Multiscale Load Adaptive Scheduling for Energy Efficient Transmission over Wireless Networks

Shutao Sun1, Simin He2, Wen Gao1,2, Bin Pang2

1Graduate School, Chinese Academy of Sciences, Beijing 100039, China
2Institute of computing Technology, Chinese Academy of Sciences, Beijing 100080, China

{stsun,smhe,wgao,bpang}@jdl.ac.cn

ABSTRACT—Some wireless communication systems, such as mobile Ad Hoc network, operate on a limited battery supply, and energy efficiency is therefore crucial to the network survival time. This paper proposes a multiscale load adaptive scheduling (MLAS) scheme to determine the transmission time of the packets. In MLAS, traffic is predicted over the scale of time window, and then a transmission time stretching algorithm is proposed to adjust the packet transmission time packet by packet. Compared with the fixed transmission time scheme, MLAS can reduce the energy consumption of data transmission by up to 90% corresponding to the load factor and the burst characteristic. MLAS also outperforms the Look Ahead based scheme in the performance of energy efficiency with the same average delay.

I. INTRODUCTION

Wireless networking is enjoying its fast growth period in history. However, this growth is greatly limited due to the insufficient battery power in mobile system, which makes low energy consumption a key issue for the mobile wireless networking. Therefore the power management technology must be addressed for energy efficient wireless communication.

There are mainly two kinds of energy efficient scheduling techniques, namely channel adaptive scheduling and load adaptive scheduling. Extensive researches have been conducted on the channel adaptive scheduling. Wong et al. [8] studied channel allocation algorithms for cellular base stations. Ferracioli et al. [10] proposed a channel based scheduling scheme for the third generation cellular networks. Fu et al. [9] studied the transmission scheduling over a fading channel with energy and deadline constrain. Zhang and Wasserman [11] used dynamic programming and Markov model to study the tradeoff between throughput and energy efficiency. The works mainly focused on single-hop networks such as cellular and satellite systems, and solved the problems of resource allocation to maximize the Shannon capacity or channel adaptive scheduling to transmit a certain amount of data with the deadline constraints.

More recently, load adaptive scheduling has attracted more attention. Based on the following observation: first, in some wireless networks, such as Mobile Ad Hoc, many mobile terminators operate as routers, and peak performance is not always required; second, some flows, such as video streams, are variable bit rate, some researchers presented the load adaptive scheduling algorithms to achieve high energy efficiency by lowering transmission power and transmitting the data over a longer period of time.

Prabhakra et al. [1] presented a lazy packet scheduling algorithm for transmitting data from one node to the other. Assuming a Possion arrival traffic model, an online Lazy scheduling algorithm was proposed to determine the transmission time of a packet based on maximal mean arrival speed(denoted $\lambda_{max}$) of the packets and the queue length. This scheme can stretch the transmission time of a packet no more than 1.65 times of $(1/\lambda_{max})$. Therefore, the Lazy scheduling is apparently conservative for some kinds of traffic, such as variable bit rate video streams, which usually have peak-to-mean ratio of 4-10 [16]. Garmal et al. [2] focused on the situation of one to multiple points and proposed an online Move-Right Express algorithm. It uses Look Ahead Buffer to buffer the arrived packets for a certain period, which is termed as Look Ahead window, and then the number of backlog is used to calculate the transmission time of the packets for next interval. Raghunathan et al [4] proposed EWFQ algorithm to support QoS over wireless networks, which also determine the packet transmission time by Look Ahead window as Move-Right Express. The disadvantage of the Look Ahead based scheme is that it will introduce much more delay.

In this paper, we address the problem of minimizing energy consumed by an intermediate node in a multi-hop wireless network to transmit self-similar traffic to next hop. A new scheduling scheme is proposed to solve the problem of larger delay and insufficient energy saving. In our scheme, traffic is predicted over the scale of time window, which usually is several hundred milliseconds; then a transmission time stretch algorithm is proposed to adjust the packet transmission time packet by packet. The analysis and simulation results show that our scheme can save much more energy than the proposed algorithms such as online Lazy and those based on Look-Ahead while keeping the average delay in a reasonable region.

The rest of this paper is organized as follows. Section 2 introduces the load adaptive scheduling system framework. In section 3 and 4, basic mechanism and main components of our scheme are described. The simulation results are shown in section 5. At last, we end this paper with concluding remarks in Section 6.

This work is supported by the Knowledge Innovation Program in CAS, the National Natural Science Foundation of China under grant No.69983008, and the National High Technology Development 863 Program of Chain under grant No. 2001AA112100.
II. SYSTEM FRAMEWORK

A crucial aspect in the design of wireless networks, e.g., mobile Ad Hoc, is to optimize the system transmission energy consumption for maximizing the system lifetime. In this paper, we focus on the energy efficiency at the wireless network node which operates as a router.

Let \( e(t) \) denote the energy required to transmit one packet over duration \( t \). The following assumption was justified in [1]:

1) \( e(t) \geq 0 \).

2) \( e(t) \) is monotonically decreasing in \( t \).

3) \( e(t) \) is strictly convex in \( t \).

There is similar result for energy function for bit.

As it can be proven that if \( N \) bits of data need to be transmitted in interval \([t_1, t_2]\), that every bit uses the same interval \((t_2-t_1)/N\) is optimal scheme for minimizing the energy consumption. However, it is difficult to know how much traffic need to be transmitted from a router to the next hop in the near future because the traffic arrival follows a stochastic process. Although the traffic can be buffered and then transmitted with a proper modulation level, the tradeoff between delay and energy saving must be considered.

Because quadrature amplitude modulation (QAM) is both efficient and easy to implement, we will discuss our energy efficiency based on this modulation scheme. The energy consumption for transmitting a bit is given by [3]

\[
E = C_s \left( 2^b - 1 \right) + C_E b,
\]

(1)

where \( b \) is the modulation level in bits per symbol, and the parameter \( C_s \) and \( C_E \) depend on the radio implementation, the transmission distance, and the required error performance.

In order to minimize the energy consumption while keeping the average delay on a reasonable level, a scheduling scheme based on traffic prediction is designed to explore the opportunities for energy saving.

The framework of our energy efficient scheduling scheme is described as follows:

1) A transmitter with the function of dynamic modulation scaling (DMS) is used to transmit the network traffic, and the modulation level is adjusted dynamically.

2) The traffic rate is measured in a short interval \( T_{win} \) periodically.

3) The traffic measurement result will be used to predict the traffic rate in the near future by a predictor.

4) Based on the results of parameter estimation, traffic rate prediction and current backlog in buffer, an appropriate transmission time for the outgoing packet, i.e., the modulation level, will be determined.

Because the proposed scheme determines the packet transmission time by combining the multiscale load adaptive, i.e., adapting in the time scale of prediction window and packet by packet, we name it multiscale load adaptive scheduling (MLAS). Next we will discuss our scheme in detail.

III. TRAFFIC MEASUREMENT AND PREDICTION

Markov models have been traditionally used to analyze the behavior of data networks. However, recent works based on measurements of Ethernets [13], wide area data networks [14], and VBR video [15] traffic have shown that the traffic generation processes display burstiness that cannot be captured by Poisson Processes. Self-similar models, which can capture burstiness over many time scales, may be more appropriate. Recent study shows that the traffic in wireless data networks exhibits self-similar behavior as well [12].

There are three mathematical models often used to model the self-similarity effect: the fractional Gaussian noise (FGN), the fractional Brownian motion (FBM), and the fractional autoregressive integrated moving average (F-ARIMA) process.

In this paper, we use the fractional Gaussian noise process to model the traffic as seen at a wireless router.

Consider the cumulative traffic model

\[
Y(t) = mt + \sigma Z(t),
\]

(2)

where \( m \) is the mean rate, \( \sigma^2 \) is the variance in a unit time and \( Z(t) \) is a centered stationary stochastic process.

In order to save energy, the bandwidth required by the traffic should be predicted to adjust the modulation level. We will obtain the prediction and variance by measuring traffic at resolution \( T_{win} \). This gives a discrete traffic process

\[
\hat{X}_i = Y((i - 1)T_{win}, iT_{win}) - Y((i - 1)T_{win}),
\]

(3)

At time \( t \), assuming that we have known the sample path in the past, \( X_0, X_{2-t}, \ldots, X_{N-t} \), we can predict the random variable \( X_{t+1} \).

The choice of a prediction method is a tradeoff between prediction error and computational cost. In this paper, we consider only linear predictions, because the linear prediction methods have been considered as a simple and effective alternative for video and network data traffic [7]. In [6], the best predictor was derived to predict the future value of a discrete time fractional Gaussian noise process.

\[
\hat{X}_{t+1} = R_j^T \Gamma_j^{-1} X_j + (1 - R_j^T \Gamma_j^{-1}) m,
\]

(4)

where \( R_j = (r_1, r_2, \ldots, r_j)^T \) is a vector with values of the autocorrelation function, \( \Gamma_j \) is the correlation matrix, and \( I \) is an \( I \times I \) identity matrix. \( r_0 = \cdots = r_I = 1 \). Vector is equal to 1.

\[
\hat{\sigma}^2 = r_0 - R_j^T \Gamma_j^{-1} R_j.
\]

(5)

It is well known that \( \hat{X}_{t+1} \) is still Gaussian.

In the scenario of wireless mobile data networks, such as mobile Ad hoc, the number of active sources sharing the capacity is quite small and they have non-stationary features.
In order to still use model like (2), we must give the current mean estimation and use the model in a semi-stationary way. The local mean is estimated by a sliding window estimator.

\[
\hat{m} = \frac{1}{N} \sum_{n=N}^{n} X_{n-1} .
\]  

(6)

Based on the estimation of traffic in next time window, the expectation service rate can be derived as follows:

\[
C_{new} = \frac{Q(l) + \hat{X}_{n+1}}{T_{win}} .
\]  

(7)

IV. PACKET TRANSMISSION TIME ADJUSTMENT

There are two kinds of prediction errors, underestimation and overestimation. With fixed modulation level in one prediction interval, the underestimation will increase the backlog in the buffer while overestimation will make the transmitter idle sometimes and the opportunity of saving energy is used insufficiently.

In order to solve these problems, a feedback loop control scheme is designed to schedule the packets transmission within a prediction interval. It is conducted in the scale of packet by packet.

The backlog in the buffer can be considered as an indicator of the current prediction error. If the backlog is small, this means the serve rate is higher than that of traffic arrival, and the delay that a packet experiences is also small. So we can stretch the transmission time of the packets. If the backlog accumulated in the buffer becomes much more, that means that the underestimation has occurred. Then we decrease the transmission time per packet slightly. Furthermore, the knowledge of underestimation error can be used to determine the modulation level of next prediction interval. So the left underestimation, which has not been eliminated by adjusting transmission time packet by packet, will be corrected on the fly according to Eq. (7).

In [1], the online Lazy scheduling scheme determines the transmission time of a packet as follows:

\[
T(l) = \alpha \frac{1 + \frac{1}{2} - \frac{\pi^{2}}{6}}{\lambda_{max}} \left( \sum_{k=0}^{l \alpha} \frac{1}{k^2} \right) ,
\]  

(8)

where \(l\) is the number of packets in buffer (exclude the outgoing packet), \(\lambda_{max}\) is the worst-case packet arrival rate, and \(\alpha\) is a factor. In order to keep the stability of the scheduling scheme, \(\alpha\) should be less than 1.

This algorithm is premised on the assumption that packets arrival follows Poisson process. It is noted that the Lazy algorithm was not necessarily optimal in [1]. In fact, according to Eq. (8), \(T(l)\) is maximized when \(l=0\),

\[
\max(T(l)) = \alpha \frac{1 + \frac{1}{2} - \frac{\pi^{2}}{6}}{\lambda_{max}} \approx 1.645 \frac{\alpha}{\lambda_{max}} .
\]  

This means that if \(1/2>1.645\alpha/\lambda_{max}\), the Lazy algorithm cannot work well. For VBR video traffic, peak-to-mean ratios usually exceed this limitation.

To design multiscale load adaptive scheduling scheme, we do not use the online algorithm as (8). Actually, any algorithm is stable only if it such that \(T(l)<1/\lambda_{max}\) for all \(l\) large enough. The difference is how fast and when it arrives at the stable region. Considering the complexity and the implementation, we choose the time stretch algorithm as follows:

\[
T(l) = \alpha(1 + \frac{\beta}{l})T_{E} ,
\]  

(9)

where \(T_{E}=P/C_{avg}\) is the predicted average transmission time of the packets in next time window \((C_{avg} \) is determined by Eq. (7), \(l\) is the number of packets in buffer(include the outgoing packet), \(\alpha\) and \(\beta\) are factors, and \(P\) is the packet size. Intuitively, \(\alpha\) relates to the utilization directly. The larger the \(\alpha\), the higher the achievable utilization, and consequently, the higher the attainable energy efficiency, but the delay may be larger. \(\beta\) is used to control how aggressive the scheduling algorithm explore the opportunity to save energy when the overestimation occurs. It also relates to the burstiness of the traffic. To achieve certain energy efficiency, the higher the burstiness is, the larger the \(\beta\) should be set.

In fact, if we set \(\beta = \sum_{j=1}^{\infty} \frac{1}{e^{\alpha j}}\) and \(T_{E} = \frac{1}{\lambda_{max}}\), Eq. (9) is very similar to Eq. (8). By carefully adjusting \(\alpha\) and using the Poisson arrival traffic in simulation, we found that the scheduling scheme with Eq. (9) can slightly improve the energy efficiency with the same average delay, and the delay jitter of scheme with Eq. (9) is also smaller than that with Eq. (8). We omit the details in this paper for lack of space.

Next we will discuss how to determine the value of parameter \(\beta\). In our multiscale load adaptive scheduling scheme, \(X_{n+1}\) is approximated as a Gaussian distribution with mean and variance given by Eq. (4) and (5) respectively. \(T(l)\) is maximized when \(l=1\), \(\max(T(l)) = \alpha(1+\beta)T_{E}\), so it can be claimed that with probability

\[
\theta = 1 - \frac{T_{win}}{\alpha(1 + \beta)T_{E}} \cdot \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\hat{x}_{n+1})^2}{2\sigma^2}} ,
\]  

(10)

the predictive error can be sufficiently corrected by our scheduling algorithm at the best situation, i.e., the traffic arrives evenly within an interval. For Gaussian traffics with different parameters, \(\theta\) is a suitable adjusting knob to control the tradeoff between delay and energy efficiency.

Given \(\theta\), we solve \(\frac{T_{win}}{\alpha(1 + \beta)T_{E}}\) from Eq. (10).

Assuming the result is

\[
\frac{T_{win}}{\alpha(1 + \beta)T_{E}} = x \cdot .
\]  

(11)

\(\beta\) can be derived as follows:

\[
\beta = \frac{T_{win}}{\alpha x T_{E}} - 1 = \frac{Q(l) + \hat{X}_{n+1}}{\alpha x} - 1 .
\]  

(12)

Assuming the maximum output link rate is \(C_{max}\), the corresponding modulation level is \(b_{max}\) bits/symbol, and the number of packets in queue is \(l\) at time \(t\). The modulation level for the outgoing packet can be derived as follows:

\[
b_{new} = \frac{C_{max}}{b_{max}} \cdot \frac{T(l)}{P} .
\]  

(13)
In practical scenarios, $b_{\text{new}}$ is limited by the maximal modulation level $b_{\text{max}}$ and minimal modulation level $b_{\text{min}}$, so $b_{\text{new}}$ can be further formulated by:

$$b_{\text{new}} = \min(\max(b_{\text{new}}, b_{\text{max}}), b_{\text{min}}).$$  \hspace{1cm} (14)

V. SIMULATION RESULTS

To evaluate the performance of our scheme, we carried out a number of simulations based on the discrete event network simulator NS [17].

A. Source Models

The aim of simulation is to investigate the energy efficiency of our scheme under the self-similar workload. Some studies have shown that the video source and the superposition of large number of ON/OFF sources with strictly alternating on and off period exhibit self-similarity [18]. So the source models used in simulation include:

- POO: this is a Pareto ON/OFF source with mean on and off time of 250 ms and Pareto shape parameter 1.9. The peak rate during the on time is 128 kbps and the packet length is assumed to be 200 bytes.
- STARWARS: this source model is a trace file produced by an MPEG-4 encoding of the Star Wars with average rate 275 kbps, packet length 200 bytes. Its peak-to-mean ratios between the corresponding modulation levels 2/8 = 0.25.

B. Wireless Link Character and Sampling Length

In simulation, we adopt the same wireless link parameter described in [4]. $Rs = 250$ kHz, $Cs = 100$ nJ, $C_s = 180$ nJ, $b_{\text{max}} = 8$ and $b_{\text{min}} = 2$. This results in a total link capacity $C_{\text{mac}} = 2$ Mbps. For $b \geq 2$, three assumptions justified in [1] are satisfied. Usually, the value of $b_{\text{new}}$ in equation (14) can only be set to even integer. In our simulation, if $b_{\text{new}}$ is not an even integer, we will transmit packet in modulation level $2i$ with probability of $b/2i+1$ and $2(i−1)$ with probability $i−b/2i$, $i = \lceil b/2 \rceil$. Then we can achieve the average modulation level $b$, and the average energy per bit is a linear interpolation between the corresponding modulation levels $2i$ and $2(i−1)$.

The number of samples is also a key factor for prediction. If it is too small, the predicted parameters cannot be estimated accurately. But with a large number of samples, more variation is expected which results in a large estimated variance. Following the guideline in [5], we choose $N = 52$ and $l = N/4$.

C. Evaluation Results

First, we evaluate the delay and the energy efficiency of Look Ahead based scheme and MLAS. Due to the convexity and monotonically decreasing property of the energy function, workload averaging results in higher energy saving. If we have infinite buffer and need not consider the delay, all bursts of the traffic can be alleviated to achieve the optimal energy efficiency. However, this is infeasible. To evaluate the energy conservation performance, the delay must be considered. So we compare the energy efficiency of different algorithms with the same average delay. We set $\theta = 0.95$, $T_{\text{win}} = 400$ ms for MLAS. $\alpha$ and Look Ahead window size will be adjusted to achieve the same average delay.

For video traffic, we choose 2, 4, and 6 video sources, which represent the light, middle and high load respectively. For POO traffic, we choose 10, 18, and 28 POO sources. The experiment results are depicted in Figure 2~5.
The figures show that our scheme achieves better energy efficiency than the Look Ahead based scheme under different workloads when the average delay is small enough. The smaller the average delay, the better the performance that our scheme achieves. Compared with Look Ahead based scheme, MLAS can save energy up to 45% with 4 video sources when the average delay is about 10 ms. The figures also show the MLAS scheme can improve the performance of energy efficiency more with video traffic than with POO traffic. In addition, as shown in Figure 2 and 5, MLAS can save about 90% energy at light traffic load Compared with the fixed modulation scheme of $b=b_{max}$, which consumes 3210 nJ per bit.

The energy efficiency and average delay of MLAS with different prediction windows are also investigated. In the simulation, we set $\alpha=0.95$, $\theta=0.95$. Figure 6 illustrates the relation between the prediction windows and energy efficiency, and Figure 7 shows the average delay varies with prediction window size. We notice that the energy curves have a sharp knee around prediction window $T=50$ms. If $T_{win}>200$ms, the energy efficiency improves slowly. Figure 7 shows that average delay has an increasing trend when prediction window is more than 50 ms large. But its increase is much slower than that of Look Ahead based scheme, in which the average delay is approximately equal to the Look Ahead window. The results mean that a prediction window longer than 200ms is therefore appropriate, and it will achieve a better tradeoff between energy efficiency and average delay.

VI. CONCLUSION AND FUTURE WORK

This paper addresses how to conserve the energy while keeping the average delay in a reasonable region over the mobile wireless network. A multiscale load adaptive scheduling or MLAS scheme is proposed, which determines the modulation level of the outgoing packet based on the short-term prediction of the traffic and the current backlog in buffer.

Simulation results show that our scheme can significantly decrease energy consumption in comparison with the fixed modulation transmission. Our scheme also outperforms the Look Ahead based scheme in terms of energy efficiency with the same average delay, especially for video sources.

We have discussed implementation of our energy efficient scheduling at an intermediate node and evaluated its performance, but there are still two topics worth further investigation. One is extending our scheme to the scenario of end-to-end and multi-service class network to support DiffServ and real time application. The other is the combination of load and channel adaptive transmission.

REFERENCES


