A Slow-sTart Exponential and Linear Algorithm for Energy Saving in Wireless Networks

Yang Song, Bogdan Ciubotaru, *Member, IEEE*, and Gabriel-Miro Muntean, *Member, IEEE*

***Abstract*— Limited battery capacity seriously restricts the usage of wireless devices in wireless communication networks. Power saving MAC layer protocols put wireless network interface into sleep mode in order to conserve energy. However, increased energy savings may sacrifice delivery performance and consequently the user perceived Quality of Service (QoS). This paper introduces Slow-sTart Exponential and Linear Algorithm (STELA), a novel battery saving mechanism which saves up to 70% energy compared with the algorithms employed by IEEE 802.11 while preventing the delivery performance from degrading significantly and consequently reducing the negative impact on user Quality of Experience (QoE).**

***Index Terms*— Energy, MAC, Networking, Quality of service**

I. INTRODUCTION

NOWADAYS wireless communications are widely employed both for daily use and professional activities. The trend is to switch from fixed devices such as desktop PCs to wireless-enabled portable ones such as Personal Digital Assistant (PDA) and Smartphone. However, mobility and wireless communication implies most devices to run on battery power, which has limited capacity and is a scarce resource. This will definitely restrict the usage of these devices at least in terms of duration between recharges with a significant negative impact on user satisfaction. In this context, techniques and algorithms which save energy for wireless devices and maximize run time between charges represent an active research area. Most mechanisms aiming to increase battery life are implemented at the lower layers and rely on periodically switching the wireless interfaces off to save power. However, there is a tradeoff between energy conservation and perceived Quality of Service (QoS).

In this paper, the Slow-sTart Exponential and Linear Algorithm for energy saving in wireless networks (STELA), a novel Medium Access Control (MAC) layer power saving scheme for wireless data transmission is proposed in order to efficiently balance energy consumption and delivery performance. The solution divides the whole process into three phases, which are slow start, exponential, and linear increase, and adjusts sleeping window of wireless interfaces dynamically according to real time traffic. Testing results have shown that this algorithm performs better in terms of energy saving while preventing severe QoS degradation.

The paper is organized as follows. Section II discusses some of the existing solution for energy saving in wireless communications. Section III presents the system architecture while in Section IV the STELA algorithm is introduced and detailed. In Section V performance testing results of STELA are compared with other two similar algorithms while the final section concludes the paper.

II. RELATED WORK

Power-saving MAC layer protocols are mainly divided into three categories based on whether the nodes are central-controlled or infrastructure free.

Contention based medium access control mechanisms are based on infrastructure free networks where a contention scheme should be employed in order to prevent collisions. IEEE 802.11 [1] saves energy by regularly waking-up the mobile nodes to sample the wireless channel, however little effort is made to avoid overhearing and idle listening. S-MAC proposed by Ye et al. [2] introduces a self-configuring algorithm to avoid contention. This protocol is energy efficient as nodes only wake up at the listening period of their cluster slot. However RTS/CTS pairs are not synchronized, consequently leading to potential collisions. Moreover it is unfair that mobile nodes with different energy levels to apply the same schedule. T-MAC proposed by Dam et al. [3] makes improvements on S-MAC and utilizes flexible duty cycle by switching off a node when there is no data for it. WiseMAC proposed by C. Enz et al. [4] solves the problem of idle listening by attaching a preamble in front of each packet header which indicates the destination.

Schedule based medium access control mechanisms on the other hand are collision free in nature, but they are not as flexible and scalable as contention based protocols. LMAC proposed by Hoesel et al. [5] splits time into slots, and assigns one slot to each node. Nodes in this situation can only transmit during their own time slots. However they should wake up for each slot to check if there is incoming data. Collision is prevented but extra energy is spent on idle listening.

Hybrid solutions such as Z-MAC proposed by Rhee et al. [6] take advantage of the two mechanisms mentioned above and

---

Y. Song is with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Ireland (phone: +353-1-700-7644; e-mail:yang.song2@mail.dcu.ie).

B. Ciubotaru is with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Ireland (phone: +353-1-700-7644; e-mail: bogdan@eeng.dcu.ie).

G.-M. Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Ireland (phone: +353-1-700-7648; e-mail: munteang@eeng.dcu.ie).

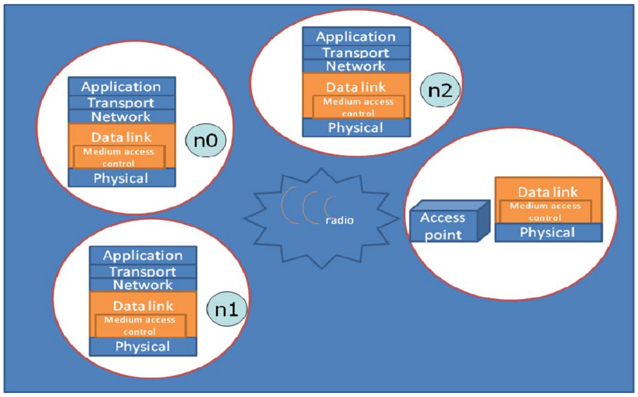

Fig. 1. Target network architecture.

combine them together. In Z-MAC, a time slot is assigned to each node similar to TDMA, but contention is also introduced when traffic load is high by allowing nodes to borrow time slots from each other for the purpose of increase bandwidth. In this way, traffic channel is fully utilized but energy could be wasted during collisions.

III. ARCHITECTURE

MAC layer lies beneath network layer and above physical layer, as can be seen in Fig. 1. In order to simplify the simulation process and focus on the energy consumption and performance, the testing model does not incorporate MAC protocol in detail. For example, node mobility and contention scheme are not included. The simulation scenario consists of a client node connecting to a server node through a wireless access point, as illustrated in Fig. 2.

To be more specific, n0 represents the client, n1 represents the access point, and n2 is the server. The duplex link between the access point and the server experiences delay which is introduced as the time between the moment when the packet is sent out from server and the time it arrives at the destination.

From the perspective of a functional block diagram, the algorithm can be presented based on the illustration in Fig. 3. Both client and server take input parameters from the user interface, and data is transmitted back and forth between the two sides. On the client side, a timer is used to count the sleep intervals in terms of beacons. QoS and energy levels are monitored and analyzed for performance evaluation purposes.

IV. ALGORITHM

The STELA mechanism consists of three phases: slow start, exponential and linear increase of sleeping window. The first phase starts when a node receives one packet from the access point or a request to send data is made by the node, and it ends when no packet is detected during one beacon interval. During this stage, the sleeping window will be kept at one beacon period. The next phase is exponential increase of listening window, which will double the sleeping window every time the wireless interface wakes up unless the wireless node receives packets and goes back to slow start phase. During this phase of the algorithm, sleeping window grows fast until it reaches a predetermined threshold value. Finally the linear mode is introduced after the threshold value is surpassed. In this mode, the listening period will increase one at each time the node wakes up until the specified maximum value.

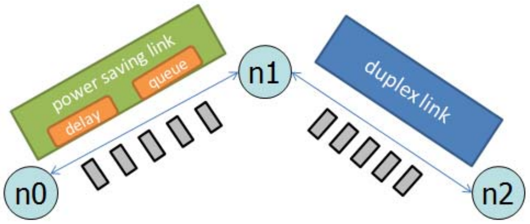

Fig. 2. Testing scenario.

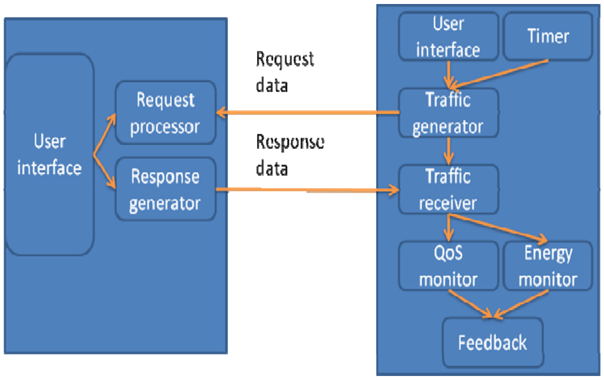

Fig. 3. Block level architecture.

According to the nature of HTTP traffic, response packets from server will only arrive and probably will arrive soon after a web request is made. Thus the slow start phase will lead to a quick response to arriving packets transmitted via the access point. Moreover, response packets are likely to form a continuous stream which can be processed by the client node at one time as the sleeping window does not grow when one packet is received. If an expected packet does not arrive in time, the second phase gets started with sleeping window growing exponentially. In this manner, energy is not wasted even for other traffic patterns like CBR where packets are transmitted regularly at constant bit rate. To be more specific, when a packet arrives, slow start process is carried on, and exponential increase will begin right after no more packets are received within one beacon period.

Normally, a specified maximum value for sleeping window is set such as in exponential increase function. However, a small value would lead to frequent waking up, while a big value will lead to long packet delay. The main reason is that predetermining the threshold does not take into account the real time traffic. In STELA, a threshold is only set for the exponential increase function. When the threshold is reached, the sleeping window keeps increasing by one each time the node wakes up in order to avoid a fast growing pattern. Thus if there is little traffic, wireless nodes can sleep longer, and they can go back to slow start phase quickly if a packet is arriving, as linear increase will lead to slow growth of sleeping duration. Above all, three phases of increase of sleeping window will generally save power without increasing packet and the

TABLE I.
TEST CASES

Case 1	Traffic type	HTTP
	Threshold value	2
	Delay between server and AP	5ms
	Beacon period	100ms
Case 2	Traffic type	HTTP
	Threshold value	4
	Delay between server and AP	5ms
	Beacon period	100ms
Case 3	Traffic type	HTTP
	Threshold value	8
	Delay between server and AP	5ms
	Beacon period	100ms
Case 4	Traffic type	HTTP
	Threshold value	4
	Delay between server and AP	40ms
	Beacon period	30ms
Case 5	Traffic type	CBR
	Threshold value	4
	Delay between server and AP	5ms
	Beacon period	20ms
	Traffic interval	2ms
Case 6	Traffic type	CBR
	Threshold value	4
	Delay between server and AP	5ms
	Beacon period	20ms
	Traffic interval	0.2ms

algorithm will go back to the first stage anytime when a new packet is received by the wireless node.

Three states are used to represent different energy consumption situations. ***Sleep mode***, which represents the state when a node has its radio turned off, and does not communicate with the environment. No packets are sent or received. Energy consumption during this status is low. ***Active mode***, which indicates that a node is having its radio turned on, and it is interacting with other nodes or listening to the channel, waiting for packets to arrive. Nodes will consume more energy in this state. ***Transition mode***, when a node switches between sleep mode and listening mode. It will take a short time, but energy consumption is relatively high.

Energy consumption in each node can be generally described as presented in Equation (1), where E implies the total energy consumption, T_s and E_s represent total sleeping time and energy consumption per time unit when a node is in sleeping mode. T_a and E_a are the total time spent in active mode and energy consumption during this period, and T_t is the time spent in transition between sleeping and listening mode, whereas $\mathrm{E_t}$ represents the energy consumption in transition mode per second.

$$\mathrm{E = T_s * E_s + T_a * E_a + T_t * E_t} \tag{1}$$

In this equation, the time spent in sleeping and listening mode is recorded by the node itself, while the time spent in transition between modes is calculated as the number of transitions from sleeping mode to active mode multiplied by the average time spent in transition. The power usage for a time unit during sleeping period, listening period and transition procedure is set according to data collected using the model proposed by Ronny Krashinsky and Hri Balakrishnan's [7].

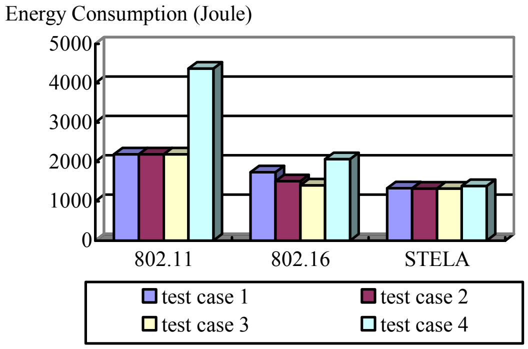

Fig. 4. Energy consumption for test case 1, 2, 3, 4

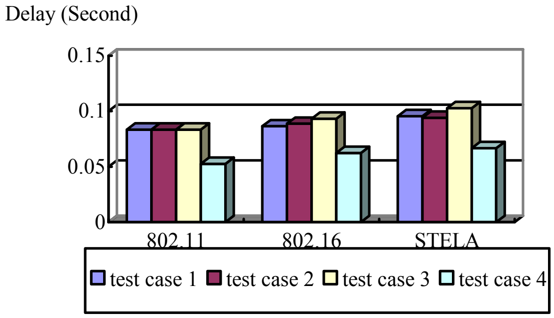

Fig. 5. Average delay for test case 1, 2, 3, 4

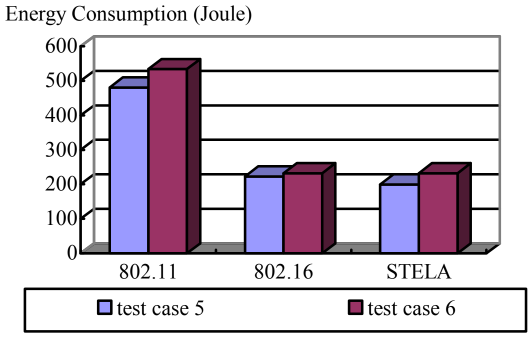

Fig. 6. Energy consumption for test case 5, 6

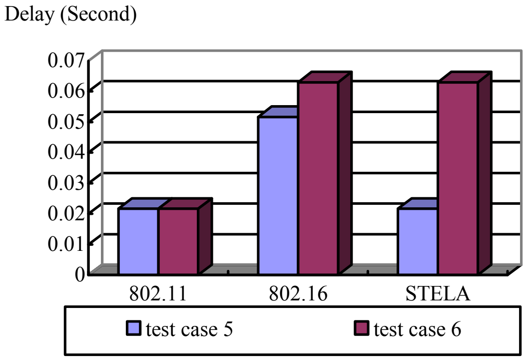

Fig. 7. Average delay for test case 5, 6

V. PERFORMANCE EVALUATION

STELA was extensively tested based on simulations performed using NS-2 [8] under both HTTP and CBR traffic load.

TABLE II
TESTING RESULT FOR TEST CASES 1- 6

Test cases	Algorithm	Energy consumption		Average delay		Packet loss rate	
		KJ	Compared with fixed function	ms	Compared with fixed function	%	Compared with fixed function
Case 1	STELA	1.33	60.4%	95.5	114.5%	0.0041	100%
	Exp.	1.74	79.2%	86.4	103.6%	0.0041	100%
	Fixed	2.20	100%	83.4	100%	0.0041	100%
Case 2	STELA	1.32	60.3%	94.1	112.8%	0.0041	100%
	Exp.	1.51	68.9%	88.7	106.4%	0.0041	100%
	Fixed	2.19	100%	83. 4	100%	0.0041	100%
Case 3	STELA	1.32	60.2%	102.8	123.3%	0.0040	97.6%
	Exp.	1.40	63.9%	93.1	111.6	0.0041	100%
	Fixed	2.19	100%	83.4	100%	0.0041	100%
Case 4	STELA	1.39	31.7%	66.7	126.8%	0.0046	92%
	Exp.	2.07	47.4%	62.2	118.3%	0.0049	98%
	Fixed	4.37	100%	52.6	100%	0.0005	100%
Case 5	STELA	0.20	41.5%	21.7	100%	0	100%
	Exp.	0.22	46.4%	51.7	238.3%	0	100%
	Fixed	0.48	100%	21.7	100%	0	100%
Case 6	STELA	0.23	43.5%	63.0	290.3%	0	100%
	Exp.	0.23	43.5%	63.0	290.3%	0	100%
	Fixed	0.53	100%	21.7	100%	0	100%

For both traffic types, different parameters are chosen:

1) Threshold values for listening window in exponential function. This parameter specifies when the exponential increase process should end and the linear function should start. The value was chosen between 1 and 32.
2) One-way delay between client and server. This value will directly influence the energy consumption level of the wireless node, as the longer it takes to transfer the packet, the more likely is the node to find no packets arriving in the slow start phase, and to start the exponential increase process. Besides this, packet delay will increase as well.
3) Beacon period combined with one-way delay will determine the amount of energy saved.
4) Traffic interval is only used for UDP traffic, and it specifies the interval between continuous packets.

Several values are generated in order to compare different solutions:

1) Energy consumption in Joule. It is the total energy consumption of a wireless node during simulation process.
2) Average packet delay. It indicates in general how long it takes for a packet to reach its destination.
3) Packet loss. This value shows how many packets are lost during transmission due to limited buffer space.

The testing results are compared with fixed listening window function such as the one introduced in IEEE 802.11 and binary exponential increase function introduced in IEEE 802.16 [9]. Parameter values for 6 distinct cases are listed in Table 1. The first three test cases are designed for HTTP traffic, where different maximum window sizes are designated. After that, delay between access point and server is set differently to demonstrate the energy consumption level and quality of service. Finally two more test cases are simulated and compared using CBR traffic. The HTTP traffic is generated with random parameters according to [7] and it is based on the code contributed to NS2 by Giao Nguyen. The transmission rate for CBR traffic is set to 0.2Mb/s and packet size is set to 1000 bytes.

Testing results are listed in Table 2, and for both traffic patterns, when using STELA energy is saved while average delay and packet loss are kept at low levels. For example, simulation results for test case 1 shows that STELA saves up to 40% of energy compared with fixed function while only generate less than 15% of packet delay. And in test case 5 with CBR as background traffic, approximately 60% of energy is conserved when using STELA compared with the fixed function; no extra average delay is introduced. When compared with the exponential function, it is also obvious that energy consumption is reduced by STELA with even shorter average packet delay.

Fig. 4 and Fig. 5 demonstrate the advantage of STELA over other two algorithms in terms of energy consumption and average delay respectively. It can be seen that STELA saves up to 70% energy in the best case compared with IEEE 802.11 with insignificant delay both in terms of percentage and absolute value. It can also be seen from Fig. 6 and Fig. 7, which outlines comparison results when CBR traffic is used, that STELA significantly improves energy conservation with little negative effects on delivery performance.

VI. CONCLUSION

In this work, a novel MAC layer algorithm, STELA is proposed which efficiently balance power consumption and QoS for mobile devices using wireless communication technologies. The key feature of this algorithm is the adaptation to real time traffic in order to reduce energy consumption with communication. The algorithm is fully tested and compared with other two widely deployed mechanisms. Testing results show up to 70% energy saving when compared with IEEE 802.11 and 20% when compared with IEEE 802.16 while harmful effect on delivery performance is minimized when using the proposed scheme.

ACKNOWLEDGMENT

The support of the Irish Research Council for Science, Engineering and Technology and Everseen Ltd. is gratefully acknowledged.

REFERENCES

[1] IEEE Std 802.11-1999, *Wireless LAN Medium Access Control and Physical Layer Specifications,* IEEE Std 802.11, 1999.

[2] W. Ye, J. Heidemann and D. Estrin, “Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks”, *IEEE/ACM Transactions on Networking*, vol. 12, issue: 3, pp. 493 - 506, June 2004.

[3] T. van Dm and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks”, *Proc. of ACM Conference on Embedded Networked Sensor Systems*, pp. 171 - 180, Los Angeles, California, USA, November 2003.

[4] C. C. Enz, A. El-Hoiydi, J-D. Decotignie and V. Peiris, “WiseNET: An Ultralow-Power Wireless Sensor Network Solution”, *IEEE Computer*, vol. 37, issue: 8, August 2004

[5] L.F.W van Hoesel and P.J.M Havinga, “A Lightweight Medium Access Protocl(LMAC) for Wireless Sensor Networks”, *Workshop on Networked Sensing Systems (INSS 2004)*, Tokyo, Japan, June 2004..

[6] I.Rhee, A.Warrier, M.Aia, J. Min, G. Subramaniam and P. Patel, “Z-MAC: A Hybrid MAC for Wireless Sensor Networks”, *IEEE/ACM Transactions on Networking*, vol. 16, issue: 3, pp. 511 – 524, June 2008.

[7] R. Krashinsky and H. Balakrishnan, “Minimizing Energy for Wireless Web Access with Bounded Slowdown,” *Proc. ACM MobiCom*, pp. 119-130, 2002.

[8] The Network Simulator- ns-2, http://www.isi.edu/nsnam/ns/.

[9] IEEE Standard for Local and Metropolitan Area Networks-- *Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE standard 802.16, 2001.