KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 6, NO. 6, Jun 2012

Throughput and Delay Analysis of a Network Coding-enabled Cooperative MAC Protocol for Ad Hoc Networks

Jaeshin Jang[1], Sang Wu Kim[2] and Sunghong Wie[3]*
[1] Dept. of Information and Communications Engineering, Inje University
Gimhae, Gyeonnam 621-749 - South Korea
[2] Dept. of Electrical and Computer Engineering, Iowa State University
Ames, Iowa 50011-3060 - USA
[3] Division of Information Technology, The Cyber University of Korea
1-21 Gye-dong, Jongno-gu, Seoul 110-800 - South Korea
[e-mail: icjoseph@inje.ac.kr , : swkim@iastate.edu , satwie@gmail.com]
*Corresponding author: Sunghong Wie

Received August 30, 2011; revised January 20, 2011 and May 18, 2012; accepted June 05, 2012;
Published June 25, 2012

Abstract

Cooperative communications and network coding schemes have been proposed to increase system throughput for ad hoc networks. In this paper, we present throughput and delay analysis of the new network coding-enabled cooperative MAC protocol called NC-MAC, which has been proposed by us in order to significantly enhance system performance. This protocol introduces an approach that can accommodate both cooperative communication and network coding for wireless ad hoc networks by slightly increasing overhead and modifying standards. The protocol's performance is evaluated using mathematical analysis and computer simulation and two performance measures, system throughput and average channel access delay, are used for a performance comparison with previous schemes. It is assumed that all the frames exchanged over a wireless channel are susceptible to transmission errors, which is a new but more reasonable assumption differentiating this research from previous research. Numerical results show this protocol provides significantly enhanced system performance compared with conventional cooperative MAC protocols used in previous research. For instance, system performance is 47% higher using the NC-MAC protocol than when using the rDCF protocol.

Keywords: Ad hoc network, cooperative MAC, DCF, NC-MAC, network coding

---

A preliminary version of this paper appeared in IEEE ICOIN 2010, January 27-29, Busan, Korea. This paper includes a concrete analysis and computer simulation under a more reasonable assumption compared with the preliminary version.
This work was supported by the 2009 Inje University research grant.

http://dx.doi.org/10.3837/tiis.2012.06.010

1. Introduction

Ad hoc networks are more advanced than the mobile communications networks available today. This is because they use homogeneous mobile nodes and they can directly communicate with each other over a wireless channel, which is impossible in mobile communication networks or conventional wireless LANs with access points [1]. However, all wireless communication networks, including ad hoc networks, have a critical problem: limited bandwidth compared with wired networks. Therefore, one of the most challenging problems is how to improve system performance, which involves increasing the frame transmission rate at the data link layer. To achieve this higher transmission rate, received signal-to-noise ratio (SNR) or receiving capability at the receiving node is required to be improved. The IEEE 802.11 wireless local area network (WLAN) standard, including the physical and data link layers, is widely deployed because its distributed coordination function (DCF) functionality is based on carrier sense multiple access with collision avoidnace (CSMA/CA) and it provides good performance by resolving hidden and exposed terminal problems with any one-hop communication [2].

To improve the system performance, recent wireless networks such as worldwide interoperability for microwave access (WiMAX) and WLAN use link adaptation schemes. Such schemes use 64-QAM under good wireless channel conditions to increase the transmission rate, but use QPSK under bad channel conditions for reliable transmission. On the other hand, cooperative communications are introduced to significantly improve system performance [3]. This kind of communication is used when a direct communication link between a source node and a destination node is operating under bad channel conditions, and thus this direct link cannot attain high-rate transmission. In this situation, another node, called a helper node, is involved and relays frames as an intermediate node. Usually, the link conditions between a helper node and a source or a destination node are better than a direct link, and thus the average frame transmission rate increases due to link adaptation. Since the destination node can receive the same frame via more than two senders, the source node and the helper node, this cooperative protocol is sometimes called the distributed MIMO (multi-input multi-output) scheme.

In order to significantly improve system capacity, a network coding scheme is introduced. To help understand the network coding scheme, a simple Bob and Alice model is shown in **Fig. 1**, where Alice and Bob want to exchange a pair of packets via a router [4]. The current approach requires four transmissions for exchanging their packets. In the case of a network coding approach, however, Alice and Bob transmit their respective packets to the router, which combines the two packets based on an XOR coding rule and then broadcasts the XOR-ed version. Alice and Bob can retrieve each other's packet by XOR-ing again with their own packets. Thus, it takes three transmissions instead of four and the saved transmission resource can be used to increase system throughput.

In this paper, we present the performance evaluation of a network coding-enabled cooperative MAC (medium access control) protocol proposed in [15] by us, called the NC-MAC protocol, which is the combination of a cooperative protocol and networking coding scheme, and we evaluate its performance using analysis and simulation. This paper, which is an enhanced version of our previous work, makes two key contributions. One is the performance evaluation of this protocol using both mathematical analysis and computer simulation under the important assumption that all the frames transmitted by any node are susceptible to transmission errors due to a bad wireless channel. This was not discussed in previous related research [5][6][9][10][13]. The other is that two different performance measures, system throughput and average channel access delay, are derived.

A survey of related studies appears in Section 2 and the NC-MAC protocol is described in detail in Section 3. The protocol's performance is evaluated using mathematical equations in Section 4, and numerical results of the analysis and computer simulation are presented in Section 5. Finally, this paper is concluded in Section 6.

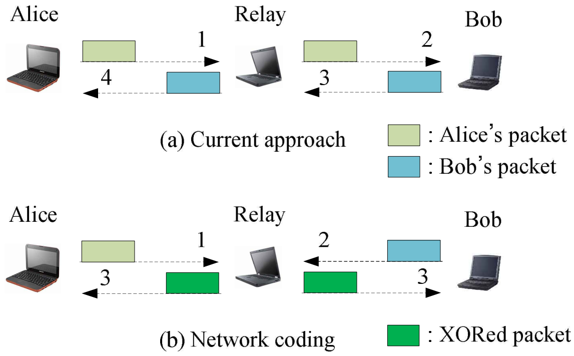


Fig. 1. Bob and Alice model

2. Related Work

Most recent studies about the cooperative MAC protocol for ad hoc networks are based on the IEEE 802.11 WLAN protocol. This WLAN protocol is based on the distributed coordination function (DCF) scheme and is anticipated to be used in the next generation of ad hoc networks. Bianchi [5] evaluated the performance of the DCF protocol using queuing theory. In his research, it was assumed that every node has frames for transmission in its buffer, which is a common assumption called the saturated condition, and that every contention window in a binary exponential backoff procedure doubles its window until the number of backoff stages for each node reaches the designated maximum value. However, this model has one problem: every node tends to send a frame until it finally succeeds in sending this frame, even after the number of backoff stages reaches the designated maximum value.

Tantra [6] solved the problem with Bianchi's model and proposed a new system model that comes close to the standard. In Tantra's model, when the number of retransmissions at a source node reaches the designated maximum value, the node terminates its retransmission and discards its related frame. According to a numerical result not shown in this paper, those two models have a large difference in system throughput, especially when the traffic load is heavy. On the other hand, wireless LAN standards such as IEEE 802.11a and IEEE 802.11g use the link adaptation scheme, which chooses a modulation and channel coding scheme based on the current channel status (i.e., received SNR value). The receiver-based auto rate (RBAR) scheme, which came up with a modifed frame header format needed for implementing the link adaptation scheme, was proposed in [7]. According to this scheme, the appropriated frame transmission rate is decided in the receiver node based on the measured current channel quality and this idea opened the study on cooperative MAC protocol in the link layer.

Based on the wireless LAN protocol currently available, various cooperative MAC protocols were suggested [8][9][10][11]. The CMAC protocol and the FCMAC protocol, which is a combination of the CMAC protocol and an error control scheme, were proposed in [8]. This cooperative MAC protocol can be implemented with slight modifications to the current IEEE 802.11 WLAN standard, which is one of its advantages; however, it doesn't specify how to select helper nodes. Zhu and Cao proposed an rDCF scheme that is based on the DCF protocol [9]. In this protocol, every node calculates the received SNR value (or corresponding frame transmission rate) for all the wireless channels between neighboring nodes and itself by checking the received frames. The nodes store the SNR values in their willing list and periodically share that list with other adjacent nodes. Nodes receiving the willing list store it in their relay tables. Whenever a packet is generated in a source node, the relay table is searched first to find a helper node that can provide a better transmission link than the direct transmission link between the source node and the destination node. If an appropriate helper node exists, then the source node transmits an RRTS1 (Relay RTS1) frame for cooperative communication to the helper node rather than the destination node. A CoopMAC protocol that has different procedures to transmit control frames was proposed in [10].

Overall procedures of the rDCF scheme and the CoopMAC scheme to exchange control and data frames are quite similar but it is noticeable that the CoopMAC protocol clearly explains its procedure to choose helper nodes using mathematical equations. The performance analyses of rDCF scheme and CoopMAC scheme were given in their works, respectively, under the assumptions that transmission errors for bad wireless channel do not take place and that there are always packets in source nodes' buffers. As a performance measure, they used only saturated throughput and their numerical results were compared with those of RBAR scheme. On the other hand, a new reliable cooperative MAC protocol, called RCO-MAC, was proposed in [11] by us, too. In order to overcome the erroneous wireless channels, this protocol proposed a new retransmission scheme where the retransmission responsibility for the frames experiencing transmission failure is given to any sending node. For a performance evaluation using a computer simulation, it was assumed that all frames are susceptible to transmission errors.

There have been lots of efforts on network-coded cooperative communications, some of which include [12][13][14][15]. Liu and Xue [12] suggested an opportunistic network coding scheduling to make up one drawback of network coding and showed that network coding achieves the maximum gain when the traffic is symmetric. Argyriou [13] suggested an extended distributed MAC protocol optimizing the transmission of network coded packets both with using opportunistic acknowledgements and maintaining virtual buffers. Performance analysis and simulation of this protocol were carried out and the throughput gain as a performance measure was derived. Sengupta et al. [14] derived a theoretical formulation for computing the throughput of network coded wireless networks including COPE-type network coding. They also suggested that network coding-aware routing protocol improves the throughput significantly and that throughput improvements are, however, dependent on the network structure, traffic pattern, and whether opportunistic listening is employed or not. The NC-MAC protocol is proposed in [15] by us and this paper is an enhanced one from our previous work [15], where the system performance evaluation was carried out only by computer simulation and frame transmission errors were assumed to take place only in the transmission of the CRTS frames.

3. NC-MAC Protocol

The NC-MAC protocol, our previous work [15], uses only one helper node with the best wireless channel status as a cooperative node and it is assumed that the helper node with the best wireless channel status between the source node or the destination node and itself can be chosen before frame transmission begins. This is because the same method as that used in the previous research [9][10] to decide the best helper node beforehand can be used for the NC-MAC protocol without any problem. In order to use this helper node decision mechanism, every node should monitor all the control frames passing by, although the node is not those frames' final destination, and measure the received SNR value to derive the appropriate frame transmission rate that the current wireless channel can afford between any neighboring node and itself. Every node should maintain a new database, a ***help_list*** table, where candidates for helper nodes and possible frame transmission rates among its neighboring nodes are stored. In addition, every node should exchange this information with its neighbors. However, this overhead procedure is not considered for the performance evaluation in this paper.

The flow chart for the NC-MAC protocol to depict how it works all at once appears in **Fig. 2**. Detailed descriptions of how source, helper, and destination nodes work will follow. The working procedure at a source node for the NC-MAC protocol is shown in **Fig. 3**. As soon as any data packet from the upper layer arrives at the MAC layer of the source node, the node generates a cooperative RTS (CRTS) frame and sends it to the assigned helper node using the CSMA/CA contention mechanism. The CRTS frame that is an extended version of the traditional RTS frame includes the following additional fields in its header:

- helper node address
- frame transmission rate between the helper node and destination node, R_{hd}
- frame transmission rate between the source node and helper node, R_{sh}.

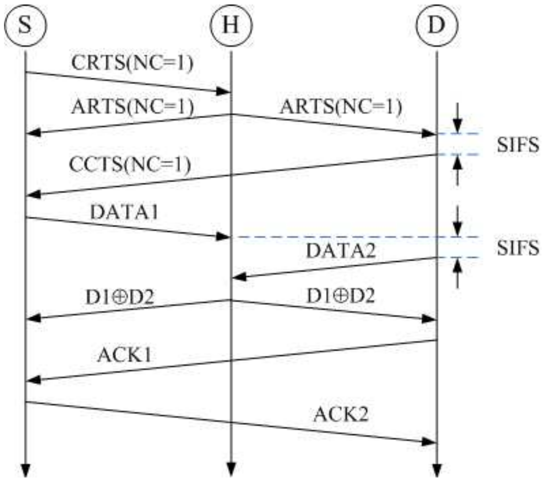


Fig. 2. Flow chart for the NC-MAC protocol

In order to decide the best helper node, the source node searches its ***help_list*** table and those related frame transmission rates among the source node, the helper node, and the destination node. The frame transmission rate is the maximum permissible transmission rate meeting the FER (frame error rate) requirement between two nodes and it is derived based on the received SNR value and the chosen modulation and coding scheme. The maximum frame transmission rates under any wireless channel can be calculated and stored in advance using either an analytic method or direct measurement. The CRTS frames includes another one-bit field, NC (network coding), in its header that is used to inquire whether the destination node has any data frame to send to the source node. If the source node wants to use the network coding scheme, it makes NC=1. Since this NC-MAC protocol uses the same channel contention mechanism as the DCF scheme, the source node is in charge of error recovery by retransmitting the erroneous packets. For this type of error recovery, the source node has three different types of retransmission timers, which are set with the following values:

- Timer 1: $T_{CRTS} + T_{ARTS} + T_{CCTS} + 2*SIFS$ (1)
- Timer 2: $T_{DATA1} + T_{DATA2} + T_{DATA1 \oplus DATA2} + 2*SIFS$
- Timer 3: $T_{ACK1} + SIFS$,

where T_{CRTS} means the CRTS frame transmission time when the CRTS frame is transmitted at the basic rate and SIFS stands for short inter-frame space. The source node triggers the retransmission timer 1 as soon as it sends the CRTS frame to the helper node. The value of timer 1 should be large enough for the source node to receive the cooperative CTS (CCTS) frame from the destination node in case of successful transmission before timer 1 expires. When this timer expires, the source node increases the contention window by two times and returns to the start to transmit the CRTS frame again. When the source node receives the CCTS frame before timer 1 expires, it sends the DATA1 frame after the SIFS period to the helper node at the rate R_{sh}, which is obtained from the CCTS frame, and triggers retransmission timer 2 for the immediate error recovery of the DATA1 frame. The value of timer 2 is slightly larger than the required delay for the source node to receive the DATA1⊕DATA2 frame from the helper node. If timer 2 expires, then the source node returns to the start to retransmit the CRTS frame with doubled contention window. Otherwise, the node decodes the received DATA1⊕DATA2 frame and determines whether the DATA2 frame is in transmission error. This source node also triggers another retransmission timer 3 to ensure the successful transmission of the DATA1 frame to the destination node. The value of timer 3 is large enough that the source node receives the ACK1 frame from the destination node before timer 3 expires. If the source node does not receive the ACK1 frame before timer 3 expires, then the node begins the retransmission procedure for the CRTS frame. The source node sends the ACK2 frame to the destination if the decoded DATA2 frame is found to be successful. The source node ends its role by sending this ACK2 frame and then it returns to the start with its contention window set with the initial value CW_{min}.

The working procedure at the helper node for the NC-MAC protocol is shown in **Fig. 4-(a)**. The helper node's role for the NC-MAC protocol starts when it receives any frame from the source node. If the received frame is the CRTS frame, it measures the received SNR value, derives the maximum frame transmission rate between the source node and itself, and then updates the R_{sh} value in the CRTS frame if those two values are different. The node creates the acknowledge RTS (ARTS) frame after the CRTS frame and puts the new R_{sh} value and the R_{hd} value extracted from the CRTS frame into the ARTS frame. After the helper node copies the NC field from the CRTS frame into the ARTS frame, it transmits this ARTS frame to the destination node. On the other hand, if the received frame from the source node is the DATA1 frame, then it should wait for the DATA2 frame to arrive. After the node receives the DATA2 frame successfully, it combines the two data frames based on an XOR coding rule and then broadcasts the combined DATA1⊕DATA2 frame to the source node and destination node. In this case, the helper node chooses the minimum frame transmission rate between R_{sh} and R_{hd} included in the CCTS frame.

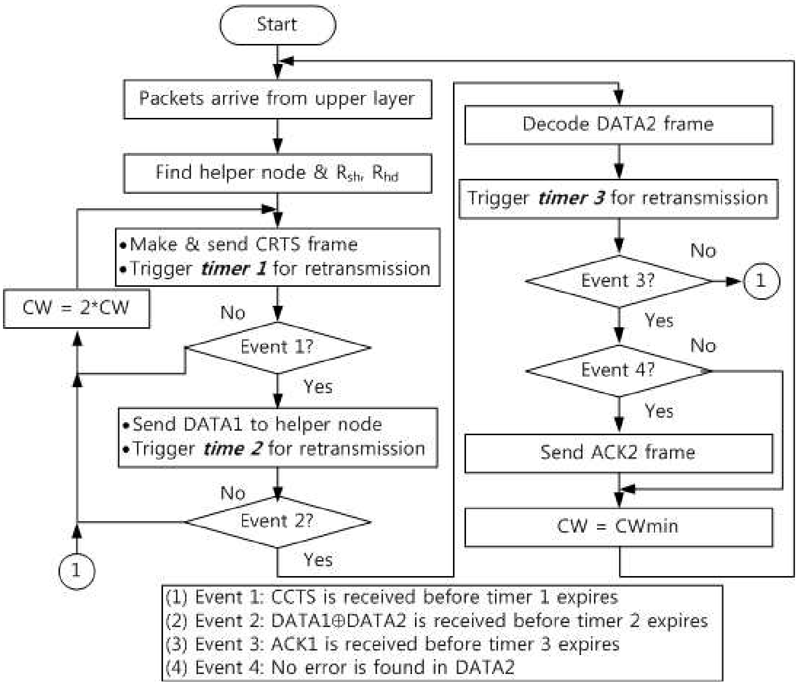
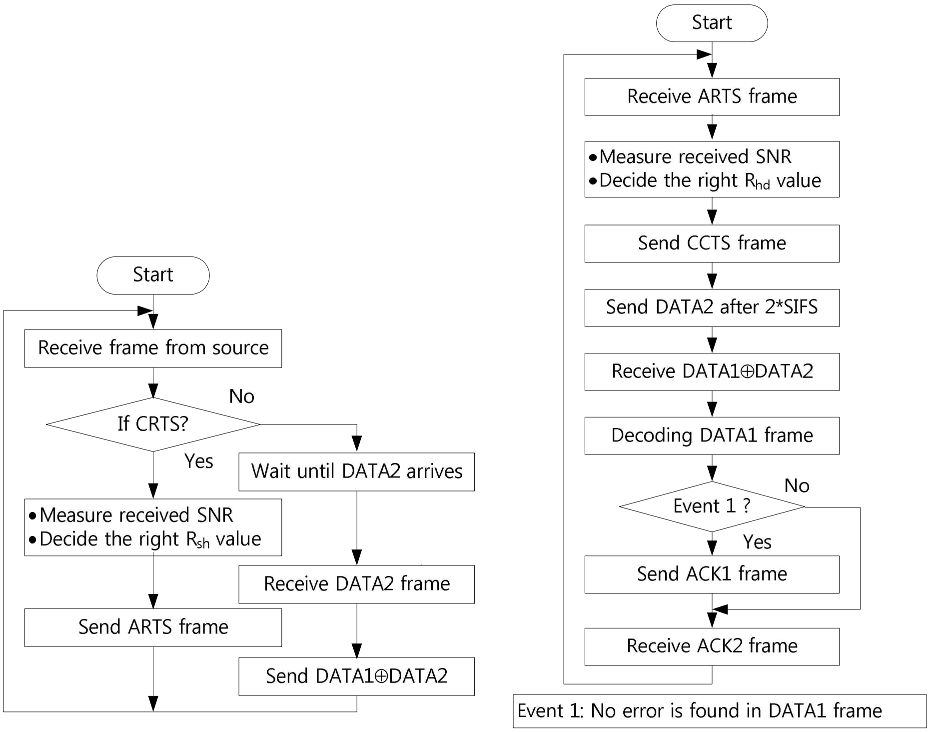


Fig. 3. Working procedure of the source node

The working procedure at the destination node for the NC-MAC protocol is shown in **Fig. 4-(b)**. When the destination node receives the ARTS frame from the helper node, it measures the received SNR value, derives the maximum frame transmission rate between the helper node and itself, and then updates the R_{hd} value in the ARTS frame with the newly calculated one. The node makes the CCTS frame after the ARTS frame and puts two values, R_{sh} and R_{hd}, into the CCTS frame. If this node has any data frame to send to the source node, it sets the NC field in the CCTS frame header as NC=1 and then sends this frame to the source node directly. This node makes the DATA2 frame, whose destination is the source node, and sends it to the helper node after 2 × SIFS plus DATA frame transmission delay. When this node receives the DATA1⊕DATA2 frame from the helper node, it starts decoding the DATA1 frame to check whether this frame was received successfully. If this decoded frame is found to be errorless, then this node sends the ACK1 frame to the source node after the end of SIFS period. After this node receives the ACK2 frame from the source node as the answer to the DATA2 frame, its whole role ends.

4. Performance Analysis

We adopt the queuing theory for the analytic method. This analytic method is used to obtain system throughput with a saturated traffic model where every node always has packets to transmit. This saturated traffic model is commonly used to derive the maximum system throughput. However, the two key differences compared to the analysis procedure used in Tantra's work [6] are that all the frames used in this paper undergo erroneous transmission due to bad wireless channel environments and that a new performance measure, average channel access delay, is derived. In this paper, the performance evaluation is limited only to a single cell and therefore, hidden and exposed terminal problems are not considered. It is assumed that there is always one helper node with the best wireless channel between source node and destination node. One communication group consists of three communicating nodes, i.e., the source node, the helper node, and the destination node, and it is also assumed there are K communication groups in a single cell. Therefore, there are K source nodes that are always ready to send CRTS frames. In addition, it is assumed that frame transmission collision happens only when more than one source node transmits CRTS frames simultaneouly. A frame exchange scenario for the NC-MAC protocol with a detailed time diagram is shown in **Fig. 5** as it is useful for the performance analysis.



(a) Helper node (b) Destination node

Fig. 4. Working procedure of the helper node and destination node

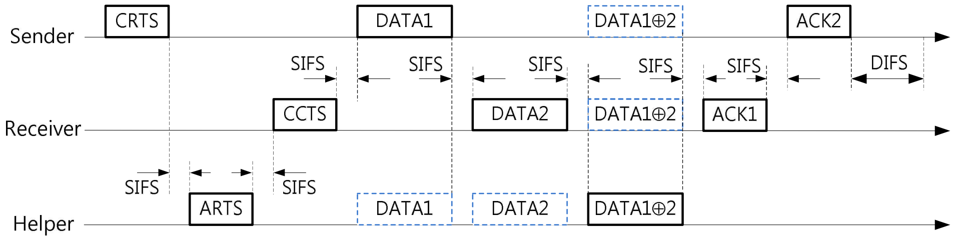


Fig. 5. An example of frame exchanges in the NC-MAC protocol

First of all, let's define several variables used in this performance evaluation.

- τ: CRTS frame transmission probability at a source node

- $\mathrm{p_c}$: CRTS frame collision probability
- $\mathrm{p_m(p_d)}$: control (data) frame transmission error probability due to a poor wirelss channel status
- $\mathrm{p_f}$: CRTS frame transmission failure probability due to poor wireless channel status or collision
- m: maximum backoff stage of the contention window where $\mathrm{CW_{max} = 2^m CW_{min}}$
- r: maximum number of retransmissions in a binary exponential backoff scheme

Transmission failures of CRTS frames are caused by either CRTS frame collision or transmission error. The probability of CRTS frame transmission failures in the NC-MAC protocol can be described as $\mathrm{p_f(\ p_f = p_m + p_c - p_c p_m)}$, the same as in the rDCF, the CODE or, the CoopMAC protocol where the CRTS frame collision probability $\mathrm{p_c}$ can be obtained with $\mathrm{p_c = 1 - (1-\tau)^{K-1}}$. The size of the contention window used in the NC-MAC protocol is represented as follows:

$$\mathrm{W_i} = \begin{cases} \mathrm{2^i CW_{min}}, & \mathrm{0 \le i \le m} \\ \mathrm{2^m CW_{min}} & \mathrm{m} < i \le r. \end{cases} \tag{2}$$

Let's define the following three system state variables for performance analysis of the NC-MAC protocol:

- $\mathrm{b(t)}$: backoff stage of a source node, $\mathrm{b(t) = 0, 1, 2, \cdots, r}$
- $\mathrm{c(t)}$: value of the backoff counter, $\mathrm{c(t) = 0, 1, 2, \cdots, W_{b(t)} - 1}$
- $\mathrm{o(t)}$: frame transmission phase, $\mathrm{o(t) = 0, 1, 2, \cdots, 7}$,

where the last variable $\mathrm{o(t)}$ represents the sending phase for each frame, i.e., 0 for backoff and CRTS, 1 for ARTS, 2 for CCTS, 3 for DATA1, 4 for DATA2, 5 for DATA1⊕2, 6 for ACK1, and finally 7 for ACK2. Therefore, the system vector $\mathrm{\{b(t), c(t), o(t)\}}$ represents how the NC-MAC protocol works clearly and thus this vector becomes a discrete-time Markov chain. The state transition diagrams are shown in **Fig. 6** to **Fig. 8**. The system states described as $\mathrm{(i, j, 0)}$ in **Fig. 6** are the same as those in the previous studies [6][9][10], but the system states described as $\mathrm{S_i}$ in **Fig. 6** or $\mathrm{(i, 0, k)}$ in **Fig. 7** and **Fig. 8** are newly added in this paper to describe the exchange of the ARTS, CCTS, DATA1, DATA2, DATA1⊕2, ACK1, and ACK2 frames. When a source node has data frames to send, it should first monitor for a period of DCF inter-frame space (DIFS) to determine whether the wireless channel is busy or idle. If the wireless channel is sensed to be idle for that period, the source node will transmit its frame immediately. However, if the channel is busy, the node will wait until the end of the busy period and a random backoff interval is chosen, which is shown in the first row of **Fig. 6**. The backoff counter is decremented whenever the channel is detected as being idle for the slot duration, which is shown as any state (i, j, 0) in **Fig. 6** moving to its neighboring state (i, j-1, 0) with probability 1, and it is paused when the channel is detected as being busy. The paused backoff counter is reactivated when the channel is found to be idle for more than a DIFS period. The source node transmits its CRTS frame when the backoff counter becomes zero, which is shown as any state (i, 0, 0) for $0 \le \mathrm{i} \le \mathrm{r}$ in **Fig. 6**. When the source node fails to send its CRTS frame, it chooses a new random backoff distributed uniformly in the doubly increased range (0, 2×CW-1), which is shown as any state (i, 0, 0) in **Fig. 6** moving to another state in the upper backoff stage (i+1, j, 0) of which the selection is decided on a uniform distribution.

Fig. 7 and **Fig. 8** show state transition diagrams representing frame exchanges from the ARTS to ACK2 frames. These two figures were unnecessary in the previous research due to the assumption that there was no transmission error due to a poor wireless channel. **Fig. 7** shows the node's state when its backoff stage is in the range (0, r-1) and **Fig. 8** shows the node's state when its backoff stage is in r. In order to explain these state transition diagrams in detail, let's consider one source node in the ARTS frame transmission phase.

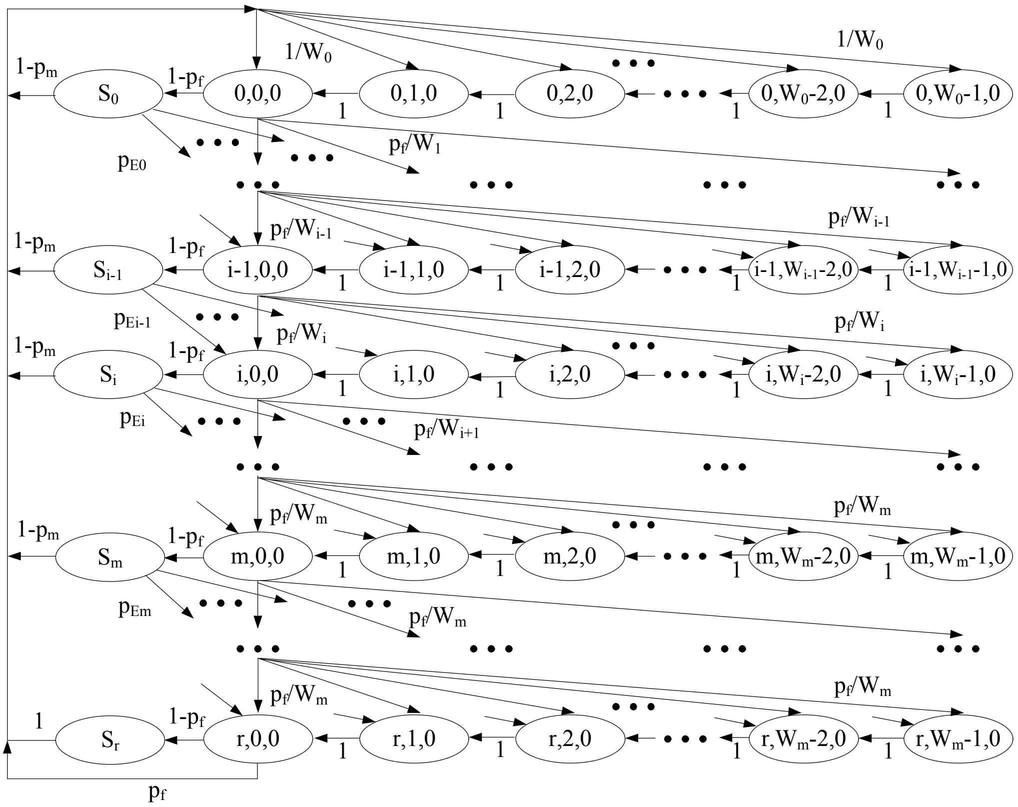


Fig. 6. State transition diagram for the NC-MAC protocol

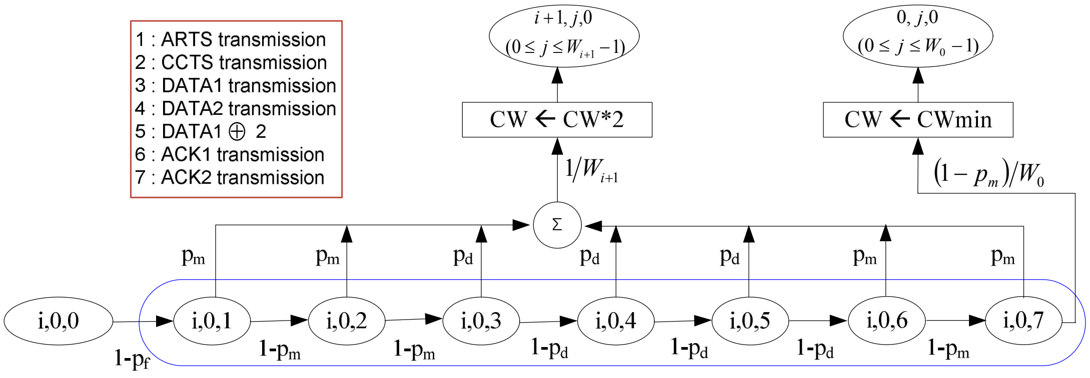


Fig. 7. State transition diagram for $S_i, 0 \le i \le r-1$

The ARTS frame transmission failure by the helper node takes place when transmission of the ARTS frame fails once due to a bad wireless channel. This transmission failure is then detected by the timeout of the retransmission timer in the source node and, according to the error recovery mechanism with retransmissions, the source node enters into a retransmission phase by increasing the contention window by two times. In this case, a system state (i, 0, k) at $b(t) = i$ moves on to one of the upper system states (i+1, j, 0) at $b(t) = i + 1$. However, the system states $(r, 0, k)$ for $0 \leq k \leq 7$ in **Fig. 8**, where a transmitted frame is in error, and the system states $(i, 0, 7)$ for $0 \leq i \leq r - 1$ in **Fig. 7** receiving the successful ACK2 frame, return to the start for sending another new data frame.

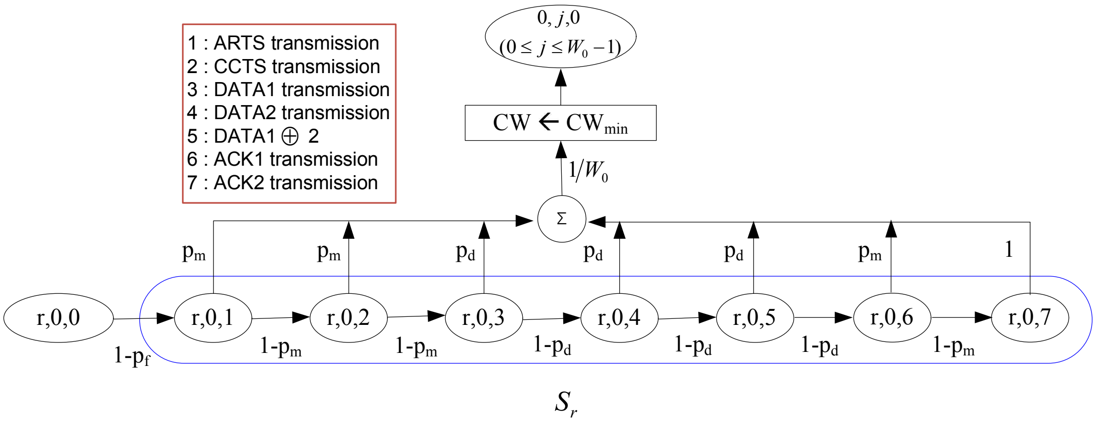


Fig. 8. State transition diagram for S_r

Let's define the steady state probability $\alpha_{i,j,k} \equiv \text{Prob.}\{b(t) = i, c(t) = j, o(t) = k\}$. Then, the balance equations from **Fig. 6** to **Fig. 8** are given by

$$\alpha_{0,0,0} = \alpha_{r,0,0}p_f + \sum_{k=1}^{2} \alpha_{r,0,k}\, p_m + \sum_{k=3}^{5} \alpha_{r,0,k}\, p_d + \alpha_{r,0,6}p_m + \alpha_{r,0,7} + \sum_{i=0}^{r-1} \alpha_{i,0,7}\,(1 - p_m) \tag{3}$$

$$\begin{aligned} \alpha_{i,0,0} &= \alpha_{i-1,0,0}p_f + \sum_{k=1}^{2} \alpha_{i-1,0,k}\, p_m + \sum_{k=3}^{5} \alpha_{i-1,0,k}\, p_d + \sum_{k=6}^{7} \alpha_{i-1,0,k}\, p_m, \qquad 1 \leq i \leq r \\ &= \alpha_{i-1,0,0}p_f + A(i-1) \end{aligned} \tag{4}$$

$$\begin{aligned} \alpha_{i,j,0} &= \alpha_{i-1,0,0}\frac{p_f}{W_i} + \frac{A(i-1)}{W_i} + \alpha_{i,j+1,0} = \alpha_{i-1,0,0}\frac{(W_i - j)}{W_i}p_f + A(i-1)\frac{(W_i - j)}{W_i} \\ &= \frac{(W_i - j)}{W_i}\alpha_{i,0,0}, \qquad 1 \leq i \leq r, 1 \leq j \leq W_i - 1 \end{aligned} \tag{5}$$

$$\alpha_{i,0,k} = \begin{cases} (1 - p_f)(1 - p_m)^{k-1}\alpha_{i,0,0}, & 1 \leq k \leq 3 \\ (1 - p_f)(1 - p_m)^2(1 - p_d)^{k-3}\alpha_{i,0,0}, & 3 < k \leq 6, 0 \leq i \leq r \\ (1 - p_f)(1 - p_m)^3(1 - p_d)^3\alpha_{i,0,0}, & k = 7, \end{cases} \tag{6}$$

where $A(i) \equiv \sum_{k=1}^{2} \alpha_{i,0,k}\, p_m + \sum_{k=3}^{5} \alpha_{i,0,k}\, p_d + \sum_{k=6}^{7} \alpha_{i,0,k}\, p_m$. The above four equations can be modified into another forms in order for all the steady state probabilities to have the probability $\alpha_{0,0,0}$. First of all, Eq. (3) and balance equations representing states in the first column in **Fig. 6** are merged into the following form:

$$\alpha_{0,j,0} = \alpha_{0,0,0} - \frac{j}{W_0}C,\ \ 1 \le j \le W_0 - 1, \tag{7}$$

where

$$C \equiv \sum_{i=0}^{r-1} \alpha_{i,0,7}(1-p_m) + \alpha_{r,0,0}p_f + \sum_{k=1}^{2} \alpha_{r,0,k}p_m + \sum_{k=3}^{5} \alpha_{r,0,k}p_d + \alpha_{r,0,6}p_m + \alpha_{r,0,7}.$$

Eq. (4) can be changed into another form, too. First of all, the following equation is given in the case where $i = 1$.

$$\begin{aligned}
\alpha_{1,0,0} &= \alpha_{0,0,0}p_f + \sum_{k=1}^{2} \alpha_{0,0,k}p_m + \sum_{k=3}^{3} \alpha_{0,0,k}p_d + \sum_{k=6}^{7} \alpha_{0,0,k}p_m \\
&= \alpha_{0,0,0}p_f + p_m(1-p_f)\alpha_{0,0,0}\sum_{k=1}^{2}(1-p_m)^{k-1} + p_d(1-p_f)(1-p_m)^2\sum_{k=3}^{3}(1-p_d)^{k-3} \\
&\quad + p_m(1-p_f)\alpha_{0,0,0}(1-p_d)^3\sum_{k=6}^{7}(1-p_m)^{k-4} \\
&= \Bigg[+p_m(1-p_f)\sum_{k=1}^{2}(1-p_m)^{k-1} + p_d(1-p_f)(1-p_m)^2\sum_{k=3}^{5}(1-p_d)^{k-3} \\
&\quad + p_m(1-p_f)(1-p_d)^3\sum_{k=6}^{7}(1-p_m)^{k-4}\Bigg]\alpha_{0,0,0} = (p_f + B)\alpha_{0,0,0}
\end{aligned}$$

Therefore, Eq. (4) is given by

$$\therefore\ \alpha_{i,0,0} = (p_f + B)^i\alpha_{0,0,0}, \tag{8}$$

where

$$B \equiv p_m(1-p_f)\sum_{k=1}^{2}(1-p_m)^{k-1} + p_d(1-p_f)(1-p_m)^2\sum_{k=3}^{5}(1-p_d)^{k-3} + p_m(1-p_f)(1-p_d)^3 \cdot \sum_{k=6}^{7}(1-p_m)^{k-4}.$$

If Eq. (5) is combined with Eq. (8), it is written as

$$\alpha_{i,j,0} = \frac{W_i - j}{W_i}\left(p_f + B\right)^i \alpha_{0,0,0}\,. \tag{9}$$

If Eq. (5) is combined with Eq. (8), it can also be written as

$$\alpha_{i,0,k} = \begin{cases} (1-p_f)(1-p_m)^{k-1}(p_f+B)^i\alpha_{0,0,0}, & 1 \le k \le 3 \\ (1-p_f)(1-p_m)^2(1-p_d)^{k-3}(p_f+B)^i\alpha_{0,0,0}, & 4 \le k \le 6 \\ (1-p_f)(1-p_m)^3(1-p_d)^3(p_f+B)^i\alpha_{0,0,0}, & k = 7. \end{cases} \tag{10}$$

Then, all the steady state probabilities from the above four equations, Eq. (7)~(10), should satisfy the following additional requirement – the sum of all those probabilities should be equal to one.

$$\sum_{i=0}^{r}\sum_{j=0}^{W_i-1}\sum_{k=0}^{7}\alpha_{i,j,k}\cdot I(j,k)=1 \qquad \text{where } I(j,k)=\begin{cases}0, & j\neq 0 \,\&\, k\neq 0\\ 1, & \text{otherwise.}\end{cases}$$

In addition to the above requirement, the CRTS frame transmission probabilty at a source node τ and the CRTS frame collision probability p_c should satisfy the following conditions, respectively.

$$\tau=\sum_{i=0}^{r}\alpha_{i,0,0}=\frac{1-\left(p_f+B\right)^{r+1}}{1-(p_f+B)}\cdot\alpha_{0,0,0} \tag{11}$$

$$p_c=1-(1-\tau)^{K-1} \tag{12}$$

where K is the number of source nodes.

For performance comparion, two performance measures are considered. One is the system throughput in bps defined as the ratio of the total amount of data transmitted successfully to the total consumed time period. The other performance measure is the average channel access delay in seconds. This channel access delay is the average waiting time of any source node for one data frame transmission, measured from the first channel contention time with the CRTS frame to the time instant that the source node receives the ACK2 frame from the destination node. In order to derive the numerical formula for the above-mentioned two performance measures, two types of average delays should be calculated in advance. The first one is the average delay until successful transmission D_S, which is the time between the ARTS frame transmission and the successful ACK2 frame reception, and the second one is the average delay until failure transmission D_E, which is the time between the ARTS frame and the failure of any nearby frame transmission. The average delay D_S can easily be derived as

$$D_S=T_{ARTS}+T_{CCTS}+3\times T_{DATA}+2\times T_{ACK}+6\times SIFS. \tag{13}$$

Derivation of the second average delay D_E is a little complex. In order to derive it, one more parameter D_E^i must be defined. This is the delay from the time the ARTS frame is sent to the instant that the frame transmission error takes place in the *i*-th transmission phase. Seven different average delays are then given by

$$D_E^1=T_{ARTS}+SIFS, \qquad D_E^2=D_E^1+T_{CCTS}+SIFS, \qquad D_E^3=D_E^2+T_{DATA1}+SIFS,$$
$$D_E^4=D_E^3+T_{DATA2}+SIFS, \qquad D_E^5=D_E^4+T_{DATA1\oplus 2}+SIFS, \qquad D_E^6=D_E^5+T_{ACK}+SIFS,$$
$$D_E^7=D_E^6+T_{ACK}.$$

Then, the average delay from the time the ARTS frame is sent to transmission failure can be derived as

$$\begin{aligned}D_E=\{&D_E^1p_m+D_E^2(1-p_m)p_m+D_E^3(1-p_m)^2p_d+D_E^4(1-p_m)^2(1-p_d)p_d\\&+D_E^5(1-p_m)^2(1-p_d)^2p_d+D_E^6(1-p_m)^2(1-p_d)^3p_m+D_E^7(1-p_m)^3(1-p_d)^3\}/P_n,\end{aligned}$$

where P_n is a probability used for a normalization to one and is given by

$$P_n=p_m+(1-p_m)p_m+(1-p_m)^2p_d+(1-p_m)^2(1-p_d)p_d+(1-p_m)^2(1-p_d)^2p_d$$

$$+(1-p_m)^2(1-p_d)^3 p_m + (1-p_m)^3(1-p_d)^3.$$

The probability P_t that there is at least one CRTS frame transmitted by *K* source nodes in a designated time duration and the probability P_s that the transmitted CRTS frame is successfully received by the helper node without any collision and transmission error are given by

$$P_t = 1-(1-\tau)^K \tag{14}$$
$$P_s = \frac{K\tau(1-\tau)^{K-1}}{P_t}(1-p_m). \tag{15}$$

The probability P_a that no transmission errors occur during the period from ARTS frame transmission to ACK2 frame reception is given by

$$P_a = (1-p_m)^4(1-p_d)^3. \tag{16}$$

Let's define the consumed time slot as the time interval between two consecutive backoff time counter decrements. Then, according to this NC-MAC protcol, there are four different types of consumed time slots, shown in **Fig. 9**.

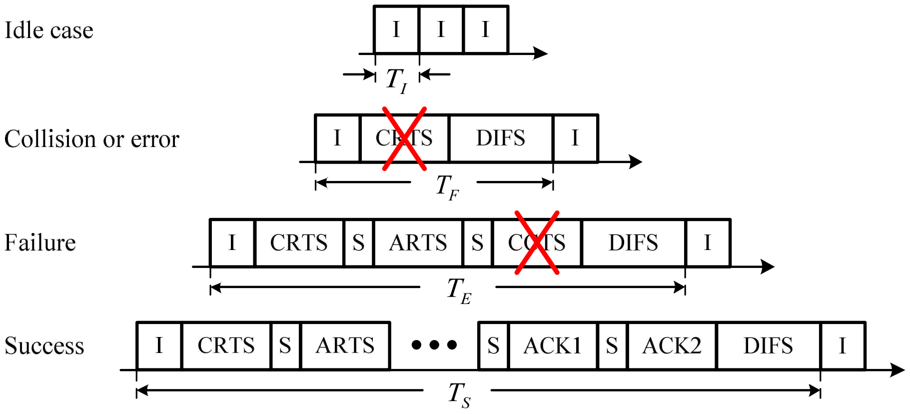


Fig. 9. Four different types of consumed time slots

First, when there is no transmission on the wireless channel, the consumed time slot means that the slot duration $T_I = \sigma$. Second, when the transmission of the CRTS frame results in failure due to collision or a bad wireless channel, the consumed time slot becomes T_F. Third, when the source node does not receive the ACK2 frame from the destination node even after the successful transmission of the CRTS frame, the consumed time slot becomes T_E. Lastly, if the total transmission scenario is successful, then this consumed time slot becomes T_S. All of them are indicated as

$$T_I = \sigma$$
$$T_F = T_{CRTS} + DIFS + \sigma$$
$$T_E = T_{CRTS} + SIFS + D_E + DIFS + \sigma$$
$$T_S = T_{CRTS} + SIFS + D_S + DIFS + \sigma.$$

Then, average size of the consumed time slots in seconds can be given by

$$\mathrm{E[slot]} = (1 - \mathrm{P_t})\mathrm{T_I} + \mathrm{P_s P_t P_a T_S} + \mathrm{P_t P_s}(1 - \mathrm{P_a})\mathrm{T_E} + \mathrm{P_t}(1 - \mathrm{P_s})\mathrm{T_F}. \tag{17}$$

The system throughput can be calculated as the ratio of the DATA frame transmission time to the average size of the consumed time slots and is given by

$$\mathrm{S} = \frac{\mathrm{P_s P_t P_a}(\mathrm{L_{DATA1}} + \mathrm{L_{DATA2}})}{\mathrm{E[slot]}}\ [\mathrm{bps}]. \tag{18}$$

Now we move on to the derivation of the other performance measure, average channel access delay. In order to obtain this, we should have the probability $\mathrm{P(i)}$ that one source node succeeds in receiving the ACK2 frame at the *i*-th backoff stage. For the sake of simplicity, it is assumed this probability is approximately modeled as a geometric distribution with a parameter $\mathrm{p_{st}}$, the successful transmission probabiltiy from the CRTS frame to the ACK2 frame. It is given by

$$\mathrm{p_{st}} = (1 - \mathrm{p_f})\mathrm{P_a}. \tag{19}$$

Then, the probability $\mathrm{P(i)}$ is given by

$$\mathrm{P(i)} = (1 - \mathrm{p_{st}})^{\mathrm{i}}\mathrm{p_{st}}, \quad 0 \le \mathrm{i}. \tag{20}$$

Finally, the average channel access delay is given by

$$\mathrm{E[D]} = \begin{cases} \displaystyle\sum_{\mathrm{i=0}}^{\mathrm{r}} \left\{ \mathrm{E[slot]} \frac{\mathrm{W_0}}{2} \sum_{\mathrm{j=0}}^{\mathrm{i}} 2^{\mathrm{j}} + \mathrm{T_S} - \mathrm{DIFS} \right\} \cdot \mathrm{P(i)}, & \mathrm{r} \le \mathrm{m} \\ \displaystyle\sum_{\mathrm{i=0}}^{\mathrm{m}} \left\{ \mathrm{E[slot]} \frac{\mathrm{W_0}}{2} \sum_{\mathrm{j=0}}^{\mathrm{i}} 2^{\mathrm{j}} + \mathrm{T_S} - \mathrm{DIFS} \right\} \cdot \mathrm{P(i)} & \\ \quad + \displaystyle\sum_{\mathrm{i=m+1}}^{\mathrm{r}} \left[\mathrm{E[slot]} \frac{\mathrm{W_0}}{2} \left\{ \sum_{\mathrm{j=0}}^{\mathrm{m}} 2^{\mathrm{j}} + \sum_{\mathrm{j=m+1}}^{\mathrm{i}} 2^{\mathrm{m}} \right\} + \mathrm{T_S} - \mathrm{DIFS} \right] \cdot \mathrm{P(i)}, & \mathrm{r} > m. \end{cases} \tag{21}$$

5. Numerical Results

It is assumed there are K source nodes (more exactly, K communication groups) in the communication network considered for performance evaluation. Each communication group consists of three nodes, i.e., a source node, a destination node, and a helper node. The system parameters used in the performance evaluation are described in **Table 1**. It is also assumed that all control frames are transmitted at the basic rate (i.e., 1 Mbps) and DATA frames are transmitted at different transmission rates based on the received SNR value. However, it is assumed here that the direct path between the source node and the destination node provides only the basic rate, and that the transmission paths between the helper nodes and the source or destination nodes provide 11 Mbps. In the case of the DCF protocol, since the source node does not use a cooperative mechanism, this node exchanges its frames with the destination directly. It is also assumed for simple calculation that two different types of frame transmission errors, p_m for control frames and p_d for DATA frames, are the same. In order to evaluate the performance of this NC-MAC protocol and verify the numerical results of the mathematical analysis from Section 4, computer simulations have been carried out. The simulation code was programmed using a C++ language based on the SMPL library [16].

Table 1. System parameters

Parameters	**Values**	**Parameters**	**Values**
CRTS length	352 bits	SIFS	10 μs
ARTS length	352 bits	DIFS	50 μs
CCTS length	304 bits	CW_{min}	32
ACK length	204 bits	CW_{max}	1024
DATA length	1024 bytes	R_{sh} , R_{hd}	11 Mbps
Slot size (σ)	20 μs	Basic rate	1 Mbps
K	variable	p_m (p_d)	variable
Simulation time	1000 sec	R_{max}	11 Mbps

Fig. 10 and **Fig. 11** show performance comparisons between mathematical analysis and computer simulation results. Analytic results are depicted by various lines and computer simulation results are depicted by various points in each figure. The analytic results of system throughput performance of the NC-MAC protocol are compared with the simulation results in **Fig. 10** under various frame transmission error probabilities and traffic loads, i.e., number of source nodes.

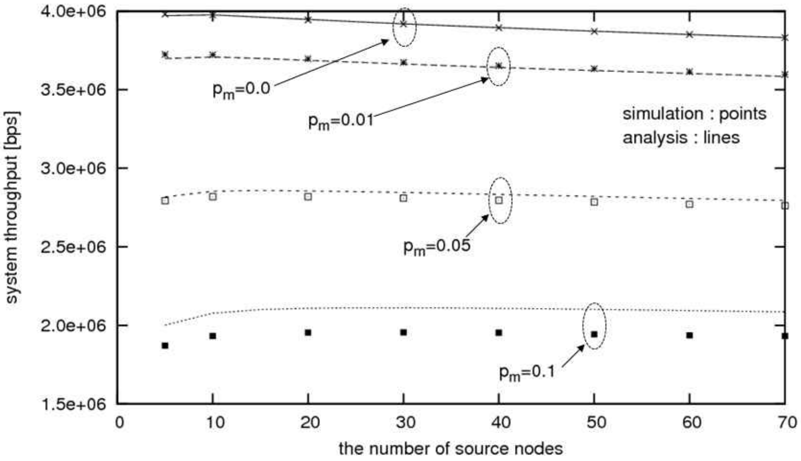


Fig. 10. System throughput comparison of analysis and simulation results

According to the system throughput results in **Fig. 10**, system throughput tends to increase slightly until the number of source nodes becomes 10. However, as the number of the source nodes exceeds 10, the system throughput for this protocol decreases. This is because the saturated traffic model is used in this performance evaluation and approximately 10 numbers of the source nodes are the maximum value this wireless channel can accommodate. Therefore, as the number of source nodes exceeds that value, the number of discarded DATA frames increases abruptly due to consecutive collisions of CRTS frames; thus, system throughput tends to decrease. When frame transmission error probability is small, the gap between analytic and simulation results is negligible. However, it is shown that the difference between analytic and simulation results tends to increase when $p_m = 0.1$, but is still small because the gap in system throughput is only about 6%. The reason for this difference is still under investigation, but it might be due to the following three factors: the first one is a slight difference in retransmission scheme between analysis and simulation, i.e., the value of the retransmission timer, the second one is the assumption of geometric distribution in Eq. (20), and the last one is that the transmission success and failure of ACK2 frames at $b(t) = r$ are not applied separately in deriving steady-state probabilities.

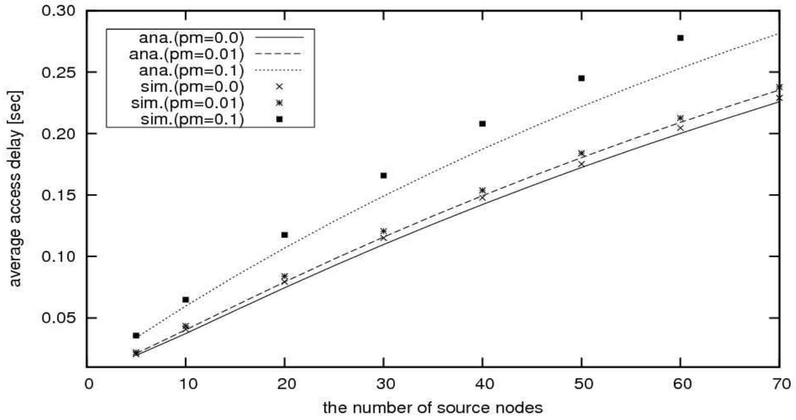


Fig. 11. Average access delay comparison of analysis and simulation results

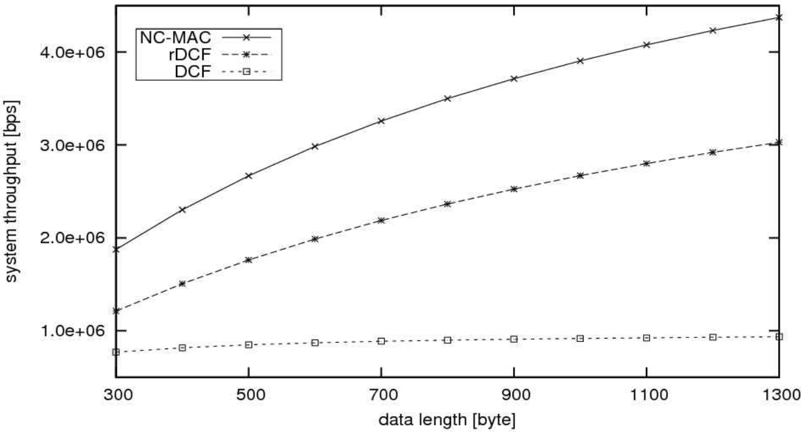


Fig. 12. System throughput comparison for different schemes

Performance comparisons of the NC-MAC protocol with the DCF and the rDCF protocol under variations of different system parameters, the data length, and the number of source nodes, are depicted in **Fig. 12** and **Fig. 13**. Those values are derived from mathematical analysis when frame transmission errors for data and control frames are both zero. **Fig. 12** shows system throughput results of the NC-MAC protocol compared with the DCF and rDCF protocols as a function of the DATA frame length when the number of source nodes is 10. System throughput performance increases as the DATA frame length increases. This is because the ratio of pure payload to overhead, such as frame header and control frame, increases as the DATA frame length increases. In **Fig. 13**, the system throughput performance of the NC-MAC protocol is compared with those of the DCF and rDCF protocols as a function of the number of source nodes. According to this result, the system throughput of the NC-MAC protocol increases by about 47% more than that of the rDCF protocol. This gain is mainly caused by network coding. Theoretically, the network coding gain for this scenario is 4/3, but the throughput gain obtained in these numerical results is greater than 4/3 (about 4/2.7). This difference is caused by the reduced overhead with control frames. For instance, the rDCF protocol requires twice the number of control frame exchanges, once for each direction. However, the NC-MAC protocol needs to exchange control frames only once for bidirectional communication.

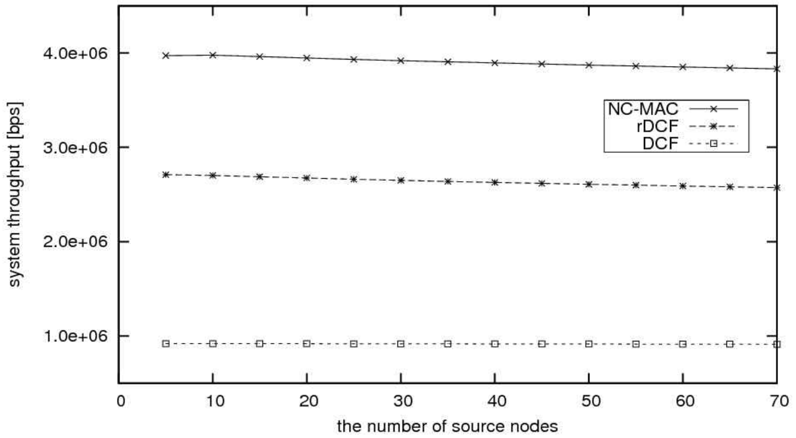


Fig. 13. System throughput comparison for different schemes

Fig. 14 and **Fig. 15** show how contention window size affects system throughput and average channel access delay under various traffic load circumstances where frame transmission error probabilities are set as $p_m = 0.001$. The similar consequences can occur using the conventional DCF protocol. According to these two figures, increased contention window size tends to provide better system performance in terms of system throughput, but worse performance in terms of average channel access delay.

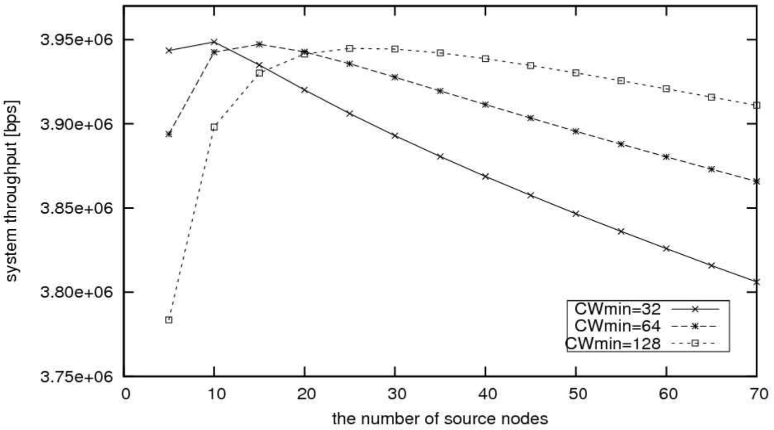


Fig. 14. System throughput comparison for various contention windows

6. Conclusion

In this paper, the performance of a cooperative MAC protocol named NC-MAC is evaluated. This protocol introduces an approach to combine cooperative communication and network coding for ad hoc networks in order to significantly increase system performance. This paper is an extended version of our previous research [12]. There are two key differences. One is that we adopt two different methods for performance evaluation of the NC-MAC protocol, mathematical analysis and computer simulation. The other is that it's the performance evaluation is carried out using an erroneous wireless channel for the source node, helper node, and the destination node. That is to say, all the frames exchanged are susceptible to transmission errors, which is a new circumstance differentiating this research from all previous research. According to the numerical results, the NC-MAC protocol provides significantly better system throughput than the DCF scheme in the IEEE 802.11 standard and better system throughput by about 47% compared to the rDCF scheme.

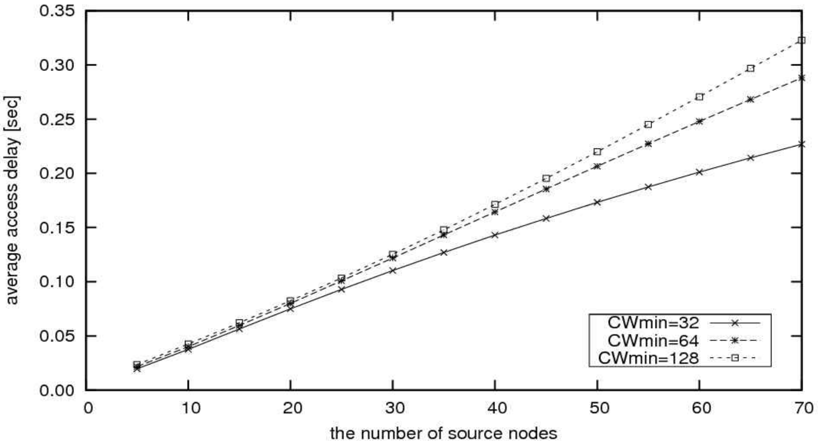


Fig. 15. Average access delay comparison for various contention windows

References

[1] IETF MANET Working Group. Available from: <http://datatracker.ietf.org/wg/manet/>. Article(CrossRef Link)

[2] IEEE Std 802.11-1999. *Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, Jun.1999. Article(CrossRef Link)

[3] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine,* vol.42, no.10, pp.74-89, Oct.2004. Article(CrossRef Link)

[4] S. Katti, H. Rahul, D. Katabi, W. H. M. Medard, and J. Crowcroft, "XORs in the Air: Practical wireless network coding," in *Proc. of ACM SIGCOMM*, pp.243-254, Sep.2006. Article(CrossRef Link)

[5] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol.18, no.3, pp.535-547, Mar.2000. Article(CrossRef Link)

[6] J. W. Tantra, C. H. Foh, and A. B. Mnaouer, "Throughput and delay analysis of the IEEE 802.11e EDCA saturation," in *Proc. of IEEE ICC*, pp.3450-3454, May.2005. Article(CrossRef Link)

[7] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. of ACM/IEEE MOBICOM*, pp.236-251, Jul.2001. Article(CrossRef Link)

[8] Sai Shankar N, Chun-Ting Chou, and Monisha Ghosh, "Cooperative communication MAC

(CMAC) - A new MAC protocol for next generation wireless LANs," in *Proc. of* IEEE International Conference on Wireless Networks, Communications and Mobile Computing, pp.1-6, Jun.2005. Article(CrossRef Link)

[9] H. Zhu and G. Cao, "rDCF: A relay-enabled medium access control protocol for wireless Ad Hoc Networks," IEEE Transactions on Mobile Computing, vol.5, no.9, pp.1201-1214, Sep.2006. Article(CrossRef Link)

[10] P. Liu, Z. Tau, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: A cooperative MAC for wireless LANs," IEEE Journals on Selected Areas in Communications, vol.25, no.2, pp.340-353, Feb.2007. Article(CrossRef Link)

[11] J. Jang and S. W. Kim, "Performance evaluation of the reliable cooperative MAC in Ad Hoc networks," in *Proc. of IEEE ICACT*, pp.18-23, Feb.2011. Article(CrossRef Link)

[12] Chun-Hung Liu and Feng Xue, "Network Coding for Two-Way Relaying: Rate Region, Sum Rate and Opportunistic Scheduling," in *Proc. of* IEEE ICC, pp.1044-1049, May.2008. Article(CrossRef Link)

[13] Antonios Argyriou, "Wireless Network Coding with Improved Opportunistic Listening," *IEEE Transactions on Wireless Communications*, vol.8, no.4, pp.2014-2023, Apr.2009. Article(CrossRef Link)

[14] S. Sengupta et al., "An analysis of wireless network coding for unicast sessions: The case of coding-aware rotuing," in *Proc. IEEE* INFOCOM, pp.1028-1036, May.2007. Article(CrossRef Link)

[15] J. Jang and S. W. Kim, "Network coding-enabled cooperative MAC protocol in Ad Hoc networks," in *Proc. of IEEE ICOIN*, Jan.2010. Article(CrossRef Link)

[16] M. H. MacDougall, "Simulating computer systems: techniques and tools", The MIT Press, Cambridge, 1992. Article(CrossRef Link)

Jaeshin Jang received the B.S. degree in Electrical Engineering from Dong-a University, Korea, in 1990, and the M.S. and Ph.D. degrees in Electric and Electrical Engineering from KAIST, Korea, in 1992 and 1998, respectively. From July 1997 to February 2002, he worked for Samsung Electronics Company. From August 2008 to July 2009, he was a visiting scholar at Iowa State University. Since March 2002, he has worked for Inje University, Korea. His major interests are wireless QoS, MAC, routing, and cooperative communications at wireless communications networks, including mobile WiMAX, ad-hoc networks, and mesh networks.

Sang Wu Kim received the B.S. degree from Yonsei University, Seoul, Korea, in 1981, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1983, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 1987, all in electrical engineering. He is currently an Associate Professor in the Iowa State University, Ames. Earlier, he held teaching and research positions at KAIST, California Institute of Technology, and AT \& T Labs-Research. His research interests are primarily in the areas of wireless communications with emphasis in code-division multiple access (CDMA), error-correction coding, MIMO, space-time coding, multiuser detection, adaptive modulation and coding, and cooperative communications. Dr. Kim is an Associate Editor of IEEE COMMUNICATIONS LETTERS and the Journal of Communications and Networks. He was a recipient of the Fulbright Graduate Study Award, U.S. Department of State, in 1983, and the best paper award at the IEEE International Symposium on Spread Spectrum Techniques and Applications, September 2000. In March 2003, he was listed in the Marquis *Who's Who in Science and Engineering*.

Sunghong Wie received the B.S., M.S. and Ph.D. degrees in electrical engineering from KAIST, South Korea, in 1995, 1997, and 2001, respectively. From 2001 to 2008, he was with the Samsung Electronics as a senior engineer. Currently, he is with Division of Information Technology at The Cyber University of Korea. His recent research interests include mobility management and future internet.