2007 International Conference on Convergence Information Technology

An Analysis of Reliable MAC Layer Multicast in Wireless Networks

Yoodoc Song, Junho Chung, Wookyung Sung, Bosung Kim, Dowon Hyun and Juwook Jang
Department of Electronic Engineering, Sogang University.
Email: { yoodoc, mywingcat, babonorae, snugness, snatcher }@eeca1.sogang.ac.kr,
jjang@sogang.ac.kr

Abstract

In this paper, we compare and investigate BMW and BMMM proposed as multicast protocol in IEEE 802.11. M.T.Sun et al. analyzed wireless multicast protocol and they proposed BMMM protocol more efficient than BMW. But efficiency of BMW had not been evaluated clearly in paper BMMM, so efficiency of BMMM is better than BMMM. However, we analyze BMW and BMMM properly and demonstrate that the transmission time of BMW is shorter than that of BMMM through simulation.

1. Introduction

There has been research for a reliable and efficient multicast protocol in IEEE 802.11 wireless network. The IEEE 802.11 multicast protocol is based on CSMA/CA procedure. When ACK is dropped, this protocol treats the transmission is failed and retransmits the packet. It remains research problems, such as hidden node problem and ACK collision problem. To solve these problems and to do better efficient multicast/broadcast, BMW[2], BMMM[1], and others are proposed.

M. T. Sun et al. proposed BMMM that has fewer contention phases and efficiency than BMW. Furthermore, they demonstrated performance about it. But performance of BMW was not demonstrated well in [1].

They defined BMW protocol by contrast with [2], [3], and then demonstrated efficiency and reliability of BMMM.

First of all, [1] assumed the number of nodes in the situation that only one data exists. In case of BMW, when one data exists, then the number of contention phases is n –the number of nodes- and it says that n contention phases/a data. But in the environment of actual transmission, it has many cases of the that transmit consecutive data, and the number of contention phases in BMW is to be n as n nodes. So it means that the number of contention phases decreases from n to one per each data.

Secondly, [1] presents the node which sends and receives RTS/CTS, has a contention phase when transmission fails to the node. In the case of actual BMW, however, as shown Figure 2, only one node among n nodes has one contention phase to one data. Due to these two differences of analysis, [1] could not properly analyze performance of BMW. And the comparison of performance between BMW and BMMM has point at issue. In this paper, following the first paper of BMW proposal as well as the paper of BMW, we reconsider BMW protocol and do not consider the situation transmitting one data, but the situation transmitting a number of data consecutively. Therefore, we analyze performance of BMW protocol anew by the numerical formula.

2. Basic concepts of BMMM and BMW

2.1. BMMM

BMMM(Batch Mode Multicast MAC)[1] is a protocol which supports to reliable multicast in IEEE802.11 and that is proposed by M. T. Sun et al.

BMMM(Figure 1b) basically uses RTS/CTS frame couple of n and RAK(Request for ACK)/ACK frame couple of n when it transmits one data to n nodes. To transmit one data, sender sends RTS first and receives CTS from each node, ready to receive mode for all nodes and transmit data.

In other words, if number of 1 … n received nodes exist, then sender transmits RTS to node 1 and receives CTS, so sender promises to ready to node 1 and exchange RTS/CTS message to 2, 3, ... n node sequentially. After all, senders confirm to all nodes that plan to transmit data and ready. When it finishes RTS/CTS message exchange, sender transmits data to all nodes and exchange RAK/ACK to all nodes that certified result of data transmission for all received nodes.

As above, a merit of BMMM is to reduce delay which takes to receive first data. But BMMM has a defect that control overhead caused by RTS/CTS, RAK/ACK couple of n is increased.

"This study was supported by the Seoul Research and Business Development Program, Seoul, Korea"

0-7695-3038-9/07 $25.00 © 2007 IEEE
DOI 10.1109/ICCIT.2007.353

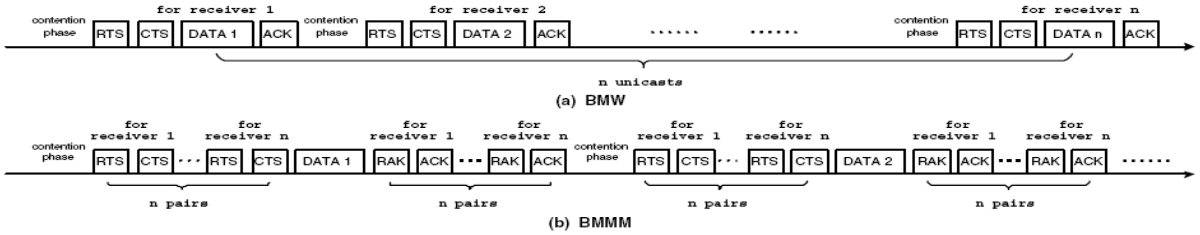

Figure 1. BMW and BMMM

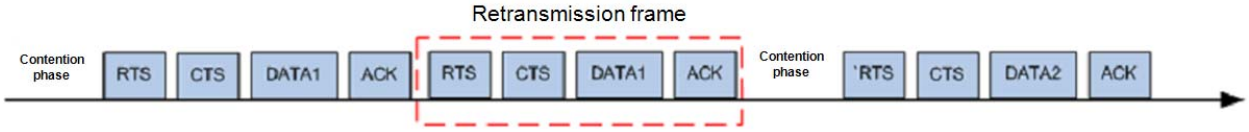

Figure 2. Retransmission frame structure in BMW

2.2. BMW

BMW(Broadcast Medium Window) protocol is the algorithm that Tang and Gerla proposed, for raising the reliability of broadcast service in IEEE 802.11 [2]. BMW(Figure 1a) was composed of RTS/CTS/DATA/ ACK. It broadcasts reliably as unicasting to each 1-hop neighbor. As sender unicasts data to neighbor node and other nodes overhear that data frame, throughput is increased. If data transmission succeeds, receiving nodes send ACK. After source node increase the number of sequence, it transmits the next data. If neighbor nodes fail to receive data frame, they send RTS with the sequence number of missing data frame.

[1] considers only one data ignoring total nodes and retransmission. So there is one contention phase(Figure 2) and embodiment is simple. [1] refers that there is large delay in transmitting one data. For example, as there is n nodes and n-th node fail to receive first data, n-th node waits for communicating from first node to n-1th node. This delay is considered by only one data. But in real system, multiple data is transferred. Namely, n-th node waits missing data with receiving other data. So comparing total data transmission, one delay is ignorable(Figure 3)

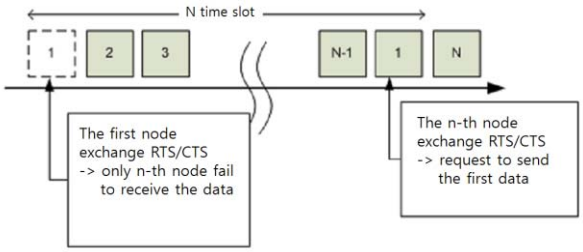

Figure 3. Illustration of message reception of the n-th node

Another weak point of BMW is that it must have n-1 contention phase in order to confirm a transmission completion. But in this case, although the data increases when the data is transmitted consecutively, necessary contention phase were fixed on n-1. So overhead to the number of data diminishes like the Figure 4.

3. Compare performance a BMW with BMMM

3.1. Average number of contention phases

When wanting a data transfer, contention phase confirms the condition of medium. Source node transmits the frame when medium is free. If medium is busy, source node has random back-off time slot until medium is free. And after back-off time, source node transmits frame when back-off timer is expired. If channel is busy during back-off time, back-off timer stops and confirms the condition of channel.

[1] assumes only one data in wireless communication. So BMW protocol regards that the number of contention phase is n (the number of nodes). Namely, n contention phases per a data.

But in real wireless communication environment, it is extended in multiple time slots where contain many data. 1 cycle is defined as RTS/CTS exchange time between source node and others. In 1 cycle, there is a contention phase at each node (Figure 2). Namely source node transmits n data to n nodes. So the number of contention phase is 1 contention phase per a node. And if the number of data is larger than the number of nodes, the number of contention phase is 1 contention phase per a data.

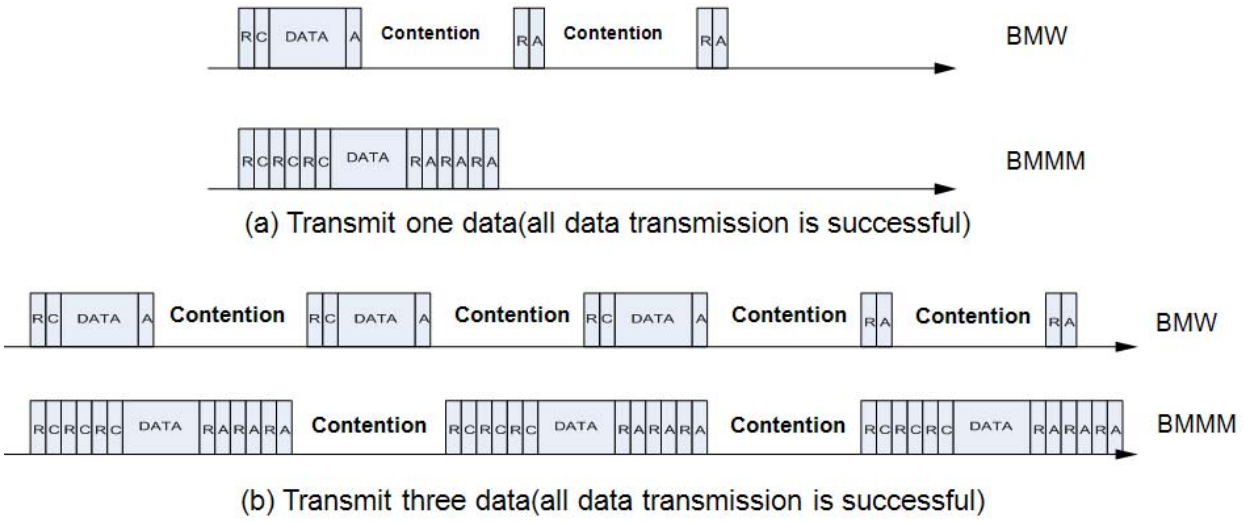

Figure 4. Consecutive transmission of data in BMW and BMMM

When we compare the number of average contention phase of BMW with that of BMMM, BMW puts out efficiently. Also BMMM has RTS/CTS as the number of nodes in each time slot. That became overhead against all data.

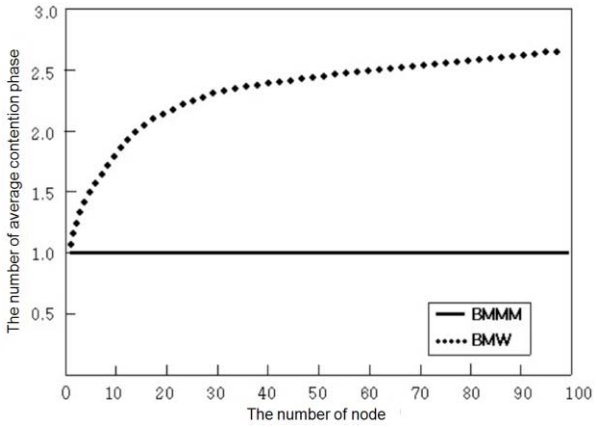

Figure 5. Comparison the average number of contention phase per one data in BMW and BMMM ($p = 0.9$)

3.2. Average data transmission time

[1] considered the situation that transmits one data to whole n nodes. But this assumption does not consider actual communication condition. To analyze performance of protocol in actual communication, we do not assume that only one data frame exists but that multiple data frame exist.

[1] considered the situation that one data frame exists with actual communication situation differently.

In this case, BMW must send and receive RTS/CTS with receiving nodes to complete communication even case one data frame exists. Therefore, when there are a few sequential data and many nodes, it caused bad performance by contention phase. Because the BMMM protocol exchanges RTS/CTS with all nodes in each data transmission, it completes communication that one data exists in one time slot. Like this specific situation, BMMM shows good performance. But actual multicast communication environment is not case send one data frame at each communication, it considers the environment that multiple data frame created and analyzes protocol efficiency. It compares average data transmission time between two protocols to analyze protocol efficiency. Average data transmission time is defined as the average value of time that one node receives all n data when n receiving nodes exist. At this time, it is one cycle processing until n receiving nodes receive all n data and consider average data transmission time in one cycle situation.

3.2.1. Average data transmission time of BMW. The most important factor may be overhead by retransmission in average data transmission time. To analyze average data transmission time of BMW, all receiving nodes will consider one cycle time that exchanges RTS/CTS each time. Considering average data retransmission time, there is initial data frame transmission with contention phase in each time slot and retransmission data frame without contention phase. Initial data frame transmission that has a feature each one node is taking charge of each data frame except contention phase. Therefore, the number of initial data frame transmission is one time independent of the total number of receiving nodes while all receiving nodes

exchange one RTS/CTS. The case of retransmission should consider n data retransmission about each node in one cycle, and effect the total number of receiving node. When only one receiving node exists and successful probability of data transmission is p, to retransmit one data frame, it must have the number of $1+(1-p)+(1-p)^2+(1-p)^3+(1-p)^4=1/p$ retransmission time. It is consumed one time (contention phase + RTS + CTS + data + ACK) about one of characteristically motivated time that is initial data frame transmission, and each time (RTS + CTS + data + ACK) without contention phase. Hence, it follows next equation about average data transmission time.

$$T_d(1)=T_{CP}+T_{frame}+(1-p)T_{frame}1/p \quad (1)$$

In case that only two receiving nodes exist, the time until one node receives two data completely has one contention phase. It is because the sender has contention phase about each node in situation two receiving nodes exist prior to initial CTS transmission, and transmit CTS without contention phase about retransmission data frame of each node. It considers the number of average transmission while one receiving node is one cycle in situation existing two receiving nodes. (The process all data transmit in situation existing two receiving node) This value constituted three partitions.

First, it considers the case only first data transfer is initial data transfer. The second data from the second node become initial data, so initial transmission of the second data from the first node becomes receivable by overhear, but considering transmission time in the first node, it is excepted. Next, it considers the case retransmit to one of two data from one of two nodes. When the first receiving node does not receive first data or initial transmission of second data, this form is probability that does not receive one data $(1-p)$ and $(1-p)_2C_1$ multiplied ${}_2C_1$. In this case, the number of average transmission that repeats retransmission and receives all data is $1/p$. Next, the case that first node could not receive whole twice initial data in two data frame is probability of ${}_2C_2(1-p)^2$, and the number of average transmission in one node for transmitting two retransmission data completely is $2/p$. The number of average data transmission in one node is form of f_2 adding number of average retransmission to these three probabilities.

$$f_2=1+{}_2C_1p(1-p)1/p+{}_2C_1(1-p)^2 2/p \quad (2)$$

To compute average data transmission time using the number of average retransmission, it multiplies data frame time having contention phase in initial data transmission 1, and is not contention phase by feature of BMW protocol in retransmission. It differs the value multiplied to retransmission portion and initial transmission portion in the number of average retransmission, computing average data transmission time,

$$T_d(2)=T_{CP}+T_{frame}+\sum_{r=1}^{2}C(2,r)p^{2-r}(1-p)^rT_{frame}(r/p) \quad (3)$$

Like this method, average data transmission time is following equation 4.

$$T_d(N)=T_{CP}+T_{frame}+\sum_{r=1}^{N}{}_NC_rp^{N-r}(1-p)^rT_{frame}(r/p) \quad (4)$$

From here, r is the number of data that must re-transmit in timing that one node exchange RTS/CTS itself, p is probability that success data transmission in that timing. n is total number of receiving nodes and considering data in 1 cycle, T_{CP} is time of contention phase, T_{frame} is time of RTS + CTS + data + ACK.

In above equation, the time of $T_{CP}+T_{frame}$ is processing time of one initial data frame transmission in each nodes, $\sum_{r=1}^{N}{}_NC_rp^{N-r}(1-p)^r$ is probability of retransmission. $p^{N-r}(1-p)^r$ is probability that cannot receive r data, considering combination of this from, it multiplies the number of retransmission 1/p that until receiving one data, this value is $\sum_{r=1}^{N}{}_NC_rp^{N-r}(1-p)^r$ and total number of average transmission. It considers only T_{frame} time without contention phase about this retransmission data frame. Because all assumption in equation 4 existing n receiving node and transmitting n data, it multiplies n to average data transmission time, so we can compute data transmission $nT_d(n)$ about n nodes.

$$T_d(N)=n(T_{CP}+T_{frame}+\sum_{r=1}^{N}{}_NC_rp^{N-r}(1-p)^rT_{frame}(r/p)) \quad (5)$$

As mentioned above equation, p is the probability of receiving data successfully and it is changed by one cycle of the number of each node.

For example, in one cycle of 4 nodes and 4 data, the probability that 4th node receives the first data (the initial transmitted data in ACK timing of the first node) successfully is different from the probability it receives the 4th data (the 4th transmitted data in ACK timing of the first node) successfully. In ACK timing of oneself, the probability that 4th node requests retransmission of data 1 is same that of all retransmissions by 1st node`s cancellation transmission and 1st, 2nd, 3rd nodes fails.

On the other hand, In ACK timing of oneself, the probability that 4th node requests retransmission of data 4 is same that of the node can`t receive first transmission of 4th data in timing of oneself. Difference of the transmission probability is decided by the first transmission time, in each node. At first, in case of the data which agrees with the ACK timing of oneself, the retransmission is accomplished when the first data transmission fails. In equation 6, F_1 is the probability of retransmission and T_1 is the number of average transmission until previous timing.

$$F_1 = (1-p) \quad T_1 = 1 \tag{6}$$

When initial transmission has done one step before its own timing, if the node can′t receive the data during T_2, then the node request to retransmit the data with probability F_2. T_2 is obtained by adding the number of initial transmission 1, the number of average retransmission of former node and probability of transmission failure(E)* the number of retransmission.

$$T_2 = 1 + E\frac{1}{1-E} \tag{7}$$

$$F_2 = E^{(1+E(1/(1-E))} = E^{T_2} \tag{8}$$

As initial transmission has done two steps before its own timing, the number of average transmission before ACK timing, T_3 is as follows.

$$T_2 = 1 + E\frac{1}{1-E} + E^{1+E\frac{1}{1-E}}\frac{1}{1-E} = T_2 + E^{T_2}T_2 \tag{9}$$

At this time, ACK timing gets to be F_3 equals to E^{T_3}, the probability that the node requests to retransmit this data.

When initial transmission has done n step before its own timing, we can obtain T_n and F_n. T_n is the number of average transmission before its own timing, and f F_n is the probability that n-th node request to retransmit.

$$T_n = T_{n-1} + E^{T_{n-1}}\frac{1}{1-E} \tag{10}$$

$$F_n = E^{T_n} \tag{11}$$

The result can be obtained like Figure 5 by applying equation 10, 11 to p and equation 5. We set transmission failure probability E, contention phase, DATA as 10% in [1], 50 time slot, and 5 time slot. And we set RTS, CTS, RAK, and ACK as 1 time slot value.

3.2.2. BMMM average data transmission times. The time when it transmits one data from BMMM it leads in all nodes and it induces. The method which gets BMMM average data transfer times with the method which gets BMW average data transfer times is different. About one node from in whole n node it gets the time when it receives n data from BMW methods.

But from BMMM methods there is not a possibility of getting an average data transfer time in BMW methods. After transmitting one data in all nodes, it transmits the next data from BMMM methods. Like this case one data is a possibility of getting an average data transfer time at the time when it is transmitted to all nodes.

But from BMMM methods there is not a possibility of getting an average data transfer time in BMW methods. After transmitting one data in all nodes, it transmits the next data from BMMM methods. Like this case one data is a possibility of getting an average data transfer time at the time when it is transmitted to all nodes. The time when it transmits data n in n nodes will double n at the time when with the feature which transmits a data sequentially it transmits one data in n nodes and there is a possibility which it will get. The average data transfer time the result which divides n at the time when the whole n node receives n data is the time when from BMMM it transmits one data in n nodes average data transfer time and directness it will be able to compare from BMW and. The time when from BMMM it transmits one data in n nodes about under using it induces a retransmission probability as well. Against one data it has contention phase of once from BMMM. And only one node it receives the case one data which will exist completely until, the average number of time had the value of $1+(1-p)+(1-p)^2+(1-p)^3+(1-p)^4+\cdots = 1/p$. The transfer time which it follows hereupon with lower part has a same result.

$$T_d(1) = (T_{CD} + T_{frame}(1))1/p \tag{12}$$

Transfer time of the case where 2 nodes will exist thought flaw. It will be able to think with the case three branch case where two nodes exist. First, two nodes basic the time when it has an initial data transfer, at the second beginning after transmitting from in two nodes only one node the case which passes by a retransmission process, last two nodes there is a possibility of getting a transfer time with the case which all passes by a retransmission process.

Contention phase (T_{CD}) with the data transfer phase where two data are included ($T_{frame}(2)$) it has from initial data transfer process. The second case was under occurring with $2*(1-p)^2$ probabilities and to this time as $(T_{CD}+T_{frame}(1))1/p = T_d(1)$ time additionally became disturbance. The third case is under occurring with $(1-p)^2$ probability and it has $T_d(2)$ result recurrent. When considering all cases, two nodes one data transmission the time when it completes with afterwards are the same.

When consider all cases, the time that two node transfer finish one data is same next.

$$T_d(2) = T_{CD} + T_{frame}(2) +_2 C_1 p(1-p)T_d(1) \\ +_2 C_2 p(1-p)^2 T_d(2) \quad (13)$$

In the same method, if calculates the value $T_d(n)$ when the n node exist, it considers an initial data transfer.

In next, it will be able to think and calculates divide the case which is the possibility the retransmission happening in possibilities of the node where the retransmission object becomes.

This time, in case of initial data transfer, it became $T_{CD}+T_{frame}(n)$ sum of contention phase, n RTS/ CTS/ ACK, the time $T_{frame}(n)$ that include one data. Next, when the number of retransmission node is 1, 2, 3 ... n, each case needs time, $T_d(1), T_d(2), \ldots, T_d(n-1)$, $T_d(n)$ for finishing of data transfer.

Each case it occurs with probability ${}_nC_1 p^{n-1}(1-p)$, ${}_nC_2 p^{n-2}(1-p)^2$, ${}_nC_3 p^{n-3}(1-p)^3$, $\cdots$ ${}_nC_n(1-p)^n$. It will show with numerical formula, became

$$T_d(n) = T_{CD} + T_{frame}(n) + \sum_{r=1}^{N} ({}_nC_r p^{n-r}(1-p)^r T_d(r)) \quad (14)$$

This value become average time that one data transfer to n receive node, and in case of n data transfer, $nT_d(n)$ become average time that n data transfer to n receive node because n repeat this case.

$nT_d(n)$ in BMMM and $nT_d(n)$ in BMW become average time that n data transfer to n receive node

.

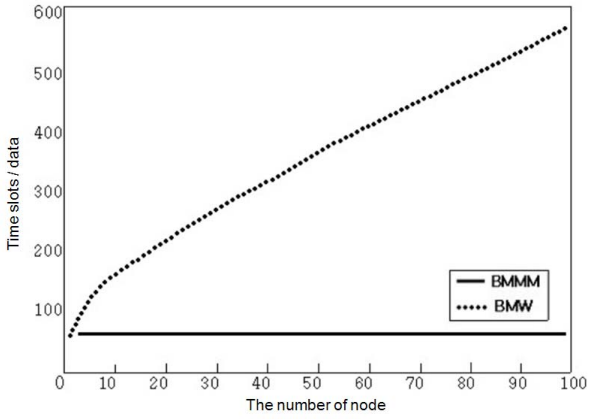

Figure 6. Time slots/data in BMW and BMMM

Figure 6 shows that the time one data is transmitted to n nodes in BMMM and BMW. We used $p = (1-E) = 0.9$ for all data in case of BMMM.

4. Performance Evaluation

Figure 7 shows average transmission time of BMW and BMMM by equation 3. It verifies Figure 3 using NS-2 simulation.

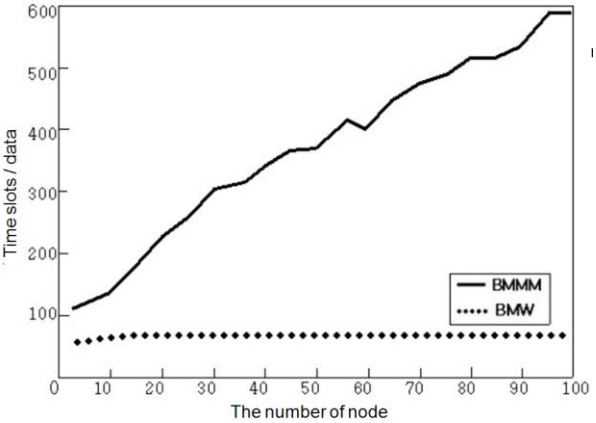

Figure 7. Average transmission time of BMW and BMMM

The number of contention phase increased as increasing of the number of nodes in BMMM. But even if the number of node increases, average data transmission time did not increase almost in BMW.

It is because width of increased transmission time that is related to the number of additional nodes does

not effect to average data transmission time greatly, but even then the number of all receiving node is increased with the result that a value of n increase, and transmission request probability F_n of data decreases in equation 11.

5. Conclusions

BMW has less contention phase numbers against a data comparing with the fact that it refers from [1]. Also it is confirmed in [2], [3].
This confronted wrong analysis of [1]. Comparing average data transferring time of BMW with that of BMMM in real wireless communication networks, BMW protocol has a possibility of knowing an efficient suitability with simple frame structure.

In case that the data is transmitted consecutively, the number of contention phases per data of BMW is nearly 1. And it shows that BMW has better performance than BMMM in average data transmission time.

We evaluated an average data transmission time related to the number of BMW and BMMM by a numerical formula and simulation. And we can also see the fact that BMW protocol is more time-efficient and it has simpler frame structure than BMMM protocol in consecutive data transmission environments.

6. References

[1] A M. T. Sun, L. Huang, A. Arora, and T. H. Lai. Reliable MAC layer multicast in IEEE 802.11 wireless networks. In *Proc. the International Conference on Parallel Processing (ICPP) 2002*.

[2] K. Tang and M. Gerla, “"Random Access MAC for Efficient Broadcast Support in *Ad Hoc Networks,”" Proc. IEEE WCNC 2000, pp. 454-459, Sep. 2000.*

[3] Min-Te Sun, Lifei Huang, Anish Arora, Ten-Hwang Lai. RMAC: A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks in *Proceedings of the 2004 International Conference on Parallel Processing (ICPP’04).*

[4] Vikram Shankar. ENHANCING THE RELIABILITY OF MEDIUM ACCESS CONTROL LEVEL WIRELESS MULTICAST A Dissertation Presented in Partial Fulfillment. of the Requirements for the Degree. Doctor of Philosophy. ARIZONA STATE UNIVERSITY.

[5] K. Tang and M.Gerla, “MAC Reliable Broadcast in Ad Hoc Networks,” Proc. IEEE MILCOM 2001, pp. 1008-1013, Oct. 2001.

[6] Editors of IEEE 802.11, Wireless LAN Medium Access Control(MAC and Physical Layer (PHY) specification, Draft Standard IEEE 802.11, 1997.

[7] K. Tang and M. Gerla, “MAC Layer Broadcast Support in 802.11 Wireless Networks,” Proc. IEEE MILCOM 2000, pp. 544-548, Oct. 2000