Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2015, Vol. 7, No. 1

Adaptive MAC Protocol for Throughput Enhancement in Multihop Wireless Networks

Nam Pham

Dept. of Computer Science, University of Nebraska-Omaha

6001 Dodge Street, Omaha, NE 68182 (U.S.A.)

E-mail: npham@unomaha.edu

Jong-Hoon Youn

Dept. of Computer Science, University of Nebraska-Omaha

6001 Dodge Street, Omaha, NE 68182 (U.S.A.)

E-mail: jyoun@unomaha.edu

Received: January 6, 2015 Accepted: April 10, 2015 Published: April 30, 2015

DOI: 10.5296/npa.v7i1.6877 URL: http://dx.doi.org/10.5296/npa.v7i1.6877

Abstract

In the paper, we propose a dynamic transmission scheduling protocol for multi-hop wireless networks by combining a fixed scheduling scheme with a simple *p*-persistent approach. In a fixed scheduling scheme, a predetermined schedule is assigned to each wireless node before transmitting data, and this schedule will be used permanently. However, in the proposed protocol, an adaptive probability-based approach is used in order to reduce the number of collisions. When a node is ready to send, it calculates the transmission probability p_{new}, using a successful transmission ratio, a previous transmission probability p_{old}, and a queue utilization ratio. The transmission probability p_{new} determines whether or not a node can transmit. Our simulation study shows that the new MAC scheduling scheme improves the network throughput and reduces the energy consumption by eliminating the number of collisions considerably.

Keywords: Dynamic Transmission Scheduling, Multihop Wireless Networks, *p*-persistent Scheduling, Time Division Multiple Access

1. Introduction

The growing demands of real-time multimedia traffic and recent advances in mobile device and wireless communication technologies are likely to drive the deployment of ubiquitous and pervasive multimedia networks for real-time applications. Furthermore, the abilities to retrieve multimedia objects with a certain level of Quality of Service (QoS) and to provide seamless, mobile communication capability are crucial in designing such networks.

In order to extend the reach of established network infrastructure, a cluster of mobile hosts without fixed base station or any wired infrastructure may form a multi-hop wireless networks, where two mobile hosts can communicate directly with each other if they are located within their radio range. If the destination is located outside of the sender's radio range, packets are relayed via intermediate hosts located between the two hosts. Each host must act as a router in the multi-hop environment.

In a multiple hop wireless networks, a Medium Access Control (MAC) protocol plays a crucial role on the performance of wireless networks such as network throughput, packet delay, and energy consumption [1-2]. Thus, in this paper, we present a new adaptive MAC protocol by combining Time Division Multiple Access (TDMA) and a *p*-persistent scheme. Although the proposed protocol use a set of fixed schedules that is pre-assigned to each node, we incorporate an advanced *p*-persistent control mechanism in a variation of TDMA to reduce the number of packet collisions. Using network simulations in *ns-2* [3], we also show that our protocol enhance the network throughput and reduce the number of collisions and energy consumption.

This paper is organized as follows. Section 2 introduces related works we base our research on. Section 3 details assumptions, parameters, and equations for the new MAC protocol. Section 4 specifies the simulation scenarios such as network topology, network traffic, and other simulation variables, and discusses the results from the simulation study. Finally, Section 5 concludes this paper.

2. Background and Related Works

In a wireless network, a node may have a fixed schedule, or it may have to contend with its neighbors to transmit on the shared channel. A fixed schedule is assigned to a node for coordinating its transmission. This section details two fixed scheduling schemes – TDMA and Modified TDMA – and then discusses some *p*-persistent approaches in the MAC layer.

In this context, a fixed scheduling scheme includes a list of nodes, and each node has its own scheduling vector. A scheduling vector is a binary vector with length *n* with the following format:

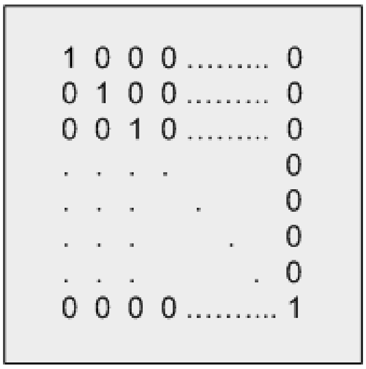
$$s_1 \quad s_2 \quad s_3 \quad \ldots\ s_i \quad \ldots\ s_n \qquad for\ 1 < i < n, \tag{1}$$

where s_i is 1 for data transmission in the i^{th} slot, or 0 for otherwise.

In practical experiments, the run time always exceeds length *n*; thus, the same scheduling vector is repeated after every s_n. The goal of the proposed scheme is maximizing the throughput by effectively and dynamically allocating the 1's and 0's in the vector.

2.1 Time Division Multiple Access (TDMA)

Time division multiple access is one of the most widely known MAC protocols because of its simplicity and effectiveness. The shared medium is time-slotted, and each node may access the shared medium according to the pre-assigned time slot. The cycle repeats when all nodes already have a chance to transmit. The typical TDMA scheduling vectors for *n* nodes are shown in Fig. 1. TDMA allows only one node to transmit at a time. The scheme ensures no collisions, but it suffers from low throughput in a multi-hop wireless environment.



```
1 0 0 0 ......... 0
0 1 0 0 ......... 0
0 0 1 0 ......... 0
. . . .           0
. . .   .         0
. . .       .     0
. . .         .   0
0 0 0 0 .......... 1
```

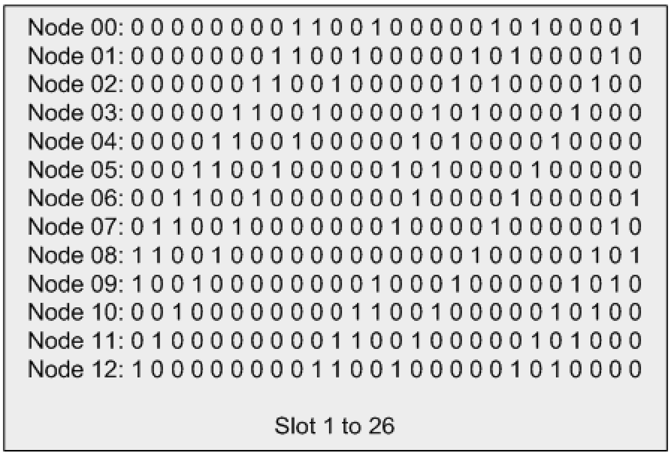
Figure 1. Sample TDMA Schedule - *n* Nodes–*n* Time Slots

2.2 Modified TDMA (MTDMA)

In [4-6], a set of different TDMA-based scheduling approaches was introduced to improve the network throughput by assigning more transmission slots in multi-hop wireless network. For simplicity's sake, we call this approach Modified TDMA or MTDMA. This approach can be summarized as follows.

Given the number of wireless nodes (*N*) and the maximum degree (D_{max}) – maximum number of adjacent nodes for a particular node − in the network, the MTDMA algorithm will output a scheduling vector using constant weight codes [4]. This method can achieve higher throughput than traditional TDMA, Galios Radio Network Design (GRAND) [5], and Ju and Li's schemes [6]. Fig. 2 shows a sample MTDMA schedule. Nodes are allowed to transmit more often, and thus the throughput can be increased.

MTDMA improves the network throughput, yet it opens doors to more data collisions. For example, in Fig. 2, both *Node 8* and *Node 9* have 1's at time slot 1'. If they are neighbors, and if they both have data to send at the time slot 1, they will transmit at the same time, which will cause a collision. If a collision occurs, the energy is wasted. Such occurrences are generally unattractive in networks where energy is limited.



```
Node 00: 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1
Node 01: 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0
Node 02: 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0
Node 03: 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0
Node 04: 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0
Node 05: 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0
Node 06: 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1
Node 07: 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0
Node 08: 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1
Node 09: 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0
Node 10: 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0
Node 11: 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0
Node 12: 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0

                        Slot 1 to 26
```

Figure 2. MTDMA 13 Nodes 26 Slots

2.3 P-persistent Protocols

There have been a number of attempts to improve the performance of the system using a *p*-persistent approach [7-13]. In a *p*-persistent scheme, the node listens to the channel first. If the channel is not busy, the message is transmitted with probability *p* (assuming $p < 1$). With probability $1 - p$, the node waits for one time slot. At the end of the wait time, the node senses the channel and repeats the process. If the channel is busy, the node continues listening to the channel persistently.

In [7], the authors describe a method to optimize the throughput of a wired local area network by varying the transmission probability of the stations. In this scheme, each node monitors throughput of the shared medium over fixed intervals of 5 milliseconds, and adjusts the probability *p* based on the previous transmission probability and the change in the channel throughput.

To adjust the *p* value dynamically in a *p*-persistent CSMA protocol, the authors in [8] calculate probability *p* based on the number of nodes (*n*) that have data ready to send as follows: If *n* is less than 63, the probability *p* is set to $\frac{1}{n*16}$; otherwise, *p* is set to $\frac{1}{63 \times 16}$. In this approach, if the channel is idle, then a node computes *p*; otherwise, it continues to listen. Then, a random wait time is set according to *p*. If the wait time expires, and if the channel remains idle, the node transmits. Otherwise, this node repeats the listening process again.

In [1], another method that changes probability *p* based on the number of tasks in a node was proposed. In brief, if a node has more tasks than its neighbors, it is guaranteed to have the higher probability. In the proposed scheme, each node periodically observes its task activity and estimates the number of times it attempts to access the shared medium over a period of time, and based on this estimation, the node compute the probability *p*.

Recently, several works have employed the *p*-persistency based techniques to various mobile applications [11-13]. Works in this direction include some schemes proposed for ad hoc networks and underwater acoustic sensor networks. The authors of the study [12] proposed a new priority supported Medium Access Control (MAC) protocol for a vehicular ad hoc network using a time slotted *p*-persistent channel access mechanism to differentiate the services packets into multi-priority on the control channel. In [13], variants of multi-hop *p*-persistent slotted ALOHA protocol were designed for monitoring specific underwater area, and the performance of the proposed techniques in underwater acoustic sensor networks was numerically analyzed.

3. Modified P-persistent TDMA (MPTDMA)

MTDMA improves the throughput of a wireless network by allowing more nodes to transmit in the same time slot. Even though our discussion is heavily based our previous TDMA-based protocol in [4], it is possible to use other scheduling schemes, such as GRAND [5] and Ju and Li's [6], to enhance the performance of others.

MTDMA provides the foundation of the proposed scheduling scheme. On top of the scheduling vectors generated by MTDMA, we employ a simple *p*-persistent method to determine whether or not a node should transmit when its time slot is 1. To recap, 1 means a node has a permission to transmit data at the time slot; 0 is for otherwise.

3.1 Assumptions

Several assumptions should be addressed before we detail our algorithm. First, nodes know their scheduling vectors and those of their neighbors. Second, we do not address the hidden- terminal problem. If hidden-terminals occur during testing, nodes will log them as collisions and will discard the data. Third, nodes use probability *p* to determine whether or not to transmit data. Fourth, nodes do not use probability $1 - p$; this is fundamentally different from the original *p*-persistent CSMA approach. Last, aside from knowing the scheduling vectors, nodes must actively monitor their transmission queue utilizations. Moreover, they also keep track of the number of failed transmissions – generally caused by collisions – and the number of successful transmissions. With the provided information, MPTDMA determines the node's eligibility to send data when its time slot has come.

3.2 Dynamic p-persistent TDMA

MPTDMA algorithm can be described in a nutshell.

$$p_{new} = (\alpha_1)(\alpha_2 \times success\_rate + \alpha_3 \times p_{old}) + \alpha_4 \times queue\_util + \beta, \tag{2}$$

$$\text{where } \beta = \alpha_5 \times (1 - [(\alpha_1)(\alpha_2 \times success\_rate + \alpha_3 \times p_{old}) + \alpha_4 \times queue\_util])$$

The alpha values determine the level of importance of the parameters. These values must satisfy the following properties:

1. α_1, α_2, α_3, α_4, and $\alpha_5 \geq 0$
2. $\alpha_1 + \alpha_4 \leq 1.0$
3. $\alpha_2 + \alpha_3 \leq 1.0$
4. $\alpha_5 \leq 1.0$.

MPTDMA algorithm depends on three crucial parameters, which comprise success ratio (*success_rate*), previous or old transmission probability (p_{old}), and transmission queue utilization (*queue_util*). Using these values, the node calculates p_{new} when its current time slot is 1. Finally, p_{new} is checked against the predetermined threshold (*PROB_THRESHOLD*). If p_{new} is greater than or equal to the assigned threshold, the node proceeds to transmit data. Otherwise, it doesn't transmit. Instead, it will wait and will re-compute p_{new} when the time slot is 1 again. We will detail these parameters in the following subsections.

3.2.1 Success Ratio (*success_rate*)

Every time a node transmits, it logs the transmission attempt, either failed or succeeded. A failed attempt means a collision has occurred while the node was transmitting. A succeeded transmission speaks for itself. The success ratio (symbolized by *success_rate*) is an exponential probability distribution function expressed in terms of failed transmission attempts and total transmission attempts. The formula is:

$$success\_rate = f(x) = e^{-x} \quad \text{where } x \text{ is } \frac{failed\ transmission\ attempts}{total\ transmission\ attmepts} \tag{3}$$

The success ratio remains high as the node achieves many successful transmission attempts. Conversely, the algorithm imposes a stricter policy on bad nodes — those with many failed transmission attempts — by exponentially decreasing the success ratio. As the bad node delivers more packets, the success ratio gradually improves.

3.2.2 Previous Transmission Probability (p_{old})

MPTDMA adopts previous transmission probability (p_{old}) to control probability p_{new}. The old probability helps the node to cope with sudden changes in other parameters. For instance, Node A has $p = 0.2$ in one time slot, and it obtains $p = 0.9$ for the next available time slot. Technically, this is not fair for the other nodes because Node A doesn't have enough credibility to obtain $p = 0.9$. Using p_{old}, we can prevent a node from doing so. The algorithm orders the node to increase or decrease the probability at a reasonable pace. In (2), we incorporate the success ratio with the previous transmission probability to smooth out the new probability.

3.2.3 Transmission Queue Utilization (*queue_util*)

Queue utilization determines how busy a node is. MPTDMA algorithm gives busy nodes more priority than non-busy ones because they need to clear their queues before queue overflow happens instead of dropping newly generated packets due to buffer overflow.

3.2.4 Beta (β)

Success ratio, previous transmission probability, and queue utilization favor nodes with high success ratio and high queue utilization. If any of these parameters is too low, nodes may suffer starvation. For example, nodes with low queue utilization have low p values, and they may never get a chance to transmit data. To prevent starvation of a certain node with low p values, we employ a boosting value β. MPTDMA systematically adds β or $\alpha_5 \times (1 - p_{new})$

to p_{new} as a certain time has passed.

For example, we can calculate β for *Node 0* in Fig. 2.

Node 0: 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1

For simplicity's sake, we call the scheduling vector a frame. Note that the frame is repeatable; for example, *Node 0*'s frame is repeated after every 26th time slot. First, we count the number of 1's in one frame; we name this variable *total-frame-count*. Since *Node 0* has 1's in slots 9, 10, 13, 19, 21, and 26, we have *total-frame-count* = 6. As the frame is repeated, the *current-frame-count* is incremented by 1 until *current-frame-count* is equal to *total-frame-count*. The pseudo-code for calculating p_{new} with the boosting value β is shown in Fig 3.

```
compute $p_{new}$ with (3)
find total-frame-count in one frame;
if (frame is repeated and
    current-frame-count < total-frame-count) {
    current-frame-count is incremented by 1;
    $\alpha_5$ = current-frame-count / total-frame-count;
    $\beta = \alpha_5 * (1 - p_{new})$;
}
// Add boosting value to $p_{new}$
$p_{new} = p_{new} + \beta$;
if ( $p_{new}$ >= PROB_THRESHOLD ) {
    node transmits data
    reset current-frame-count to 0;
}
else
    node waits for next time slot
```

Figure 3. Pseudo-code for calculating p_{new}

3.2.5 Probability Threshold (*PROB_THRESHOLD*)

The *PROB_THRESHOLD* value is between 0.0 and 1.0. When the probability threshold is 0.0, both MPTDMA and MTDMA have similar performance. We've performed an experimental study to determine a good probability threshold. We assume both the success ratio and old probability are high initially. This study includes a network of 16 nodes using the scheduling vectors in Fig. 2. There are two network conditions. One network is for low traffic, and the other is for high traffic.

Fig. 4 shows the attempt ratios as a function of the probability threshold in a low traffic network. When the threshold is 0.1 and 0.2, we experience a lot of bad attempts and collisions. This is correct; more transmitters mean more collisions. As the threshold increases, the number of good transmission attempts improves. According the results shown in Fig. 4, any threshold value between 0.70 and 0.80 exhibits a good number of good transmissions attempts and reduces the number of bad transmission attempts.

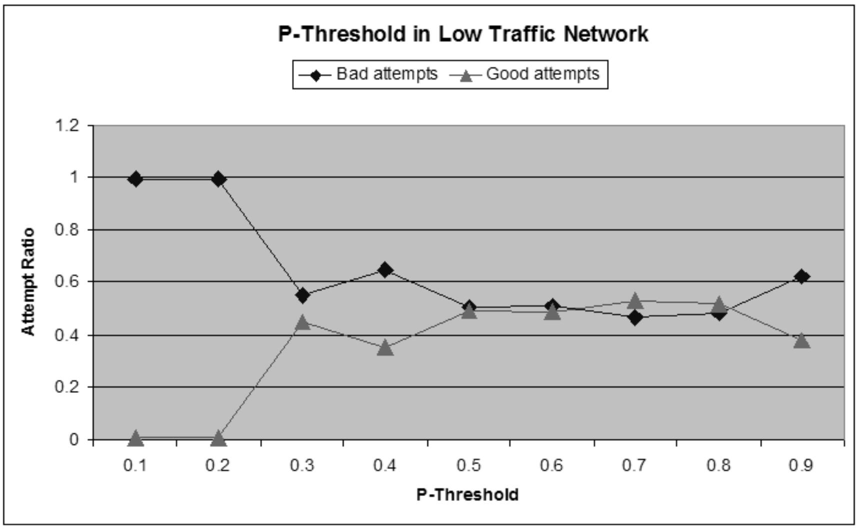


Figure 4. *P*-threshold for Low Traffic Networks

Fig. 5 demonstrates the effects on the number of bad or good attempts when all nodes are busy (i.e., the queue utilization is high at all times). The busier the network, the more collisions occur. Therefore, the probability threshold should be set higher. This small study allows us to select an optimized probability threshold for both busy and non-busy networks. In our ns-2 simulation study, we predefine the *PROB_THRESHOLD* = 0.70, which is a good value for both low and high traffic scenarios. However, we can further improve the performance by designing an adaptive algorithm that automatically adjusts the threshold value (*PROB_THRESHOLD*) according to the current traffic condition.

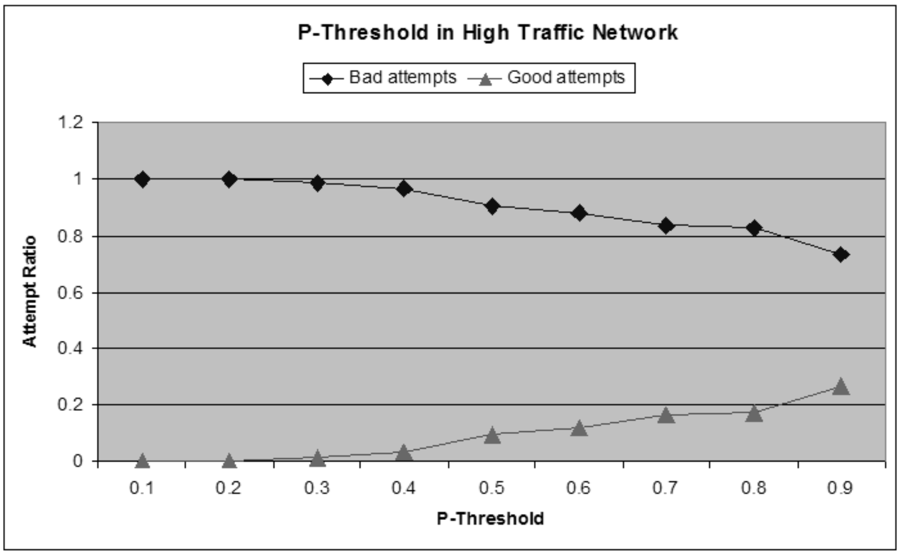


Figure 5: *P*-threshold for High Traffic Networks

4. Simulation Study

We have implemented the proposed MAC scheduling scheme using a network simulator ns-2 [3], and have conducted a simulation study using various network configurations and traffic patterns. In this simulation study, we compare the performance of TDMA, Modified TDMA, and Modified P-persistent TDMA. At the MAC layer of ns-2, we alternate these three MAC protocols for performance analysis. Table 1 summarizes the default values of the parameters of the proposed approach used in the simulation study.

Table 1. Fixed Simulation Parameters of MPTDMA.

Simulation Parameter	*Value*	*Simulation Parameter*	*Value*
α_1	0.75	*success_rate*	1.00
α_2	0.60	*old probability* (p_{old})	1.00
α_3	0.40	PROB_TRESHOLD	0.70
α_4	0.25		

4.1 Scenario Scenarios

The simulation scenarios include networks with 90 and 130 nodes; the sizes of these networks are 500m x 450m and 500m x 650m, respectively. The nodes are aligned in a grid topology, and each node is placed 50 units away from its neighbors. Furthermore, the transmission radius for a node is 70 units to ensure multi-hop connections. Therefore, each node should have a maximum of eight neighbors.

For each of the above networks, eight different sub-scenarios are generated, and each sub-scenario has 2, 4, 6, 8, 10, 12, 14, or 16 concurrent connections. Concurrent connections are simply UDP agents with constant bit rate (CBR) traffic generators. Sources and destinations are randomly chosen and are attached to an UDP agent. For simplicity in the evaluation process, a node is used once. If a node is chosen as a source, it will not be listed as another source or destination. For CBR traffic generators, the packet size is 512 bytes, and the interval for the packets is 500 milliseconds.

In the simulation study, we use the two-ray ground reflection radio model, which incorporates direct and ground reflection paths. Along with this propagation model, nodes are equipped with omnidirectional antennas [3]. Each node operates in half-duplex mode, meaning the node allows either transmitting or receiving at a given time. Additionally, the default routing protocol in the simulations is the dynamic source routing (DSR) [14].

4.2 Simulation Results

For a complete study of how the protocols affect the network performance, we capture simulation information for all layers of the network, which include medium access control (MAC), network layer (RTR), transport layer (AGT), and queue (IFQ).

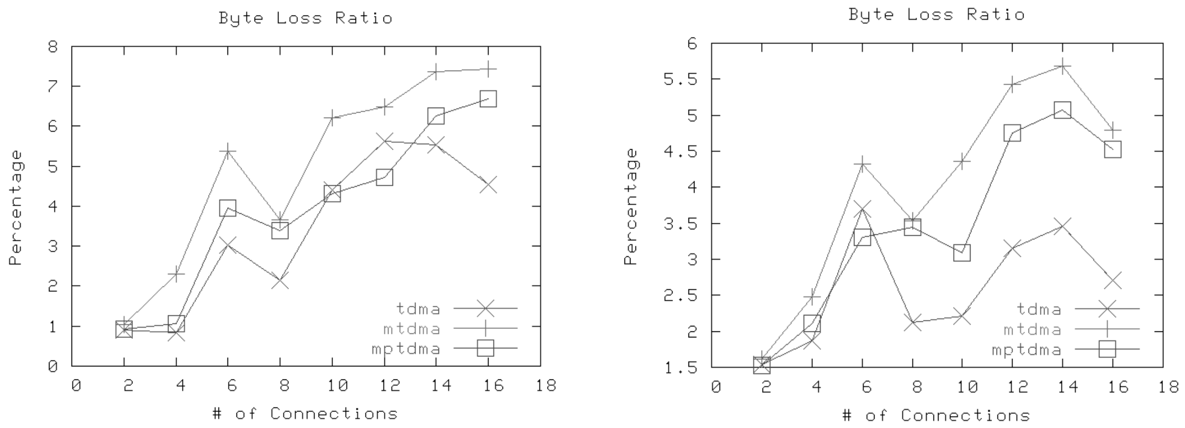
4.2.1 Performance Metrics

We use the following performance metrics to evaluate TDMA, MTDMA, and MPTDMA:

- Byte Loss Ratio: Byte loss ratio is the ratio of the total number of bytes dropped to the total number of bytes sent. At the MAC layer, there is no collision in TDMA. However, at higher layers, the data may get dropped due to queue overflow as well as wireless errors. Both MTDMA and MPTDMA drop bytes because of queue overflow, collisions, and wireless errors.
- Average Throughput: The throughput is defined as the number of bits per second correctly received. First, we compute the throughput for nodes individually, and then, by averaging throughputs for all nodes, we derive the average throughput in Mbps.
- Collision Count: Collision count is an extra performance evaluation tool exclusively for MTDMA and MPTDMA. This is the total number of collisions occurred during the entire simulation time.

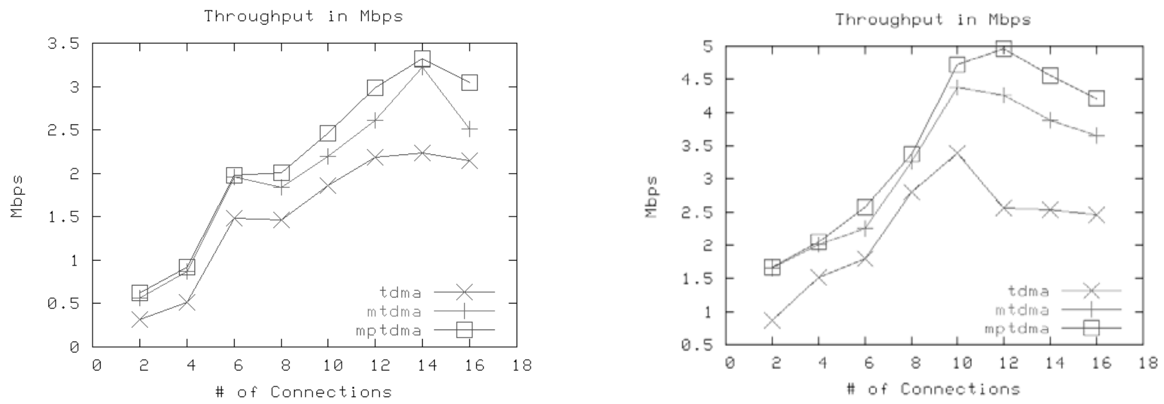
4.2.2 Results for Byte Loss Ratio

As you can see in Fig. 6, MTDMA loses more packets due to collisions. Since MPTDMA prevents nodes with low probabilities for sending, its byte loss ratio is not as severe as MTDMA’s. Obviously, TDMA has the lowest byte loss ratio since there are no collision losses. If we are to trace only at the MAC layer, this ratio will be 0 for TDMA. However, TDMA experiences byte loss due to queue overflow and losses related to wireless errors.



(a) 90 nodes (b) 130 nodes

Figure 6: Byte Loss Ratio (%)



(a) 90 nodes (b) 130 nodes

Figure 7: Throughput (Mbps)

4.2.3 Results for Throughputs

According to the throughput charts in Fig. 7, MPTDMA has the best throughput followed by MTDMA and TDMA. The result aligns with the byte loss ratio. MPTDMA sends more but doesn't lose much data, whereas MTDMA sends out data often with a higher byte loss ratio.

As the number of concurrent connections increases, the offered load generally increases. At a certain congestion threshold, increasing the offered load at a node runs the risk of losing more packets. For the 90-node network, the congestion threshold is between 14 and 16 concurrent connections. The congestion threshold for the 130-node network is between 10 and 12 concurrent connections. The network throughput is affected by the congestion threshold, and, when the threshold is reached, the network throughput deteriorates. This is evidenced after 14 concurrent connections in Fig. 7 (a) and after 10 concurrent connections in Fig. 7 (b).

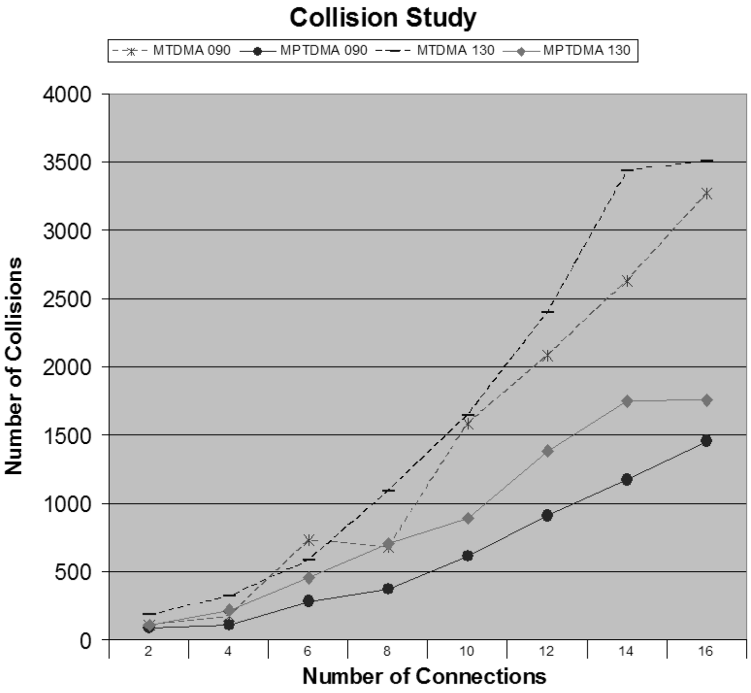


Figure 8. Total number of Collisions: MTDMA vs MPTDMA

4.2.4 Results for Collisions

The final study relates to the number of collisions in MTDMA and MPTDMA. For this study, we consider network with 90 nodes and 130 nodes. Table 2 and Fig. 8 show that MTDMA's number of collisions is almost twice of that of MPTDMA. Based on the results of this simulation study, we conclude that MPTDMA conserves more energy by eliminating the number of dropped packets. Note that, in MPTDMA, only half of the data sent out get dropped.

Table 2. Total number of Collisions for networks with 90 nodes and 130 nodes

MAC Algorithms	Number of Connections							
	2	4	6	8	10	12	14	16
MTDMA with 90nodes	113	175	731	6798	1584	2081	2626	3268
MPTDMA with 90 nodes	92	111	282	375	618	913	1172	1453
MTDMA with 130 nodes	186	319	584	1091	1645	2400	3439	3504
MPTDMA with 130 nodes	106	217	454	706	890	1384	1748	1756

5. Conclusion

In this paper, we have introduced a new MAC protocol by combining a *p*-persistent approach and a fixed TDMA-based schedule protocol. Based on prior research on *p*-persistent protocols, we've devised a new *p*-persistent scheme named MPTDMA. Using the transmission success rate, previous probability p_{old}, and the transmission queue utilization, MPTDMA computes an updated transmission probability p_{new}, which determines whether or not a node can transmit. According to the results of the simulation study, the proposed protocol outperforms TDMA and MTDMA in terms of network throughput. Also, MPTDMA overtakes MTDMA by experiencing fewer collisions and more efficient in terms of energy consumption. This MAC protocol study offers valuable insights on how different MAC protocols behave and how to optimize performance in a wireless network. In the future, we will continue to optimize the number of collisions (hence, energy consumption) and increase the throughput even more by adjusting some parameters dynamically based on the measured network traffic characteristics.

References

[1] Pack S, Choi J, Kwon T, Choi Y., "TA-MAC: Task Aware MAC Protocol for Wireless Sensor Networks". The 63rd IEEE Vehicular Technology Conference (VTC 2006-Spring), Melbourne, Australia, May 7-10, 2006. Vol. 1. Pp 294–298. http://dx.doi.org/10.1109/VETECS.2006.1682823

[2] Zha W, Rose QH, Yi Q, Yu C., "An Adaptive MAC Scheme to Achieve High Channel Throughput and QoS Differentiation in a Heterogeneous WLAN". The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine'06), Waterloo, Canada, August 7-9, 2006. http://dx.doi.org/10.1145/1185373.1185407

[3] The Network Simulator - ns-2. Available at: http://www.isi.edu/nsnam/ns/ (last accessed Jan. 6, 2015)

[4] Youn J-H, Bose B., "A Topology-Independent Transmission Scheduling in Multihop Packet Radio Networks". The IEE Global Telecommunications Conference

(GlobeCom’01), San Antonio, USA, Nov. 25-29, 2001. Vol. 3, Pp 1918–1922. http://dx.doi.org/10.1109/GLOCOM.2001.965908

[5] Chlamtac I, Farago A., “Making Transmission Schedules Immune to Topology Changes in Multi-hop Packet Radio Networks”. IEEE/ACM Transaction Networking 1994; Vol. 2: Vol. 2, Issue 1. Pp 23–29. Feb. 1994. http://dx.doi.org/10.1109/ICC.1993.397600

[6] Ju J-H, Li VOK., “An Optimal Topology-Transparent Scheduling Method in Multihop Packet Radio Networks”. IEEE/ACM Transaction Networking, Vol. 6. Pp 298–306. June 1998. http://dx.doi.org/10.1109/90.700893

[7] Siew CK, Er MH., “An Adaptive Algorithm for Performance Optimization of Modified P-persistent CSMA/CD Local Area Network”. The IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering (TENCON’93), Beijing, China, Oct. 19-21, 1993. Vol. 3, Pp 345–351. http://dx.doi.org/10.1109/TENCON.1993.327951

[8] Xiaoming C, Geok-Soon H., “A Simulation Study on the Predictive P-persistent CSMA Protocol”. The 35th Annual Simulation Symposium (SS’02), San Diego, USA, April 14-18, 2002. Vol. 3, Pp 175–178. http://dx.doi.org/10.1109/SIMSYM.2002.1000172

[9] Bruno R, Conti M, Gregory E., “Optimization of Efficiency and Energy Consumption in P-Persistent CSMA-based Wireless LANs”. IEEE Transactions on Mobile Computing. Vol. 1, No. 1. Pp 10–31. January 2002. http://dx.doi.org /10.1109/TMC.2002.1011056

[10] Hu Jin, Jun-Bae Seo, Dan Keun Sung, “Stability Analysis of -Persistent Slotted CSMA Systems With Finite Population”. IEEE Transactions on Communications. Vol. 62, Issue: 12. Pp 4373–4386. 2014. http://dx.doi.org/10.1109/TCOMM.2014.2366749

[11] Alwakeel, S., Prasetijo, A., “A virtual P-Persistent bandwidth partitioning manager for VANET's broadcast channel”. International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, April 14-16, 2014. Vol. 1. Pp 1212–1215. http://dx.doi.org/10.1109/ICMCS.2014.6911411

[12] Caixing Shao, Supeng Leng, Yan Zhang, Huirong Fu, “A multi-priority supported p-persistent MAC protocol for Vehicular Ad Hoc Networks”. The IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, April 1-4, 2012, Pp 2532–2537. http://dx.doi.org/10.1109/WCNC.2012.6214225

[13] YanPing Zhang, “Performance of p-persistent slotted Aloha for underwater sensor networks”. The IEEE International Conference on Computing, Networking and Communications (ICNC), Honolulu, USA, Feb. 3 – 6, 2014, Pp 583–587. http://dx.doi.org/10.1109/ICCNC.2014.6785401

[14] D. Johnson, D. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing”, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996. http://dx.doi.org/10.1007/978-0-585-29603-6_5

Copyright Disclaimer