+1)%n may be ready or not.
- If the source detects a collision of feedbacks, then it infers that i is ready but (i+1)%n is not ready.

In the last three cases, the source defers its transmission and retries the next RTS-CTS round before broadcast data transmission.

In the transmitted broadcast data packet, it is indicated that receiver (i+1)%n and i shall respond with ACK and NACK, respectively. Then the following cases are possible.

- If the source receives ACK, the two receivers both received the packet. So, the source repeats the same procedure at (i+2)%n to transmit the next packet.
- If the source receives NACK, then neither i nor (i+1)%n received the packet. The source

retries the same packet to i and (i+1)%n.

- If the source receives nothing, then (i+1)%n did not receive the packet, but receiver i may have received it. The source retries the same packet to (i+1)%n and (i+2)%n.
- If the source detects a collision, then it infers that (i+1)%n received the packet, but i did not. The source starts a new phase to send the same packet to i and (i+2)%n.

For example, Fig. 1 shows one phase of broadcast transmission which consists of RTS-CTS-DATA-ACK exchange. In each CTS, NCTS, ACK, and NACK, the reception state of a receiver is carried so that the source can determine up to which packet the receiver has successfully received and which packets are missing. If a missing packet is found at either of two receivers during polling, the source tries to retransmit the missing packet by polling the two receivers again. Packet loss that occurs at non-polled receivers can be recovered when the turn comes to poll them later and the source knows their reception status.

Since polling is performed on a group of receivers in a circular way, a packet that failed to arrive at a particular receiver in one cycle has a next chance to be retransmitted in later cycles. In this way, packet stability is guaranteed.

Ⅲ. Numerical Analysis

Unlike most previous works that use simulation, we provide a formal numerical analysis from the perspective of packet stability. A tradeoff between queueing delay at a multicast source and system throughput is pointed out in [3]. A protocol in which a source does not transmit until all the receivers are ready, e.g., [6], may lead to large delay at the source, while a protocol in which the source can transmit when only one receiver is ready, e.g., [10], may result in low system throughput. We are particularly interested in how the three feedback schemes differ in terms of packet transmission delay, T^w, and packet stable time, S^w for protocol w. Notations are given in Table 1.

Let the all-polling feedback [6], the 1-polling feedback [10], and the 2-polling feedback denoted by Pb, Pu, and Pm, respectively. Time is measured in logical unit time. For the simplicity of our analysis, we make the following assumptions.

- The source first checks if it can send an RTS. If so, it sends the RTS and waits for a CTS

Table 1. Notation

n	Number of broadcast receivers.
c	Packet loss probability that a receiver will experience collision if the source broadcasts a packet, i.e., the receiver is not ready to receive.
T^w	Packet transmission delay. Time until all the broadcast receivers receive a given packet for protocol $w \in \{Pb, Pu, Pm\}$.
X^w	Number of contention rounds until the source waits for the polled receivers are ready to receive for protocol $w \in \{Pb, Pu, Pm\}$.
M^w	Number of transmission attempts by the broadcast source until all the broadcast receivers receive a given packet for protocol $w \in \{Pb, Pu, Pm\}$.
X^w_i	Number of contention rounds until i-th transmission attempt is made for protocol $w \in \{Pb, Pu, Pm\}$.
S^w	Packet stable time. Time until a packet is stable such that the source can release the packet from buffer for protocol $w \in \{Pb, Pu, Pm\}$.
N^w_c	Bytes of control packets (RTS and CTS packets) until a data packet is stable such that the source can release the packet from buffer for protocol $w \in \{Pb, Pu, Pm\}$.
N^w_d	Bytes of data packets (data and ACK packets) until a data packet is stable such that the source can release the packet from buffer for protocol $w \in \{Pb, Pu, Pm\}$.

or NCTS. If it receives a CTS it sends a data packet. Otherwise, it defers the data packet transmission to the next RTS-CTS exchange.

- One RTS-CTS exchange takes T_c while DATA-ACK exchange takes T_d.
- Once the transmission is reserved by the RTS-CTS exchange, a data packet is delivered to reserved receivers without loss. A subsequent ACK is also delivered to the sender without loss. At the physical layer, the unstable nature of the wireless medium may cause packet loss even after RTS-CTS reservation, but considering that the time scale of a change of the wireless medium quality is far larger than one packet transmission time which is typically less than 1 ms, we can assume no loss in DATA and ACK packet transmission from the protocol perspective once RTS-CTS reservation is successfully done. Note that the same assumption is usually made in the literature for the feasibility of analysis[4].
- The probability of successful packet transmission is independent and identically distributed (i.i.d.) among receivers. The wireless channel condition may impact some receivers by a single source of interference, but it is shown that the analysis result under this assumption can also have a similar trend in real networks with the dependency among receiver states [4]. Also, wireless broadcast transmission typically uses the lowest modulation and coding scheme that can accommodate receivers with the worst signal-to-noise ratio due to local interference. In such cases, receiver channel heterogeneity may not have so much impact on packet reception states of individual receivers.

3.1. Packet Transmission Delay

For the protocol Pb, the source waits for RTS-CTS rounds until all the receivers are ready to receive a packet. Thus, at the very one data transmission attempt, the source is able to send a data packet to all the receivers without loss.

Since $\Pr[X^{Pb}=x]=(1-(1-c)^n)^{x-1}(1-c)^n$, we easily obtain,

$$\begin{aligned} E[T^{Pb}] &= T_c \sum_{x=1}^{\infty} x \Pr[X^{Pb}=x] + T_d \\ &= T_c \sum_{x=1}^{\infty} x \cdot (1-(1-c)^n)^{x-1}(1-c)^n + T_d \\ &= T_c \frac{1}{(1-c)^n} + T_d \end{aligned} \tag{1}$$

For the protocol *Pu*, we divide computation of X^{Pu} into (1) how many attempts are required and (2) how much time the source waits before each attempt of them. Thus,

$$E[T^{Pu}] = \sum_{m=1}^{n} \Pr[M^{Pu}=m]\left(T_c \sum_{x=m}^{\infty} x \Pr[X^{Pu}=x|M^{Pu}=m] + mT_d\right)$$

$$= \sum_{m=1}^{n}\left(\Pr[M^{Pu}=m]\left(T_c \sum_{x_1=1}^{\infty} \cdots \sum_{x_m=1}^{\infty} (x_1+\cdots+x_m) \Pr[X_1^{Pu}=x_1,\ldots,X_m^{Pu}=x_m]\right)+mT_d\right)$$

$$\begin{aligned} &= \sum_{m=1}^{n} \Pr[M^{Pu}=m]\left(T_c E[X_1^{Pu}+\cdots+X_m^{Pu}] + mT_d\right) \\ &= \sum_{m=1}^{n} \Pr[M^{Pu}=m]\left(T_c \frac{m}{1-c} + mT_d\right) \end{aligned} \tag{2}$$

At the last step, we used the fact that X_1^{Pu}, ..., X_m^{Pu} are independent and $E[X_1^{Pu}]=\cdots=E[X_m^{Pu}]=\sum_{x=1}^{\infty} x c^{x-1}(1-c) = 1/(1-c)$.

$\Pr[M^{Pu}=m]$ is derived as follows. The source polls the readiness state of each receiver in sequence over the set {1, 2, ..., n}. Let z_i denote the receiver on which *i*-th attempt is made. Obviously 1-st attempt is made on receiver 1 (z_1=1). In order that $M^{Pu}=m$ (m>1), each receiver z_i ($2 \leq i \leq m$) should experience collision at all of the first *i*-1 among *m* attempts, and that probability is c^{i-1} for receiver z_i and the probability for all receivers from z_2 to z_m is $\prod_{i=2}^{m} c^{i-1} = c^{\frac{(m-1)m}{2}}$. All other receivers between z_i and z_{i+1} ($i<m$) should receive the broadcast packet within the first *i* attempts, and that probability is $(1-c^i)^{z_{i+1}-z_{i-1}}$. For *i*=*m*, the probability is $(1-c^m)^{n-z_m}$. Therefore, for 1<*m*,

$$\Pr[M^{Pu}=m] = \sum_{2 \leq z_2 < \cdots < z_m \leq n} c^{\frac{(m-1)m}{2}} (1-c)^{z_2-z_1-1} \cdots (1-c^m)^{n-z_m} \tag{3}$$

Obviously, for m=1, $\Pr[M^{Pu}=1]=(1-c)^{n-1}$. One receiver is successfully polled when it is ready to receive a data packet. Thus, the probability that all the other n-1 receivers can successfully receive a data packet is $(1-c)^{n-1}$.

In a similar manner, by using the fact that X_1^{Pm}, ..., X_m^{Pm} are independent and for all $1 \leq i \leq m$,

$$E[X_i^{Pm}] = \sum_{x=1}^{\infty} x \cdot (1-(1-c)^2))^{x-1}(1-c)^2 = \frac{1}{(1-c)^2} \tag{4}$$

we have

$$E[T^{Pm}] = \sum_{m=1}^{\lceil \frac{n}{2} \rceil} \Pr[M^{Pm}=m]\left(T_c \frac{m}{(1-c)^2} + mT_d\right) \tag{5}$$

$$\begin{aligned}\Pr[M^{Pm}=m] = &\sum_{3 \leq z_3 < \cdots < z_{2m-1} < z_{2m} \leq n} \left(c^{(m-1)m}(1-c)^{z_4-z_2-2} \cdots (1-c^m)^{n-z_{2m}}\right) \\ &+ \sum_{3 \leq z_3 < \cdots < z_{2m-1} \leq n} \left(c^{(m-1)(m-1)}(1-c)^{z_4-z_2-2} \cdots (1-c^{m-1})^{n-z_{2m-2}-1}\right)\end{aligned} \tag{6}$$

$\Pr[M^{Pm}=m]$ is derived as follows. The source polls two receivers at a time. Let z_{2i-1} and z_{2i} denote the two receivers on which i-th attempt is made. Obviously, z_1=1 and z_2=2. In order that $M^{Pm}=m$ (m>1), receiver z_{2i-1} and z_{2i} ($2 \leq i \leq m$) should experience collision at all of the first i-1 among m attempts , and that probability is c^{i-1} for receiver z_{2i-1} and z_{2i}. The probability for all receivers from z_3 to z_{2m} is $\prod_{i=2}^{m} c^{i-1}c^{i-1} = c^{(m-1)m}$. All other receivers between z_{2i} and $z_{2(i+1)}$ (i<m) should receive the broadcast packet within the first i attempts, and that probability is $(1-c^i)^{z_{2(i+1)}-z_{2i}-2}$. For i=m, the probability is $(1-c^m)^{n-z_{2m}}$.

There is another case to consider where the source polls one receiver for retransmission if the receiver is the only node that did not get a packet yet. The second term in Equation 6 corresponds to the exceptional case. The probability for all receivers from z_3 to z_{2m-1} is $\left(\prod_{i=2}^{m} c^{i-1}c^{i-1}\right)c^{m-1} = c^{(m-1)(m-1)}$. All receivers after z_{2m-2} should receive the packet within the first m-1 attempts, except one receiver z_{2m-1}. For $1<m$,

For m=1, $\Pr[M^{Pm}=1]=(1-c)^{n-2}$. Two receivers are successfully polled when they are ready to receive a data packet. Thus, the probability that all the other n-2 receivers can successfully receive a data packet is $(1-c)^{n-2}$.

3.2. Packet Stable Time

Let us derive the packet stable time for each protocol. For protocol Pb, the source should finish the successful transmission of n packets to n receivers, before gathering n acknowledgments for a packet.

$$E[S^{Pb}] = n\left(T_c \frac{1}{(1-c)^n} + T_d\right) \tag{7}$$

For protocol Pu, the source should finish successful transmission of the packet (say k-th packet) to n receivers, the next packet ((k+1)-th) to n-1 receivers, ..., and the (k+n-1)-th packet to 1 receiver, before gathering n acknowledgments for the packet. Note that, in Equation 8, $\Pr[M^{Pu}=m]$ is calculated for i receivers instead of n in Equation 3.

$$E[S^{Pu}] = \sum_{i=1}^{n}\sum_{m=1}^{i}\left(\Pr[M^{Pu}=m]\left(T_c \frac{m}{1-c} + mT_d\right)\right) \tag{8}$$

For protocol Pm, the source should finish successful transmission of the packet (say k-th packet) to n receivers, the next packet ((k+1)-th) to n-2 receivers, ..., and the (k+n/2-1)-th packet to 2 receivers, before gathering n acknowledgments for the packet. Note that, in Equation 9, $\Pr[M^{Pm}=m]$ is calculated for 2i receivers instead of n in Equation 3.

$$E[S^{Pm}] = \sum_{i=1}^{\lceil \frac{n}{2} \rceil}\sum_{m=1}^{i}\left(\Pr[M^{Pm}=m]\left(T_c \frac{m}{(1-c)^2} + mT_d\right)\right) \tag{9}$$

3.3. Asymptotic Complexity

Regarding asymptotic complexity, we can see from the numerical analysis that the all-polling feedback has exponential complexity with the number of receivers while the 2-polling feedback and 1-polling feedback have complexity bounded by the order of the square of the number of receivers.

Table 2. Asymptotic complexity

Protocol	Packet transmission delay	Packet stable time
Pb	$O\left((\frac{1}{1-c})^n\right)$	$O\left(n(\frac{1}{1-c})^n\right)$
Pu	$O(n^2)$	$O(n^3)$
Pm	$O(n^2)$	$O(n^3)$

3.4. Network Traffic

Network traffic analysis is done during the derivation of packet delay. We analyze the byte size of transmitted control packets and data packets needed for stability of a single data packet We have the byte size of control packets ($N^{Pb}{}_c$) and the byte size of data packets ($N^{Pb}{}_d$) for protocol *Pb*.

$$E[N_c^{Pb}] = B_c\frac{1}{(1-c)^n}$$
$$E[N_d^{Pb}] = B_d \quad (10)$$

Similarly, for protocol *Pu* and *Pm*, we obtain

$$E[N_c^{Pu}] = B_c\sum_{m=1}^{n}\Pr[M^{Pu}=m]\frac{m}{1-c}$$
$$E[N_d^{Pu}] = B_d\sum_{m=1}^{n}\Pr[M^{Pu}=m]m \quad (11)$$

$$E[N_c^{Pm}] = B_c\sum_{m=1}^{\lceil\frac{n}{2}\rceil}\Pr[M^{Pm}=m]\frac{m}{(1-c)^2}$$
$$E[N_d^{Pm}] = B_d\sum_{m=1}^{\lceil\frac{n}{2}\rceil}\Pr[M^{Pm}=m]m \quad (12)$$

Ⅳ. Results

4.1. Analysis Results

The parameters used for numerical evaluation are T_c and T_d. The payload size for a data packet is set to 2048 bytes. Thus, one data packet size is 2082 bytes including headers and frame check sequence. An RTS packet size, a CTS packet size and an ACK packet size are 20, 14, and 14 bytes, respectively. Data packets and ACK packets are transmitted at the maximum rate while RTS and CTS packets are transmitted at the basic rate of the underlying wireless medium. For IEEE 802.11a where the maximum data rate is 54 Mbps and the basic rate is 6 Mbps, T_c (RTS, CTS, and 2 SIFS) is 25 + 17 + 2*16 = 74 us and T_d (DATA, ACK, and 2 SIFS) is 294 + 2 + 2*16 = 328 us.

Fig. 2 depicts the average packet delay as the number of broadcast receivers varies. Figure 3 depicts the packet stable time. As *n* increases, packet delay and packet stable time for protocol *Pb* abruptly increase while those for protocol *Pu* and *Pm* are less changed. Thus, for large-scale broadcast, we conclude that it is not a good strategy to wait until all receivers are ready. This is consistent with the asymptotic complexity analysis in the previous section. In small-scale broadcast conditions, protocol *Pm* exhibits packet delay comparable to *Pb* and *Pu*. Note that protocol *Pm* exhibits almost as half stable time as *Pu*. This is because *Pm* enables the source to know the reception status of two receivers, rather than one, at each data transmission attempt. This highlights the benefit of the proposed 2-polling feedback, i.e., we can achieve about 50 % improvement of packet stable time through a simple feedback extension to the current standards.

In Figure 4 and Figure 5, we evaluate the impact of packet loss probability parameter *c*. Larger *c* means higher load in the network while smaller *c* means lower load. In practical ranges of packet loss probability (this should not be confused with bit error rate (BER)), we have an interesting observation for protocol *Pu* and *Pm* that check the readiness state of one or two receivers before transmission: packet delay and stable time slowly increase with the packet loss probability. Meanwhile, the packet delay and stable time of protocol *Pb* are exponential with the packet loss probability. In very low load conditions where packet loss occurs below 5 %, *Pm* still shows similar performance to *Pb* while *Pu* shows less performance than *Pb*. In high load conditions, the probability for all receivers to be ready at one time is low and thus, it is better to check one or two receivers and guarantee delivery to them and

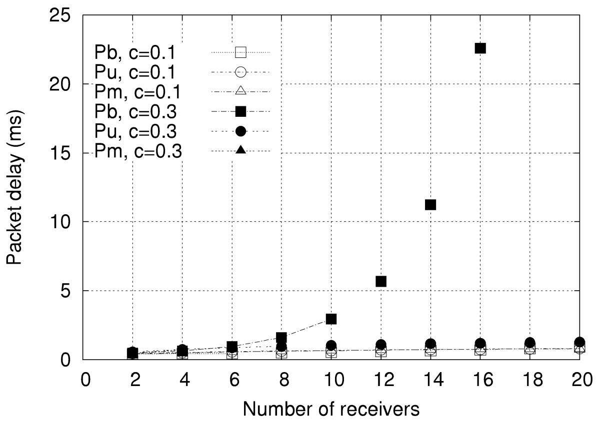

Fig. 2. Packet transmission delay

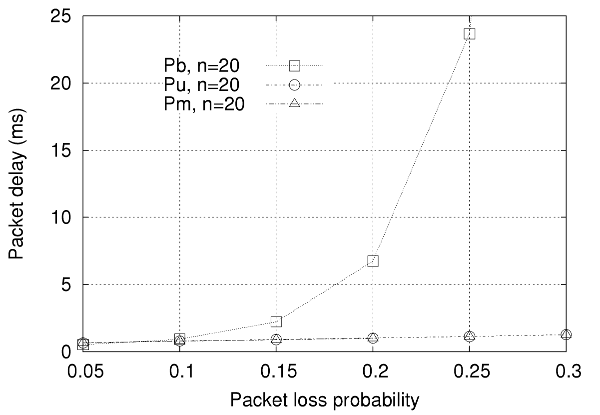

Fig. 4. Impact of packet loss probability on packet transmission delay

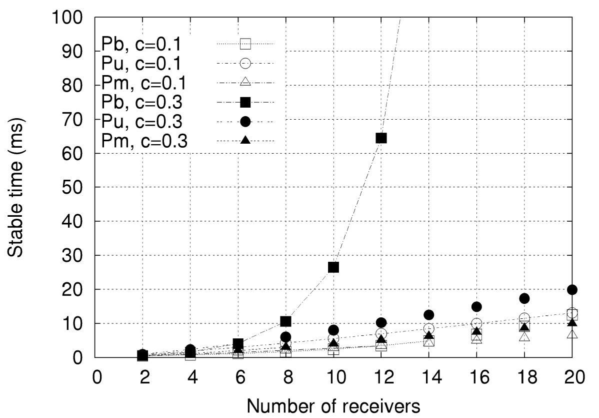

Fig. 3. Packet stable time

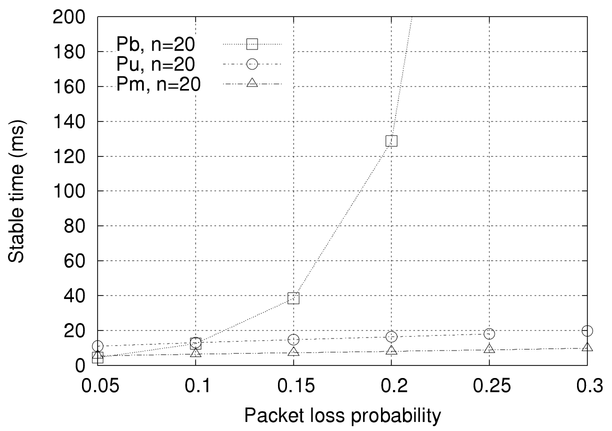

Fig. 5. Impact of packet loss probability on packet stable time

repeat for next receivers. In low load conditions, even though we do not check the readiness state of all the receivers, many of them are likely to be ready and thus, it is enough to check only a few of them before transmission.

To evaluate network performance, we set parameters B_d to 2096 bytes for a data packet and ACK packet, and B_c to 34 bytes for an RTS and CTS packet. We set c to 0.3 for example. Figure 6 plots the network utilization for each protocol in terms of time per packet. Time utilized is the amount of network time used for transmission of control packets or data packets. Note that *Pb* exhibits the heaviest network usage and the lowest network utilization whereas *Pu* and *Pm* have comparable network utilization. This is attributed to the fact that as the number of receivers increases, the number of required RTS-CTS rounds and hence time required for inter-frame space abruptly increases for protocol *Pb*.

4.2. Simulation Results

To verify the analysis model, we conduct a custom simulation written in C and present simulation results. Figure 7 depicts the simulation results for the average packet delay as the number of broadcast receivers varies. Note that since the purpose of simulation is the verification of analysis results, we only present one representative result for c=0.3 and n is between 2 and 20. As seen in Fig. 7, simulation results show similar patterns to analysis results. and thereby verifying the correctness of numerical analysis.

V. Conclusions

In this article, we deal with a stable reliable broadcast problem in wireless networks. We focus

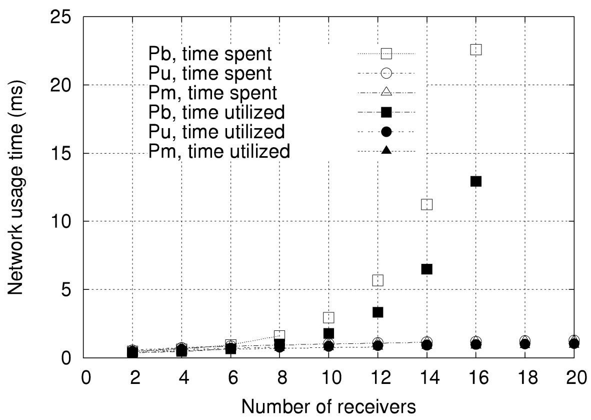

Fig. 6. Network utilization

on efficiency of feedbacks in stable reliable broadcast depending on how many receivers'

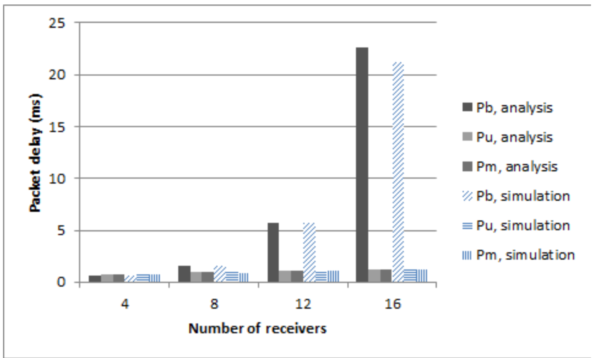

Fig. 7. Simulation results.

status is polled before broadcast transmission and propose a new class called 2-polling feedback which is simple to implement. We make the first formal numerical analysis of these protocols, and then derive asymptotic complexity and give numerical performance evaluation. The results show that the proposed 2-polling feedback scheme achieves comparable packet delay, and guarantees stable delivery with almost as half packet stable time as existing feedback methods.

Although we use a simple two-state packet loss probability to reflect receiver readiness for the feasibility of numerical analysis, it is an interesting future work to extend the analysis with various channel models. It is another direction for future research to implement the classes of protocols in real networks and compare by measurement data.

References

[1] *IEEE 802.11: Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications*, Jun. 2007.

[2] Bluetooth SIG, *Bluetooth core specification version 3.0+HS,* Apr. 2009.

[3] P. Chaporkar, A. Bhat, and S. Sarkar, "An adaptive strategy for maximizing throughput in MAC layer wireless multicast," in *Proc. ACM MobiHoc*, pp. 256-267, May 2004.

[4] P Chaporkar and S. Sarkar, "Minimizing delay in loss-tolerant MAC layer multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4701-4713, Oct. 2006.

[5] S. K. S. Gupta, V. Shankar, and S. Lalwani, "Reliable multicast MAC protocol for wireless LANs," in *Proc. IEEE ICC*, pp. 93-97, May 2003.

[6] J. Kuri and S. K. Kasera, "Reliable multicast in multi-access wireless LANs," *ACM Wireless Networks,* vol. 7, no. 3, pp. 359-369, Jul. 2001.

[7] W. Lou and J. Wu, "A reliable broadcast algorithm with selected acknowledgements in mobile ad hoc networks," in *Proc. IEEE GLOBECOM*, Dec. 2003.

[8] W. Si and C. Li, "RMAC: A reliable multicast MAC protocol for wireless ad hoc networks," in *Proc. IEEE ICPP*, pp. 494-501, Aug. 2004.

[9] M. T. Sun, L. Huang, A. Arora, and T. H. Lai, "MAC layer multicast in IEEE 802.11 wireless networks," in *Proc. IEEE ICPP*, Aug. 2002.

[10] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," in *Proc. IEEE MILCOM*, pp. 1008-1013, Oct. 2001.

[11] T. Zhu, Z. Zhong, T. He, and Z. L. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *Proc. ACM NSDI*, Apr. 2010.

[12] Y. H. Jung and B. G. Kim, "Performance evaluation of common feedback channels for multicast transmission and optimization of

system parameters," *J. KICS,* vol. 3, no. 4, pp. 489-497, Apr. 2009.

[13] R. Oliveira, M. Lu´ıs, L. Bernardo, and R. Dinis, "Towards reliable broadcast in ad hoc networks," *IEEE Commun. Lett.,* vol. 16, no. 3, pp. 314-317, Mar. 2012.

Wonyong Yoon

1996 Seoul National University (B.S.)

1998 Seoul National University (M.S.)

2002 Information and Communications University (Ph.D)

2002~2004 Samsung Advanced Institute of Technology, Senior researcher

2004~2007 University of Illinois at Urbana-Champaign, Visiting postdoctoral researcher

2008~2011 LG Electronics, Chief research engineer

20011~Dong-A University, Department of Electronics Engineering, Assistant professor