Delay-Constrained Optimized Packet Aggregation in High-Speed Wireless Ad Hoc Networks

Peyman Teymoori, Nasser Yazdani, and Ali Sehati

Router Lab., Electrical & Computer Engineering Department,
University of Tehran, Tehran, Iran
{p.teymoori,yazdani}@ut.ac.ir, a.sehati@ece.ut.ac.ir

Abstract. In recent years, high-speed WLANs are introduced to service growing demand of delay-sensitive and multimedia applications. To improve efficiency at the MAC layer of high-speed WLANs, few researches have tried to utilize approaches such as Aggregation in which a number of packets concatenated into a larger frame to reduce protocol overheads. Since transmitting larger frames causes increases in delay and jitter which are crucial, especially in delay-sensitive and multimedia applications, selecting the best aggregated frame size is significant. In this paper, we propose an analytical model for optimized packet aggregation (OPA) that finds the optimized aggregation size with regard to delay constraints of nodes. OPA enhances one of the aggregation methods, aggregation with fragment retransmission (AFR) scheme, and models aggregation by a constrained convex optimization problem to maximize network throughput while constraining the delay. Simulation results show that OPA increases throughput and decreases the average delay as well.

Keywords: High-speed wireless networks, IEEE 802.11n, Aggregation, Convex optimization.

1 Introduction

Increasing the number of users of wireless technologies has raised the demand for real-time and delay-sensitive applications, and higher bandwidth [1]. For real-time applications and delay-sensitive traffic, constraining and preserving predictable delay is of significant importance [2]. Real-time applications over WLANs require that packets arrive at their destinations in a timely manner, and many applications require high throughput. Toward meeting this demand, IEEE 802.11n has emerged that supports physical rates of up to 600 Mbps and inherent QoS [3].

It has been shown that the inefficient protocol overhead of IEEE 802.11 distributed coordination Function (DCF) results in a theoretical throughput upper limit and delay lower limit for the IEEE 802.11 based protocols, even the wireless data rate goes to infinity [4]. Frame aggregation is one of the efficiency improvement methods at the MAC layer. It not only reduces the transmission time for preamble and frame headers, but also reduces the waiting time during random backoff period for successive frame transmissions [5]. IEEE 802.11n supports aggregated data frame sizes of up to 64KB at the MAC layer [3].

P. Sénac, M. Ott, and A. Seneviratne (Eds.): ICWCA 2011, LNICST 72, pp. 389–400, 2012.

Although frame aggregation can increase throughput at the MAC layer under ideal channel conditions, a larger aggregated frame will cause each station to wait longer before its next chance for channel access and makes channel access unpredictable [5]. Therefore, there is a tradeoff between throughput and delay for frame aggregation at the MAC layer. Moreover, under error-prone channels, corruption in a large aggregated frame may waste a long period of channel time and lead to a lower MAC efficiency [5], and also increase in delay.

In this paper, due to the importance of limiting delay in delay-sensitive applications in high-speed WLANs [6], in order to constrain delay at the MAC layer of wireless ad hoc networks as well as preserving the performance gain of frame aggregation, we propose an analytical model for wireless the MAC layer of high-speed wireless networks, called optimized packet aggregation (OPA). OPA finds the maximum aggregation size for nodes that does not increase nodes' delays. In other words, as the main parameter of optimization, OPA considers delay requirements of nodes and permits them to transmit a particular amount of data through determining the aggregation size at the MAC layer.

The main approach of increasing throughput is to aggregate more packets. Since increasing aggregation size results in more transmission time and increasing delays of nodes, the tradeoff between throughput and delay is calculated using an accurate analytical model by OPA. It optimizes throughput using a convex optimization method [7] while prevents delay increase through optimization constraints.

The rest of the paper is organized as follows. Section 2 reviews similar works. We introduce the analytical model of OPA in Section 3, and discuss in Section 4 the implementation issues. Section 5 explains algorithmic issues of our model while Section 6 presents detailed simulation results. Finally, we summarize our conclusions in Section 7.

2 Related Works

There are some approaches to improve the efficiency of high-speed wireless networks such as burst acknowledge (Burst ACK) proposed by [8], and block acknowledgement (Block ACK), e.g., [9], which try to reduce the protocol overhead. Approaches such as [9], [10] and [11] aggregate several packets to compensate overhead by increasing data length. In addition, in [11], through PM, the receiver station is allowed to piggyback a data frame to the sender station once if the receiver station has a frame to send to the sender. IEEE 802.11e [12] MAC mechanism also introduces transmission opportunity (TXOP) through which a station is allowed to transmit multiple data frames without entering backoff procedure.

With increasing demand for real-time and multimedia applications over wireless, the IEEE 802.11n Working Group standardized a new Medium Access Control (MAC) and Physical Layer (PHY) specification [3]. The throughput performance at the MAC layer of IEEE 802.11n is improved by aggregating several frames of at most 64KB before transmission. Although simulation results of [13] demonstrates the effectiveness of 802.11n MAC layer enhancement, the standard does not specify exactly how many packets should be aggregated and how delay requirements are treated.

In [5] authors study the performance of IEEE 802.11n under unidirectional and bi-directional data transfer. They also numerically propose an optimal frame size adaptation algorithm with A-MSDU under error-prone channels. Authors in [14] try to solve the performance anomaly of IEEE 802.11 multi-rate wireless networks. They introduce a transmission time which is similar to transmission opportunity (TXOP) proposed by IEEE 802.11e.

Convex optimization has been used in many engineering applications in order to reach an optimum situation. As an instance, the authors in [15] address the problem of rate assignment to sources of transmitting data and solve it through its dual problem using gradient projection algorithm.

These works only concentrate on how to concatenate packets into a larger frame to reduce protocol overhead. Since most of the works in the literature such as [9] only focus on proposing effective aggregation schemes, the effect of aggregation on delay and bounding such delay are paid no attention. To the best of our knowledge, this is the first work on analyzing and bounding delay caused by aggregation which is crucial to high-speed WLANs, especially for multimedia applications that are one the requirements of these networks [16], [17], and [18].

3 Optimized Packet Aggregation (OPA)

3.1 Network Model

We assume that there are n nodes belonging to the set of nodes $\mathcal{N}=\{1,\ldots,n\}$ which are contending for the wireless channel. Physical transmission rate is r_i for node i and the goal is to maximize the channel throughput by considering delay constraints of nodes. Upon accessing channel, we assume that the node i aggregates a number of x_i packets each of which is of the average length l_{avg}. This assumption can be utilized as the underlying aggregation mechanism in all methods.

MAC overhead is usually caused by specific headers and frame checksums. In addition, there is an extra overhead imposed by the physical layer to transmit a packet such as SIFS and DIFS, control frames such as RTS and CTS, physical layer preambles, etc. We denote overhead of the protocol by P^{OH} meaning that summation of all overhead durations, in seconds, in which no data packet is transmitted.

By these assumptions, now, we can define the node utility function which indicates how much a situation is preferable for a node. In the context of wireless channel access, the more a node accesses the channel, the more this situation is profitable for the node. Throughput of a node has a direct relation with its number of aggregated packets [9]. Because of the use of BlockACK, the more packets a node aggregates, the more utility it achieves and, accordingly, the more throughput increase it has [9]. In other words, the utility function of a node i is an increasing function of its number of aggregated packets x_i. We used a logarithmic function which is strictly increasing as the utility function and for node i, it is defined as

$$U_i\left(x_i\right)=\log x_i \quad , \quad x_i>1 . \tag{1}$$

The behavior of a logarithmic function to its input is closer to a real throughput function of the number of packets, and also, simulation results of [9] approve this. The logarithmic function (1) is strictly concave and gives us some interesting properties as discussed in Section 5. With the goal of maximizing the network utility

$$U(\mathbf{x}) = \sum_{i=1}^{n} U_i(x_i) \tag{2}$$

which is the summation of utilities of nodes and **x** means a vector whose elements are x_i, $i \in \mathcal{N}$. This function has an optimum point which maximizes the function U, and we require finding the optimum point in order to maximize the network utility.

We define delay constraint of a node as the maximum duration the node can postpone sending and let the other nodes transmit their packets. In other words, this duration specifies how much the node can wait to access the channel, and it is usually specified by traffic characteristics of the node. Since access to the channel of a wireless ad hoc network is stochastic, nodes may access the channel differently but we assume that the channel access mechanism is fair and the expected values of the number of node accesses to the channel are the same. Moreover, we assume fair access to the channel meaning that if node i finishes transmitting, all other nodes can have access to the channel if they have packets to send. IEEE 802.11-based protocols provide such fairness in a long-term basis [19]. Using the delay calculation of [5], by d_i, we denote how long it takes that node i is able to transmit. Analytically, for each node i we should have $B_i(\mathbf{x}) \le d_i$ where

$$B_i(\mathbf{x}) = (n-1)P^{OH} + l_{avg} \sum_{j \ne i} (x_j / r_j) \, . \tag{3}$$

This equation means that the time required for the other nodes to transmit their packets should be less than or equal to d_i. This time comprises the time of transmitting the number of packets that each node sends and the protocol overhead.

3.2 Optimization Problem

Primal Problem. We model the aggregation problem as a solution to the following optimization problem which is formulated as:

$$\textit{Maximize} \;\; U(\mathbf{x}) \tag{4}$$

$$\textit{Subject to} \;\; B_i(\mathbf{x}) \le d_i \;\; \text{for all} \;\; i \in \mathcal{N} \tag{5}$$

This means that we try to find a vector **x** which specifies how many packets should be aggregated by each node in order to achieve the maximum utility in the network. This is performed by considering constraints (5).

Dual Problem. Although the problem (4) can be separated among nodes, its constraints will remain coupled over the network. The coupled nature of the problem necessitates using a centralized method which imposes great computational overhead to the system. In order to have a distributed solution and for the sake of simplicity in designing the channel access protocol, we solve the problem through its dual. First, by defining the Lagrangian problem, we take the constraints into account which leads to

$$L(\mathbf{x},\boldsymbol{\lambda}) = U(\mathbf{x}) - \sum_{i=1}^{n} \lambda_i (B_i(\mathbf{x}) - d_i) . \tag{6}$$

where λ_i is the Lagrange multiplier associated with the i^{th} inequality constraint and the vector $\boldsymbol{\lambda}$ is called the dual variables of the problem (4) where $\boldsymbol{\lambda} = (\lambda_i, i \in \mathcal{N})$. Then, the Lagrangian dual function is defined as

$$g(\boldsymbol{\lambda}) = \sup_{\mathbf{x}} L(\mathbf{x},\boldsymbol{\lambda}) . \tag{7}$$

which is the maximum value of the Lagrangian over **x**. The dual function yields upper bounds on the optimal value of the problem (4) and (5) [7]. In order to solve (7), we should find x_i such that

$$\frac{\partial L(\mathbf{x},\boldsymbol{\lambda})}{\partial x_i} = 0 . \tag{8}$$

Solving the above equation results in

$$x_i = \frac{r_i}{l_{avg} \sum_{j \neq i} \lambda_j} \tag{9}$$

Substituting (9) in (7) yields

$$\begin{aligned} g(\boldsymbol{\lambda}) = {} & \log \frac{r_1}{l_{avg}(\lambda_2 + \lambda_3 + \ldots + \lambda_n)} + \log \frac{r_2}{l_{avg}(\lambda_1 + \lambda_3 + \ldots + \lambda_n)} + \ldots + \log \frac{r_n}{l_{avg}(\lambda_1 + \lambda_2 + \ldots + \lambda_{n-1})} \\ & - \lambda_1 (n-1) P^{OH} + \lambda_1 d_1 - \lambda_2 (n-1) P^{OH} + \lambda_2 d_2 - \ldots - \lambda_n (n-1) P^{OH} + \lambda_n d_n \\ & - \frac{\lambda_1}{(\lambda_1 + \lambda_3 + \ldots + \lambda_n)} - \frac{\lambda_1}{(\lambda_1 + \lambda_2 + \lambda_4 + \ldots + \lambda_n)} - \ldots - \frac{\lambda_1}{(\lambda_1 + \lambda_2 + \ldots + \lambda_{n-1})} \\ & - \frac{\lambda_2}{(\lambda_2 + \lambda_3 + \ldots + \lambda_n)} - \frac{\lambda_2}{(\lambda_1 + \lambda_2 + \lambda_4 + \ldots + \lambda_n)} - \ldots - \frac{\lambda_2}{(\lambda_1 + \lambda_2 + \ldots + \lambda_{n-1})} \\ & \ldots \\ & - \frac{\lambda_n}{(\lambda_2 + \lambda_3 + \ldots + \lambda_n)} - \frac{\lambda_n}{(\lambda_1 + \lambda_3 + \ldots + \lambda_n)} - \ldots - \frac{\lambda_n}{(\lambda_1 + \ldots + \lambda_{n-2} + \lambda_n)} \end{aligned} \tag{10}$$

The Lagrangian dual problem is expressed as:

$$\textit{Minimize} \;\; g(\boldsymbol{\lambda}) \tag{11}$$

$$\textit{Subject to} \;\; \boldsymbol{\lambda} \succeq 0 . \tag{12}$$

The Lagrange dual problem (11) is a convex optimization problem, since objective to be maximized is concave and constraint is convex. Due to the duality theory, a dual problem is always convex, and due to the strong convexity of the primal problem (4), it is guarantees that solving the dual problem will result in optimal solution for the primal problem. The above problem can be solved by differentiating $g(\boldsymbol{\lambda})$ of λ_i which leads to

$$\frac{\partial g}{\partial \lambda_i} = -(n-1)P^{OH} + d_i - \sum_{j \neq i} \frac{1}{\sum_{k \neq j} \lambda_k} . \tag{13}$$

As an example, the above equation for $i = 1$ is as follows:

$$\frac{\partial g}{\partial \lambda_1} = -(n-1)P^{OH} + d_1 - \frac{1}{(\lambda_1 + \lambda_3 + \ldots + \lambda_n)} - \frac{1}{(\lambda_1 + \lambda_2 + \lambda_4 + \ldots + \lambda_n)} - \ldots - \frac{1}{(\lambda_1 + \lambda_2 + \ldots + \lambda_{n-1})} \tag{14}$$

Solving the equation $\partial g / (\partial \lambda_i) = 0$ computes the optimum λ_i but because of the complexity of the (10), we cannot represent a closed-form solution. Then, the optimum point is calculated iteratively.

In order to obtain a distributed solution with low computational complexity, we solve the dual problem using gradient projection method [7] which iteratively steps toward the opposite direction of the gradient of the objective function of the problem. Using the following iterative equations, the optimum values of λ_i for each node *i* is calculated. Therefore, for the dual problem (11), we get

$$\lambda_i^{(k+1)} = \left[\lambda_i^{(k)} - \gamma^{(k)} \frac{\partial g}{\partial \lambda_i^{(k)}} \right]^+ \tag{15}$$

$$\gamma^{(k+1)} = \frac{1}{\sqrt{k}} \tag{16}$$

where $\lambda_i^{(k)}$ is the value of λ_i at iteration k and $[z]^+ = \max(z, 0)$. This means that at iteration *k*, λ_i is updated and improved. Equation (16) shows the step length of that round. Equation (15) is the descent method which produces a minimizing sequence to solve an optimization problem [20]. By this equation we mean an algorithm that computes a sequence of points $\lambda_i^{(0)}, \lambda_i^{(1)}, \ldots \in dom\ g$ with $g(\lambda_i^{(k)}) \to p^*$ as $k \to \infty$ where p^* is the optimum point. The algorithm is terminated when $p^* - g\left(\lambda_i^{(k)}\right) \le \varepsilon$, and $\varepsilon > 0$ is some specified tolerance.

4 Implementation

In the absence of a centralized coordinator in an ad hoc network, nodes should follow a distributed approach to compute how many packets they should aggregate to reach the optimal situation. Each node only knows its delay requirement (d_i), physical rate (r_i), and protocol overhead (P^{OH}). From (13) it is inferred that each node also requires $\boldsymbol{\lambda}$ and the number of nodes, *n*, in order to solve (15).

The nodes can infer the number of nodes contending for the channel by listening to transmissions in the channel. In addition, the nodes should have λ_i for all *i*. For this reason, a particular field is considered in the MAC header of the protocol which is called *LM* (Lagrange Multiplier). Each transmitting node *i*, calculates the latest value of its λ_i and puts this value in this field. The receiving node and the other listening nodes extract this value, and update their local information.

Local information of **λ** values of the other nodes is organized as a list, called *LM_list*. As communication proceeds and data packets are transmitted, nodes extract values of the LM field of transmitting packets and update their local list. This process continues until all nodes reach an optimized value for **λ**.

Table 1. Optimized Packet Aggregation (OPA) pseudo code for each node *i*

Initialization	On sending	On listening/receiving

$\lambda_i = 1$;
$k = 1$;
LM_List = Empty;
	p = output_packet;
	if(p.type = data_packet) {
	$\lambda_i^{(k+1)} = \left[\lambda_i^{(k)} + \gamma^{(k)} \frac{\partial g}{\partial \lambda_i^{(k)}}\right]^+$;
	$\gamma^{(k+1)} = \frac{1}{\sqrt{k}}$;
	$x_i = \frac{r_i}{l_{avg} \sum_{j \neq i} \lambda_j}$;
	$x_i = \min(\max(x_i, Min\_Frame\_Len), Max\_Frame\_Len))$;
	update_LM_list(node$_i$.ID, $\lambda_i^{(k+1)}$);
	p.LM = $\lambda_i^{(k+1)}$;
	$k = k + 1$;
	aggregate(p, $\lfloor x_i \rfloor$); // aggregates at most $\lfloor x_i \rfloor$ packets into p
	}
		p = input_packet;
		if(p.type = data_packet)
		update_LM_list(
		extract_ID(p),
		extract_LM(p));

The only value that should be transmitted is **λ** which is carried by the LM field of the MAC header. This value implicitly consists of all information required to find an optimal solution like node physical rate and delay requirement. Therefore, transmitting entire parameters of nodes is not required. In Table 1, pseudo code of this process is presented. The parameters *Min_Frame_Len* and *Max_Frame_Len* represent the minimum and maximum allowed frame lengths. Nodes also assume zero-waiting meaning that if there are fewer packets than x_i in their queues, they just aggregate available packets and do not wait for further packets.

5 Evaluation

5.1 Algorithm Analysis

Time complexity of OPA is $O(n^2)$. Each node *i* should compute the value of (15) in order to update its λ_i value. The only time-consuming part of this formula is calculating $\partial g / (\partial \lambda_i)$. Equation (13) shows how many calculation steps are required to find this value for node *i*.

OPA does not impose any message complexity since each node informs other nodes of its **λ** only through the LM field in the MAC header. Since **λ** values are in a small fixed range of real numbers, the LM field that encodes **λ** also imposes a very insignificant overhead.

Since time complexity of OPA is not high, and also, OPA does not impose any message complexity, the overhead of OPA, especially on energy consumption, is insignificant. This means it not required that nodes worry about energy issues.

One important issue regarding OPA is whether it can find the optimum point, i.e., the aggregation size leading to maximum throughput. If a function is concave/convex, it has a unique maximum/minimum point.

The primal problem stated by (4) is strictly convex and admits a unique maximizer because $(\partial^2 U_i(x_i)) / (\partial x_i{}^2) < 0$ which indicates that the entire utility functions are concave, and since according to (2), $U(\mathbf{x})$ is a nonnegative and non-zero weighted sum of strictly concave functions, it is strictly concave.

5.2 Algorithm Stability

The main concern regarding the proposed algorithm is its stability under erroneous and dynamic conditions of a WLAN under which nodes may join or leave the network. We analyzed the algorithm in the presence of error in estimating the number of nodes. Referring to Table 2, error percent means that how much the number of nodes estimated by a node may deviate from what it actually is. For example, if there are 20 nodes in the network, 20% error may cause that a node estimates the number of nodes to a value in the range of [16, 24]. Results show that if, for example, there are 50 nodes in the network, and all nodes may have 20% deviate in their estimations, their average delay may increase or decrease to at most 25% of their optimum value.

Although the above results show that the delay may increase/decrease during these conditions, the fact is the algorithm should quickly converge to the new optimum point after any changes in the network. Referring to Table 2, convergence time shows how long, in seconds, it takes to reach the new optimum point in average. Evaluations for an extensive amount of input data show that the algorithm can converge to the optimum point after 100 iterations on average. This means that in real network scenario, each node can converge to the optimum point after receiving nearly 100 MAC data frames that according to Table 2, it takes less than 1 second which is fast enough to track changes such as node movement and sleep in WLANs.

Table 2. Delay deviation in the presence of error

Number of nodes	10%	20%	30%	Convergence time (s)
6	10%	20%	35%	0.29
10	9%	22%	36%	0.42
20	11%	25%	35%	0.65
50	11%	25%	39%	0.89

6 Simulation Results

We used the AFR implementation as the base aggregation method and enhanced it by our proposed approach. AFR is one of the best schemes proposed for high-speed wireless networks. The implementation is performed in *NS-2* [21]. We used implementation and simulation scenarios of [22]. This code represents AFR implementation which is published by the authors of [9]. In the network topology, STA *i* sends packets to STA *i*+1. Results are reported for two different types of traffic, CBR and HDTV which are requirements of high-speed WLANs. In addition, improvement in the network utilization is computed by *Throughput* and *Average delay* which are introduced by [9].

HDTV is one of the requirements of high-speed wireless LAN protocols such as IEEE 802.11n [6]. It has a constant packet size of 1500 bytes, a sending rate of 19.2-24Mbps, and a 200ms peak delay requirement. We investigate OPA and AFR HDTV performance with a 128Mbps PHY data rate.

In addition, the simulation time is 10 seconds, that is, the nodes keep transmitting packets for 10 seconds. Results are averaged over all nodes and over 15 different runs with 95% confidence level. Fig. 1 shows the throughput and delay performance of these schemes for different number of STAs. In all of the following scenarios, AFR is executed with the frame size to which it responds well, i.e. 32KB. On the contrary, OPA follows a dynamic packet aggregation scheme and the frame size may vary. It equals to the optimized packet length that the algorithm proposed for that situation.

From Fig. 1, it can be observed that OPA results in shorter delays than AFR and, approximately, this value is half of the AFR average delay. Moreover, in the cases where BER is 10^{-5} and 10^{-6}, OPA improves throughput and where BER is 10^{-4}, these two approaches reach the same throughput although OPA decreases the average delay for all different BER values. When the number of nodes is small, both approaches cause short delay. As more nodes are added to the network, satisfying delay constraints becomes more critical since the number of contending nodes for the channel increases.

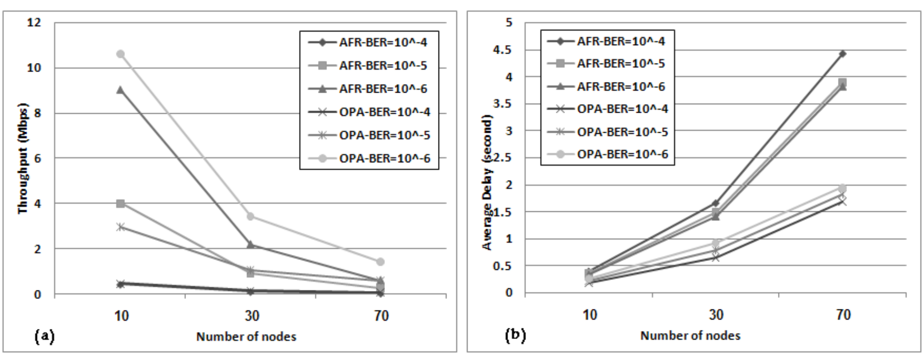

Fig. 1. HDTV traffic for various numbers of nodes

Other type of traffic which is used as test traffic is CBR which generates UDP packets at a constant rate. Two different scenarios which utilize CBR traffic are implemented. In the first one, each node constantly generates packets and sends them to a particular destination. The network is saturated and nodes transmit CBR streams at the physical rate. The number of stations varies from 10 to 90 and results are extracted for different BERs. The simulation duration is also 10 seconds, and results are depicted in Fig. 2. Results show that OPA method outperforms AFR especially where the network is heavily overloaded, i.e., the number of stations increases.

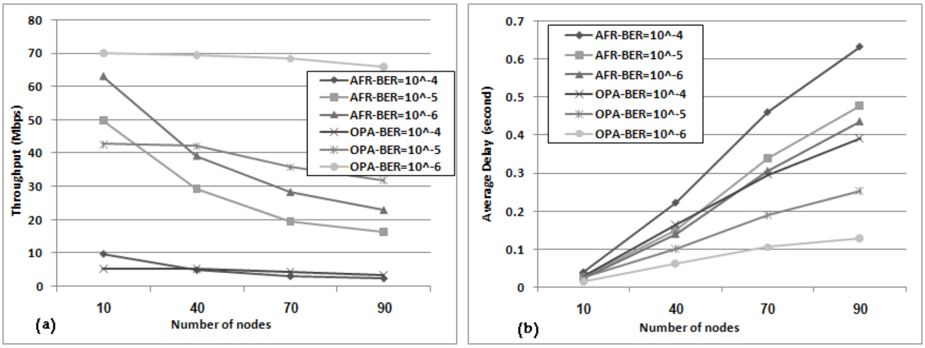

Fig. 2. CBR traffic for various numbers of nodes

From Fig. 2(a), it can be observed that OPA is not very sensitive to increase in the number of nodes in terms of throughput and decreasing slope of the lines are smaller in OPA. This means that the throughput decreases slower as the number of nodes increases in OPA than that of AFR. Fig. 2(b) shows the average delay of the above network scenario. Referring to this Figure, results indicate that OPA is more successful is decreasing delay under different BERs.

In the second scenario of using CBR traffics, the number of nodes is fixed and the physical rate varies from 54 to 432Mbps. There are 40 nodes sending CBR streams to one another. In Fig. 3, performance results of this scenario are presented for BERs 10^{-5} and 10^{-6}.

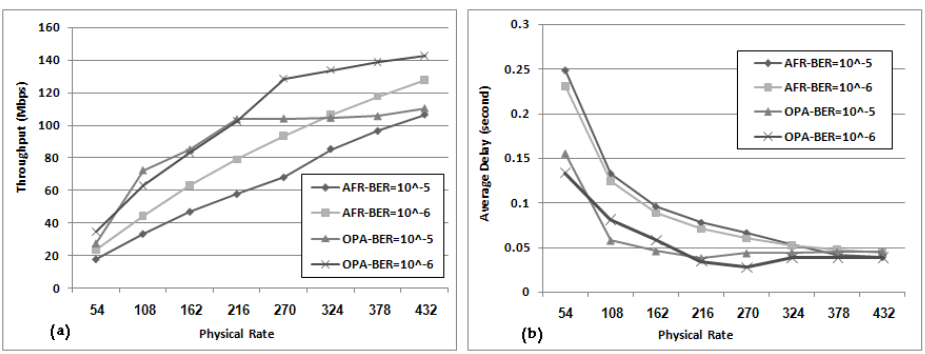

Fig. 3. CBR traffic for various physical rates

Fig. 3 shows that when the physical rate is low, OPA outperforms AFR. This consequence is obvious because OPA optimizes network access and the optimization process, in general, performs well when there is lack of resources. Therefore, in this situation, access to the channel should be accomplished wisely, and OPA handles this situation well. Delay constraints of OPA impose restrictions on access durations to the channel and as a result, delay in not increased. As the physical rate increases, and the network becomes less saturated, the average delay of both methods converges to a particular value although the OPA throughput is larger. Delay values of both approaches converges to 0.05 for the physical rate 432Mbps in which the network becomes unsaturated, and the same result is also held for throughputs.

7 Conclusion

One of the requirements of wireless networks is providing high-speed transmission of data, especially, multimedia traffics. There are few approaches that increase efficiency of the medium access control (MAC) layer through aggregating packets and reducing the protocol overhead. In this paper, in order to achieve high efficiency at the MAC layer of these networks as well as constraining resultant delay of large aggregation sizes, we proposed an analytical model of the wireless medium access, optimized packet aggregation (OPA), that finds the optimized aggregated size. This model, as the main parameter of optimizing, considers delay requirements, and permits nodes to transmit a particular amount of data as an aggregated data frame at the MAC layer to bound channel access delay. To evaluate the proposed model, we extended the AFR implementation in *NS*-2. Simulation results indicate that our method, OPA, decreases the average delay while increasing throughput, especially in saturated situations where the number of nodes and the traffic rate are large. As future work, we will try to extend the analytical model to consider different aspects such as the error rate.

References

1. Marcelo, M., Carvalho, J.J., Garcia-Luna-Aceves: Delay Analysis of IEEE 802.11 in Single-Hop Networks. In: 11th IEEE International Conference on Network Protocols (2003)
2. Raptis, P., Vitsas, V., Paparrizos, K.: Packet Delay Metrics for IEEE 802.11 Distributed Coordination Function. Mobile Networks 14, 772–781 (2009)
3. IEEE, IEEE 802.11n-2009—Amendment 5: Enhancements for Higher Throughput (2009)
4. Xiao, Y., Rosdahl, J.: Performance analysis and enhancement for the current and future IEEE 802.11 MAC protocols. ACM SIGMOBILE Mobile Computing and Communications Review 7, 6 (2003)
5. Lin, Y., Wong, V.W.: Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs. In: IEEE Globecom 2006, pp. 1–6 (2006)
6. IEEE: IEEE 802.11n TGn Sync proposal technical specification (May 2005)
7. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ Pr, Cambridge (2004)
8. Vitsas, V., Chatzimisios, P.: Enhancing performance of the IEEE 802.11 distributed coordination function via packet bursting. In: IEEE Global Telecommunications Conference Workshops, GlobeCom Workshops 2004, pp. 245–252. IEEE, Los Alamitos (2004)

9. Li, T., Ni, Q., Malone, D., Leith, D., Xiao, Y., Turletti, T.: Aggregation With Fragment Retransmission for Very High-Speed WLANs. IEEE/ACM Transactions on Networking 17, 591–604 (2009)
10. Li, T., Ni, Q., Xiao, Y.: Investigation of the block ACK scheme in wireless ad hoc networks. Wireless Communications and Mobile Computing 6, 877–888 (2006)
11. Xiao, Y.: IEEE 802.11 performance enhancement via concatenation and piggyback mechanisms. IEEE Transactions on Wireless Communications 4, 2182–2192 (2005)
12. IEEE, IEEE 802.11e, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements (2005)
13. Wang, C., Wei, H.: IEEE 802.11n MAC Enhancement and Performance Evaluation. Mobile Networks and Applications 14, 760–771 (2009)
14. Razafindralambo, T., Lassous, I.G., Iannone, L., Fdida, S.: Dynamic packet aggregation to solve performance anomaly in 802.11 wireless networks. In: Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems - MSWiM 2006, p. 247. ACM Press, New York (2006)
15. Low, S., Lapsely, D.: Optimization flow control. I. Basic algorithm and convergence. IEEE/ACM Transactions on Networking 7, 861–874 (1999)
16. Ketchum, J., Al, E.: System Description and Operating Principles for High Throughput Enhancements to 802.11. 11, vol. 802, pp. 11–14. IEEE, Los Alamitos
17. Mujtaba, S.A., Al, E.: TGn sync proposal technical specification. TGn Sync Technical Proposal R 802, 11-04/889 (2004)
18. Xiao, Y.: IEEE 802.11n: enhancements for higher throughput in wireless LANs. IEEE Wireless Communications 12, 82–91 (2005)
19. Li, Z., Nandi, S., Gupta, A.K.: Modeling the Short-term Unfairness of IEEE 802.11 in Presence of Hidden Terminals. Performance Evaluation 63, 441–462 (2006)
20. Zhu, H., Chlamtac, I.: Performance analysis for IEEE 802.11e EDCF service differentiation. IEEE Transactions on Wireless Communications 4, 1779–1788 (2005)
21. Hindi, H.: A Tutorial on Convex Optimization II: Duality and Interior Point Methods. In: American Control Conference, pp. 686–696 (2006)
22. Network Simulator (NS), `http://www.isi.edu/nsnam/ns/`
23. AFR Implementation, `http://www.hamilton.ie/tianjili/afr.html`