A CONTEXT-AWARE APPROACH FOR MULTIMEDIA PERFORMANCE OPTIMIZATION USING NEURAL NETWORKS IN WIRELESS LAN ENVIRONMENTS*

Po-Chiang Lin, Chiapin Wang, Tsungnan Lin
Graduate Institute of Communication Engineering, National Taiwan University,
Taipei, Taiwan, R.O.C.
{d94942014, d92942012, tsungnan}@ntu.edu.tw

ABSTRACT

Packet size is one of the most important factors that would affect the user-perceived multimedia QoS in the wireless LAN environments. The time-varying channel characteristics make it difficult to find the exact relationship between the packet size and the throughput and decide an optimal packet size in advance. Furthermore, every node would suffer different channel conditions. In this paper, we tackle this problem by an optimization approach. A context-aware framework is designed to optimize the packet size adaptively in order to maximize the throughput. In this approach each node abstracts its specific context via the throughput from the time-varying wireless environments. The obtained throughput information is the instantaneous integrated effect of all contexts in wireless LAN environments. This approach adopts neural networks to learn the complex nonlinear function between the packet size and the throughput and adaptively adjusts the packet size. Simulation results show that out method can cope with the time-varying wireless channel conditions and improve the perceived QoS of wireless multimedia services.

1. INTRODUCTION

Providing multimedia services over wireless networks is challenging because of the time-varying channel characteristics and the scarce wireless resources. Hence it is important to optimize the protocols and parameters in order to meet the multimedia QoS requirement. In the error-prone wireless channels, the packet size is one of the important factors that would affect the user-perceived QoS. However, deciding an appropriate packet size is not a trivial work. If the packet size is too large, since merely a bit error would destroy the whole packet when there is no error correction mechanism, the packet success rate would be decreased and the throughput is degraded. If the packet size is too small, the headers of the MAC and PHY layer would occupy a large proportion of the packet and make the transmission inefficient.

Some previous works try to analyze the relationship between the packet size and the throughput in 802.11 wireless LAN environments. Bianchi proposes a Markov Chain model to analyze the relationship between the throughput and the various parameters, including the transmission probability, initial size of the backoff window, number of stations, maximum backoff stage, and packet size [1]. However, he assumes that the wireless channel is ideal, i.e., the transmission is always successful if there is no collision. His research results show that the throughput is a monotonically increasing function of the packet size, i.e., the larger the packet size, the better the throughput. This assumption is not realistic since every node in the wireless channel would suffer different error-prone channel conditions. Yin et al. analyze the performance of IEEE 802.11 DCF and consider the effects of several factors, including the number of contention nodes, packet size, the transmission collision and the packet error probability [2]. Following this work, they analyze the effects of contention window and packet size on the energy efficiency of wireless LANs [3]. In their analytical works, all the nodes are assumed to be homogeneous with the same channel condition and transmission parameters. This assumption is not realistic in real wireless LAN environments. S. Ci and H. Sharif present an optimal packet size predictor in [4]. They derive the mathematical equation of throughput, which is a function of the packet size. By differentiating this equation to the packet size and set the result equal to zero, the optimal packet size is obtained.

In brief, these previous works have some drawbacks that make them unsuitable for the packet size optimization problem in wireless multimedia environments. First of all, in order to derive the mathematical equation of throughput, these analytical works have some simplistic assumptions. For example, no hidden terminal problem, saturated traffic, homogeneous channel condition for all nodes, and no interference from other communication systems such as Bluetooth. These assumptions make these works unrealistic for real world environments. Secondly, these approaches require the knowledge of channel conditions which are always changing. Therefore, the timely measurement or estimation of all the necessary factors is another challenging problem. Therefore, it requires other solutions to optimize the packet size.

In this paper we propose a context-aware approach that adjusts the packet size adaptively. Context means what in the real world is captured, which is a subset of what is given in the real world [5]. Context-awareness provides a way to compensate for the abstraction of real-world situations in time-varying environments. Instead of measuring or estimating all the parameters in the wireless multimedia environments, we adopt the context-aware approach to abstract the integrated information, and use this information to adjust the packet size. The abstracted context may contain some noise. Therefore, we adopt neural networks, which have been proved to be effective in modeling a system which contains some noise. Our approach adopts on-line training to cope with the time-varying channel conditions, and to optimize the packet size adaptively based on the gradient information. Experiment results show that our method can effectively optimize the packet size and maximize the throughput for wireless multimedia services.

The paper is organized as follows. In Section 2 we formulate the packet size optimization problem and present our context-aware approach using neural networks. In Section 3 we present the simulation and results to show the effectiveness of our approach. Finally, conclusions are given in Section 4.

* This work was supported in part by Yulong Corp. under grant 95-S-C01-A and by Taiwan National Science Council under grant 94-2219-E-002-017 and 94-2213-E-002-066.

2. THE CONTEXT-AWARE APPROACH USING NEURAL NETWORKS

In this section we present our context-aware approach to maximize the throughput. First of all, we formulate the packet size optimization problem as follows. Suppose that P is the packet size which does not include the headers of MAC and PHY layers. The throughput T is defined as the successfully received amount of data normalized to the channel capacity C in a unit of time. Therefore T can also be regarded as the efficiency of the wireless channel. Suppose that H is the header size including the MAC and PHY headers. It is also assumed that the propagation time between the wireless transmitter and receiver is ignored. Thus the time needed to transmit a packet including the headers is

$$t_{packet} = \frac{H+P}{C}. \qquad (1)$$

The effective time that transmits useful information is defined as

$$t_{effective} = \frac{P}{C} = t_{packet} \cdot \frac{P}{P+H}. \qquad (2)$$

However, for a successful transmission of one packet, it takes some ineffective time $t_{ineffective}$, including the DIFS, SIFS, random waiting time, RTS/CTS frames, the unavoidable retransmission time, and so on [6]. Therefore, the normalized throughput T can be expressed in time domain as

$$T = \frac{t_{effective}}{t_{packet} + t_{ineffective}}. \qquad (3)$$

There are many factors that would affect $t_{ineffective}$, including the packet size, number of nodes, noise, interference, and so on. Hence the throughput T also depends on these factors. If we only concentrate on the packet size and set other factors as fixed, the throughput T is a complex nonlinear function of the packet size P, i.e., T = F(P). Here comes the problem about how to maximize the throughput T based on the optimal packet size P. We formulate the packet size optimization problem as

$$P^{optimal} = \arg\max_{P} T = \arg\max_{P} F(P). \qquad (4)$$

Previous solutions focus on analyzing and obtaining the function F(P). When the analytical formula F(P) is known, solving the equation $\partial F(P)/\partial P = 0$ would get the optimal value of P. However, as described above, the throughput T depends on many other factors such as the number of nodes, noise, interference, and so on. In wireless channels these factors vary greatly with time and it is difficult to measure or estimate them. In other words, F(.) is a context-dependent function. Besides, the previous solutions require some simplistic assumptions in order to derive the mathematical equation of F(P), as mentioned in Section 1. These assumptions make these works unrealistic for real world environments. Therefore, it requires other solutions to optimize the packet size.

We propose a context-aware approach that adjusts the packet size adaptively. The contexts in wireless LAN environments, such as the number of nodes, the location and moving speed, the contention state, noise, and interference, are always changing with time. Furthermore, the contexts encountered by the different nodes in the wireless LAN environments are also different. In our context-aware approach each node abstracts its own context via the throughput from the wireless environments. The obtained throughput information is the instantaneous integrated effect of all contexts in wireless LAN environments. Assume that at the *n*th time of adjusting, the throughput is T(n) and the packet size is P (n). At the next time of adjusting, the packet size is set as

$$P(n+1) = P(n) + \Delta P(n). \qquad (5)$$

The ΔP(n) depends on the gradient of T(n) with respect to P(n), i.e.,

$$\Delta P(n) = \alpha \frac{\partial T(n)}{\partial P(n)} \qquad (6)$$

where α is the adjusting rate. We use the context information, i.e., the local information of T(n), and neural networks learn the context-dependent function F(.) and then to obtain the gradient $\partial T(n)/\partial P(n)$. Neural networks have been proven to be effective in modeling the complex relationship between the input signal and the output signal in a noisy circumstance. We adopt the multilayer perceptron (MLP) [7], one of the most popular neural networks, to model the relationship between the throughput and the packet size and get the gradient. The back propagation algorithm is used to adjust this network and minimize the error between the actual response and the desired (target) response. After the modeling is accomplished, the packet size is adjusted based on our modified multilayer perceptron and back propagation algorithm.

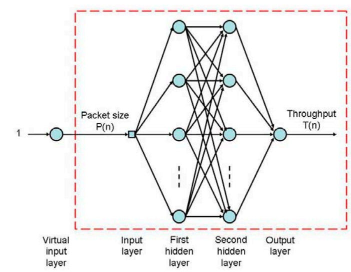

Fig. 1. Architecture of a multilayer perceptron with two hidden layers.

i, j, k	The indices of the neurons in the different layers of the network. Neuron j is located in the right layer next to neuron i, and neuron k is located in the right layer next to neuron j
n	the identification of training patterns
L	The total number of layers. Layer L means the output layer.
E(n)	the instantaneous sum of error squares
$e_j(n)$	the error signal at the output of neuron j for iteration n
$d_j(n)$	the desired response for neuron j
$y_j(n)$	the function signal appearing at the output of neuron j at iteration n
$w_{ji}(n)$	the synaptic weight connecting the output of neuron i to the input of neuron j at iteration n
$\Delta w_{ji}(n)$	the weight correction applied to the synaptic weight $w_{ji}(n)$
$v_j(n)$	the weight sum of all synaptic inputs plus bias of neuron j at iteration n
$f_j(.)$	the activation function of neuron j
$b_j(n)$	the bias applied to neuron j
η	learning rate
m_l	the number of nodes in layer l
T(n)	the throughput of the *n*th training patterns
P(n)	the packet size of the *n*th training patterns

Table. 1. The notation used in this paper

A multilayer perceptron consists of an input layer, one or more hidden layers, and an output layer. The dashed rectangle of Fig. 1 shows the architecture of a multilayer perceptron with two hidden layers. The input signal (the packet size in this case) is applied to the input layer and propagates through the network layer by layer from left to right. Each arrow in this figure represents an adjustable synaptic weight, and each node represents a computation neuron. To ease the detailed description of the multilayer perceptron, we first summarize the notation in Table. 1. The original back propagation algorithm consists of two passes: a forward pass and the backward pass. In order to adjust the packet size, we add the third pass: the adjusting pass. The details of these operations are described in the following paragraphs.

A. forward pass

The purpose of the forward pass is to get the actual response to the specified input signal. When the signal propagates through a synapse, its value is multiplied by the synaptic weight. The input of a neuron is the synaptic weighted sum of the output of its previous layer plus the bias. For example, the input of the *j*th neuron in the *l*th layer is

$$v_j(n)=\sum_{i=1}^{m_{l-1}} w_{ji}(n)\, y_i(n)+b_j(n)\,. \tag{7}$$

The input of this neuron is applied to the activation function $f_j(.)$. Therefore, the output of this neuron is

$$y_j(n)=f_j\left(v_j(n)\right). \tag{8}$$

The activation function should be differentiable everywhere. It is usually a linear function or a sigmoid function. The output of this neuron propagates through the network to the next layer, and the same operation is performed again. Finally the signal aggregates in the output layer, and the output signal (the throughput) is obtained. During the forward pass the synaptic weights and the bias are all fixed.

B. backward pass

The purpose of the backward pass is to adjust the synaptic weights and the bias in order to minimize the error between the actual response and the desired (target) response. The instantaneous sum of error squares E(n) can be written as

$$E(n)=\frac{1}{2}\sum_{j=1}^{m_L} e_j^{\,2}(n)=\frac{1}{2}\sum_{j-1}^{M_L}\left(d_j(n)-y_j(n)\right)^2 \tag{9}$$

where neuron j is an output node. The back propagation algorithm applies a correction $\Delta w_{ji}(n)$ to the synaptic weight $w_{ji}(n)$, which is proportional to the gradient $\partial E(n)/\partial w_{ji}(n)$. The correction $\Delta w_{ji}(n)$ is defined by

$$\Delta w_{ji}(n)\equiv -\eta\frac{\partial E(n)}{\partial w_{ji}(n)}=\eta\,\delta_j(n)\, y_i(n) \tag{10}$$

where η is the learning rate of the back-propagation algorithm and $\delta_j(n)$ is the local gradient which is defined as $-\partial E(n)/\partial v_j(n)$. When neuron j is located in the output layer, it is straightforward to compute the local gradient $\delta_j(n)$ from its definition. When neuron j is located in the hidden layer, by recursive back-propagation and chain rule manners, the back-propagation formula for the local gradient $\delta_j(n)$ is

$$\delta_j(n)=f_j'\left(v_j(n)\right)\sum_k \delta_k(n)\, w_{kj}(n) \tag{11}$$

The bias could be adjusted with the similar manner. The forward pass and the backward pass are executed recursively until E(n) falls below some level. Then the adjusting pass is executed.

C. adjusting pass

The purpose of the adjusting pass is to adjust the packet size in order to maximize the responding throughput. The synaptic weights and the bias that have been adjusted well in the backward pass are set as fixed in the adjusting pass. Following we show that how to obtain the gradient $\partial T(n)/\partial P(n)$ so that the packet size P(n + 1) can be adjusted based on Eq. (5) and (6). We add a virtual input layer to the original architecture of the multilayer perceptron as Fig. 1 shows. The input signal to the virtual input layer is set as 1. The activation function of the neuron in this virtual input layer is linear. Therefore, the original input signal (packet size) to the input layer is equal to the synaptic weight between the virtual input layer and the input layer. Therefore,

$$\frac{\partial T(n)}{\partial P(n)}=\frac{\partial T(n)}{\partial w_{ji}(n)},\begin{cases}\text{neuron j is an input node, } j=1\\ \text{neuron i is a virtual input node, } i=1\end{cases} \tag{12}$$

Now we calculate the partial derivative similar to the back propagation algorithm.

$$\begin{aligned}\frac{\partial T(n)}{\partial w_{ji}(n)}&=\frac{\partial T(n)}{\partial v_j(n)}\frac{\partial v_j(n)}{\partial w_{ji}(n)}\\ &=\lambda_j(n)\, y_i(n)\end{aligned} \tag{13}$$

where $\lambda_j(n)$ is the local gradient which is equal to $\partial T(n)/\partial v_j(n)$. When neuron j is an output node, it is obvious that $\lambda_j(n)$ is equal to $f_j'(v_j(n))$. When neuron j is located in the hidden layer, $\lambda_j(n)$ would have to be determined recursively. Assume that neuron k is located in the output layer and the neuron j is located in the hidden layer which is adjacent to the output layer. When $\lambda_k(n)$ is known, $\lambda_j(n)$ can be derived as follows:

$$\lambda_j(n)=\sum_{k=1}^{m_L}\lambda_k(n)\, w_{kj}(n)\, f_j'\left(v_j(n)\right) \tag{14}$$

Using Eq. (14) recursively can obtain the gradient $\partial T(n)/\partial P(n)$. Apply this gradient to Eq. (5) and (6), and then the packet size can be adjusted adaptively.

In order to cope with the time-varying channel conditions, we use on-line training and operate the three passes iteratively to model the instantaneous (P, T) relationship and optimize the packet size.

D. Considerations on Complexity

Our approach requires on-line training to learn the non-linear relationship between the throughput and the packet size. Although it leads to additional computing cost, there are some parameters in our approach that can be adjusted in order to reduce the complexity and to meet the multimedia QoS requirement. First, each training is stopped when one of these two conditions occurs: when the mean square error is lower than the pre-defined error threshold, or when the number of iteration exceeds the pre-defined maximum value. Therefore, the error threshold and the maximum number of iteration can be designed to get the compromise between the performance and the complexity. Secondly, the number of hidden layers, hidden nodes and most-recent training patterns would also affect the complexity, and they can be designed to reduce the complexity while keep the acceptable performance. Finally, the computing process can be implemented on the more powerful wireless nodes, such as the access points. For downlink transmissions from the access point to the wireless nodes, the access point directly executes the computing process; for uplink transmissions from the wireless nodes to the access point, the access point executes the computing process based on the information provided by the wireless nodes, and sends the computing results back to them. This method is applicable to the low computing-power wireless nodes, but the performance may not be optimal due to the transmission delay of channel context information and computing results.

3. SIMULATION AND RESULTS

In this section we show the simulation results based on two simple scenarios. The packet size is set to be adjusted once every time interval t. The simulation is kept for 100t, i.e., the packet size is adjusted for 100 times. There are two nodes in this wireless channel. The original bit error rate is set as 1e-5. The training patterns are derived from the modified analytical model of Bianchi’s work. In this modified analytical model, we consider more factors such as the error-prone channel conditions and the retry limit. For each time of modeling and adjusting, the most recent five (P, T) data sets are adopted as the training patterns. The detailed simulation parameters are shown in Table. 2. In the first scenario, the bit error rate changes from 1e-5 to 4e-5 at 50t in order to test the capability of our approach under the time-varying channel conditions. The simulation results are shown in Fig. 2. In this figure we also plot the theoretical curves of (P, T) relationships in different channel conditions in order to see that whether our approach can adjust the packet size to reach the maximum throughput. In Fig. 2, the packet size is adjusted from the initial 200 bytes to around 1250 bytes, which is the optimal packet size to get the maximum throughput. At 50t, the channel condition changes, and our approach automatically responds to the change. The packet size is adaptively adjusted to around 600 bytes, which is the optimal packet size to get the maximum throughput in that specific channel condition.

In the second scenario, the bit error rate is kept in 1e-5, but one more node joins in this wireless channel at 50t, and the initial packet size is set as 1800 bytes. The simulation result is shown in Fig. 3. This result also shows that our approach can effectively adjust the packet size to the optimal value under time-varying channel conditions.

We list the performance improvement of our proposed approach versus the original transmission without packet size adaptation in Table. 3. The percentages of throughput improvements before and after the change of channel conditions are all listed. These results show that no matter what the channel characteristics are, our approach can significantly improve the throughput for almost all the cases. This is because that the original transmission without packet size adaptation fails to guess an optimal packet size. Even when the original setting of packet size is coincident to be near optimal, such as the case of 1800 bytes in scenario 1, its value would be out-of-date when the channel conditions vary with time.

Data rate = 1Mbps	MAC header = 34bytes	DIFS = 50μs	Initial window size = 32
Slot time = 20μs	PHY header = 16bytes	SIFS =10μs	Max. window size = 1024
Learning rate = 0.01	ACK = 64bytes	Propagation delay = 1μs	Increasing factor = 2
# training patterns = 5	# hidden layers = 1	# hidden nodes = 4	Retry limit = 5

Table. 2. Simulation parameters

4. CONCLUSION AND FUTURE WORK

In this paper we present a novel optimization framework under the time-varying wireless multimedia environments. The packet size is adaptively adjusted in order to maximize the throughput. In our approach each node abstracts its own context information and the neural network is adopted to model the complex nonlinear relationship between the throughput and the packet size. After the appropriate modeling, the neural network can be used to optimize the packet size. The supervised learning is kept on-line in order to deal with the time-varying channel conditions. Simulation results show that our approach can effectively optimize the packet size and maximize the throughput.

This optimization framework can be further extended to jointly optimize the cross-layer parameters based on the multimedia QoS. For example, the modulation and coding schemes in the physical layer, and the backoff parameters and retry limit in the MAC layer, would affect the multimedia QoS. Moreover, the delay is also an important performance metric that would affect the user-perceived QoS other than the throughput. Our future work is to investigate the solutions of this joint optimization problem based on the optimization framework.

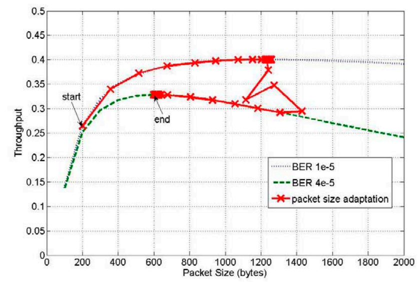

Fig. 2. The simulation result for the first scenario. The bit error rate changes from 1e-5 to 4e-5 at 50t. The initial packet size is set as 200 bytes.

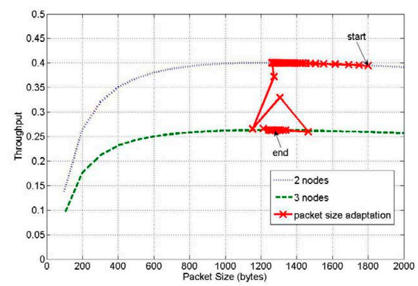

Fig. 3. The simulation result for the second scenario. The number of nodes changes from 2 to 3 at 50t. The initial packet size is set as 1800 bytes.

	Initial packet size (Bytes)	Throughput improvement (%)	
		Before	After
Scenario 1	200	51.25	30.60
	1800	1.40	28.44
Scenario 2	200	51.25	48.84
	1800	1.40	1.39

Table. 3. Throughput improvements of our approach before and after the channel condition changes for the two scenarios.

5. REFERENCES

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, March 2000, pp535-547.

[2] J. Yin, X. Wang, D. P. Agrawal, "Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function," in Proc. The 2004 IEEE Wireless Communications and Networking Conference (WCNC'04), Atlanta, 21-25 March 2004, pp.1654-1659.

[3] X. Wang, J. Yin, D. P. Agrawal, "Effects on contention window and packet size on the energy efficiency of wireless local area network," in Proc. IEEE Wireless Communications and Networking Conference 2005 (WCNC'05), New Oreland, 21-25 March 2005, pp.94-99.

[4] S. Ci, H. Sharif, "Adaptive optimal frame length predictor for IEEE 802.11 wireless LAN," in Proc. The 6th IEEE International Symposium on Digital Signal Processing for Communication Systems (DSPCS 2002), pp.59-63.

[5] H. W. Gellersen, A. Schmidt, M. Beigl, "Multi-sensor context-awareness in mobile devices and smart artifacts," Mobile Networks and Applications, Vol. 7, Issue 5, October 2002, pp.341-351.

[6] M. S. Gast, "802.11 Wireless Networks – The Definitive Guide," O'Reilly, 2002, pp.23-50.

[7] S. Haykin, "Neural Networks – A Comprehensive Foundation," second edition, Prentice-Hall, Inc. 1999, pp.156-175.