INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2020, VOL. 66, NO. 2, PP. 339–345
Manuscript received September 13, 2019; revised April, 2020. DOI: 10.24425/ijet.2020.131883

Energy-Aware WiFi Network Selection via Forecasting Energy Consumption

Atef Abdrabou*, Mohamed Darwish, Ahmed Dalao, Mohammed AlKaabi, and Mahmoud Abutaqiya

***Abstract*—Covering a wide area by a large number of WiFi networks is anticipated to become very popular with Internet-of-things (IoT) and initiatives such as smart cities. Such network configuration is normally realized through deploying a large number of access points (APs) with overlapped coverage. However, the imbalanced traffic load distribution among different APs affects the energy consumption of a WiFi device if it is associated to a loaded AP. This research work aims at predicting the communication-related energy that shall be consumed by a WiFi device if it transferred some amount of data through a certain selected AP. In this paper, a forecast of the energy consumption is proposed to be obtained using an algorithm that is supported by a mathematical model. Consequently, the proposed algorithm can automatically select the best WiFi network (best AP) that the WiFi device can connect to in order to minimize energy consumption. The proposed algorithm is experimentally validated in a realistic lab setting. The observed performance indicates that the algorithm can provide an accurate forecast to the energy that shall be consumed by a WiFi transceiver in sending some amount of data via a specific AP.**

***Keywords*—Energy, consumption, forecast, WiFi, IoT**

I. Introduction

CURRENTLY, human life becomes so dependent on wireless-enabled devices, which are mostly battery-powered. Laptops, tablets, smartphones, wireless sensors/actuators are all examples of devices that become an essential part of our today's life. Admittedly, a multitude of these devices are capable of running communication-intensive software such as multimedia, video surveillance, e-health applications, and many others. The growth of using such applications stems from the increasing capabilities of the hardware of these devices that come in a small form factor with an affordable cost. This, in turn, is anticipated to increase the amount of CO_2 emissions generated worldwide by telecommunication industry to 4% of the total emissions by 2020 [1]. Meanwhile, the progress in battery technology is somewhat limited compared with other technologies such as telecommunications and semiconductors. In fact, the progress in battery capacity development does not exceed 5% each year [2]. This difference in technology advancement results in a gap between the required energy to be consumed and the actual supply as indicated in [3] and this gap yearly increases by 4% [4]. Moreover, the current expected lifetime of rechargeable batteries is approximately 2 to 3 years. This results in 25000 tons of disposed batteries per year, which triggers environmental and financial concerns [5].

On the other hand, the simplicity, the popularity, and cost effectiveness make a significantly large number of devices depend on a wireless access that is mainly provided by WiFi networks in order to realize the realm of IoT. For instance, recent initiatives such as smart city relies to a large extent on using the WiFi technology in order to provide a city-wide wireless access for IoT applications. Moreover, traffic offloading from cellular to WiFi networks is currently under investigation of many researchers [6] as it can save considerable amount of energy for cellular network devices [7]. Furthermore, many research works, such as in [8], address the usage of cloud computing to offload the data processing from mobile devices to the servers of the cloud via WiFi networks in order to reduce the number of battery charge/discharge cycles. This, in turn, reduces battery disposal rate. However, the offloading decision has to be made based on a careful comparison between the estimated amount of consumed energy in local data processing and a forecast of the transmission/reception energy required to transfer the data for processing to/from the cloud via a WiFi network.

Apparently, the aforementioned applications require a ubiquitous WiFi access, which is mostly attained by deploying a large number of APs (predominantly connected via a wired backbone) with overlapped coverage as shown in Figure 1. However, due to the imbalanced traffic load distribution among these APs, transmitting a bulk amount of traffic over a loaded AP leads to a significant increase in the energy consumption of WiFi-enabled devices.

Thus, the objective of this research is to provide a forecast for the energy that shall be consumed by the transceivers of WiFi-enabled devices if they transmit some amount of data through a specific AP. Consequently, we propose an algorithm that allows a WiFi device, covered by multiple WiFi networks (multiple APs), to select the best network in terms of energy consumption. The proposed algorithm takes into account the interference of the other nearby networks and the devices competing to access the network assuming the network status does not significantly change during data transmission.

The contributions of this paper are two-fold. First, an analytical model is devised. The model employs the channel utilization data obtained by the transceiver of a WiFi-enabled

This work was supported by the SURE Plus research grant number 31N293 from the UAE University.

A. Abdrabou is with the Electrical Engineering Department, UAE University, Al-Ain, UAE, PO 15551 (e-mail: atef.abdrabou@uaeu.ac.ae).

M. Darwish, A. Dalao, M. AlKaabi, and M. Abutaqiya are also with the Electrical Engineering Department, UAE University, Al-Ain.

*Corresponding author.

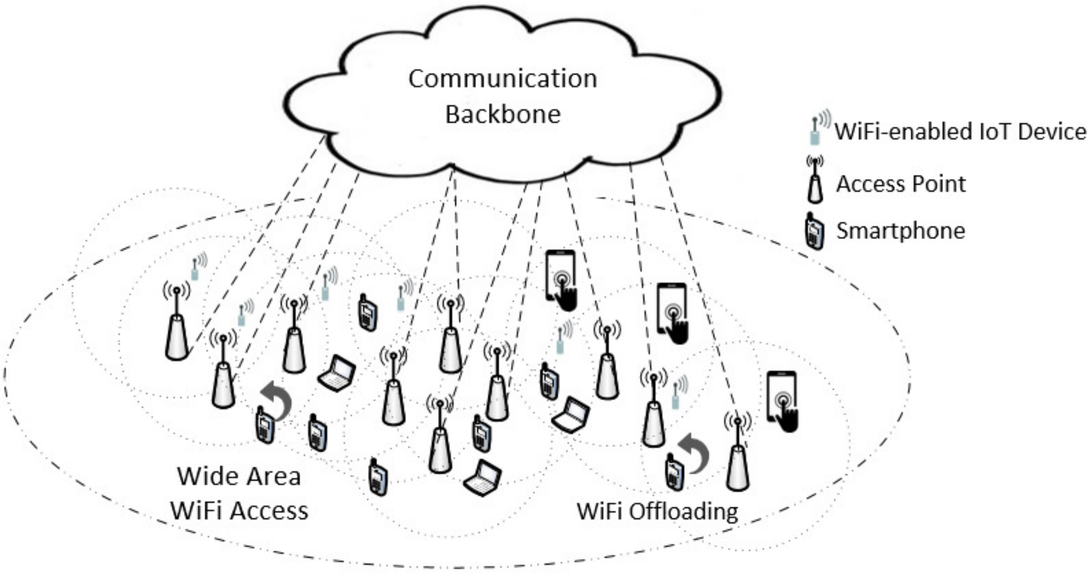


Fig. 1. WiFi wide-area coverage.

device in order to mathematically evaluate the data transfer time to a certain AP. Second, an energy estimation algorithm is proposed. In conjunction with the analytical model, the algorithm predicts the amount of energy that is supposed to be consumed in sending some amount of data over any WiFi network covering the WiFi-enabled device. The proposed algorithm can then determine the best network that the WiFi device can communicate to in order to minimize the energy consumption. This, generally, leads to decreasing CO_2 emissions and maximizes the battery lifetime for battery-powered devices. Furthermore, it helps a wireless device in selecting the best AP for a cellular-to-WiFi energy-aware data offloading and/or supports the decision of performing data processing locally or via cloud computing.

The rest of the paper is organized as follows. Section II briefs the most relevant research works. Section III provides the details of the proposed analytical model and energy estimation algorithm. Section IV introduces the experimental setup used to validate the proposed analytical model including the hardware and software tools. It also describes the lab setting. The experimental and analytical results are compared in Section V. Finally, Section VI concludes this paper.

II. Related Works

In the literature, a number of recent research works address WiFi energy modeling. Some of these models mainly focus on smartphones and rely on the parameters that can be available for application developers. For instance, the authors in [9] present a WiFi energy model for smartphones that is based on application layer throughput but it is only for data reception. Hao et al. [10] introduce energy estimation of smartphone applications via code analysis, which is largely hinged on the availability of the application source code. The authors in [11] and [12] offer battery-based power consumption models for different smartphone components. Their models mainly intend to provide sufficiently accurate power/energy measurements via software functions running on the smartphone without using external hardware.

Other research works in the literature address WiFi energy consumption mathematically such as the works of [13] and [14]. These work are meant to address energy efficiency performance analysis and/or optimization with no proposals for practical implementation. In [15], an energy model has been devised based on recorded driver statistics, which cannot be used to forecast energy consumption before transmission.

To the best of our knowledge, no other research work in the literature proposes an analytical-based energy forecasting algorithm for the sake of selecting the best AP that minimizes the energy consumption for data transmission. The algorithm can be employed for IoT device connectivity or by smartphone cloud/WiFi offloading applications where a wide WiFi coverage is realized by a large number of APs with overlapped coverage.

III. Mathematical Model and Algorithm Description

The mathematical model aims at estimating the amount of energy that is anticipated to be consumed when sending a certain amount of data (e.g., a file) with a specific size.

A. The Mathematical Model

The proposed analytical model estimates the amount of time that some volume of data takes to be transferred using the user datagram protocol (UDP) over a single IEEE 802.11 network. The data is assumed to be sent from a tagged node to the network AP in order to be forwarded to other network(s). Meanwhile, N other active network nodes utilize the channel by sending data frames of the same size as the ones sent by the node under study. All nodes access the channel using the RTS-CTS-Data-ACK handshaking in order to reduce the effect of the hidden-terminal problem.

For a fixed packet size, the packet transmission time T_s can be expressed as [16]

$$T_s = T_{RTS} + T_{CTS} + 3\ SIFS + T_{ACK} + T + DIFS \quad (1)$$

where T_{RTS}, T_{CTS} and T_{ACK} are the frame transmission times for the RTS, CTS and ACK frames, respectively. DIFS represents the distributed inter-frame spacing, whereas the SIFS is the short inter-frame spacing as described in the IEEE 802.11 standard [17]. The network nodes compete to access the channel using the IEEE 802.11 distributed coordination function (DCF) [17]. It allows a node to start transmitting a packet after waiting for a random backoff interval of time slots selected from a contention window with a minimum size $CW_{\min}$. According to the DCF operation [17], the contention size doubles after each packet collision.

The time for sending the packet payload can be obtained from

$$T = \frac{L_{OH} + L}{\text{Data Rate}} \quad (2)$$

where L_{OH} is the IEEE 802.11 overhead and L is the payload size in bits.

The packet collision time can be calculated using [18]

$$T_c = T_{RTS} + DIFS. \quad (3)$$

In order to simplify the mathematical evaluation of the channel utilization (before the node under study joins the network), the IEEE 802.11 channel is assumed to be homogeneously utilized by N nodes generating an equivalent background traffic load. This causes packet collisions with a probability that can be obtained by solving [16]

$$p = 1 - \left(1 - \frac{\rho}{\overline{W}(p)}\right)^{N-1}. \quad (4)$$

after substituting by the average backoff window size $\overline{W}(p)$ as

$$\overline{W}(p) \approx \frac{1 - p - p(2p)^{m_b}}{1 - 2p}\frac{CW_{\min}}{2}. \quad (5)$$

The utilization factor ρ of the queue of a network node is given as

$$\rho = \frac{\lambda}{\mu} \quad (6)$$

where λ and μ are the packet arrival rate and service rate, respectively.

Thus, the average channel service time as seen by a network node can be obtained as [16]

$$\begin{aligned} S_t(N-1, p) &= \frac{1}{\mu} \\ &= \rho(N-1)\left[T_s + \tfrac{T_c}{2}\tfrac{p}{1-p}\right] + \overline{W}(p)\delta + (T_s + \tfrac{T_c}{2}\tfrac{p}{1-p}). \end{aligned} \quad (7)$$

where δ is the slot time. In (7), three terms are presented. The first one is related to the background traffic. The second characterizes the backoff time, whereas the third models the transmission and collision time of the node under study. In fact, (7) can be viewed as the cycle time between successful packet transmissions given certain node utilization factor. For RTS-CTS-Data-ACK access scheme, the channel reaches its maximum utilization at the network saturation condition (where all the nodes have always backlogged packets) irrespective of the number of nodes for $CW_{min} = 32$ [18]. Thus, the channel utilization ratio can be obtained as

$$u = \frac{\rho(N-1)\left[T_s + \frac{T_c}{2}\frac{p}{1-p}\right] + \overline{W}(p)\delta + T_s + \frac{T_c}{2}\frac{p}{1-p}}{(N-1)\left[T_s + \frac{T_c}{2}\frac{p_s}{1-p_s}\right] + \overline{W}(p_s)\delta + T_s + \frac{T_c}{2}\frac{p_s}{1-p_s}} \quad (8)$$

where the numerator presents the cycle time for the non-saturated condition, whereas the denominator is the cycle time in the saturated case with p_s the collision probability at network saturation obtained by solving (4) with $\rho = 1$.

For the node under study, the model intends to predict the transfer time of some amount of data F_s. The number of data packets to be transmitted can be calculated as

$$N_r = \left\lceil \frac{F_s}{L} \right\rceil \quad (9)$$

where L is the packet size.

The model estimates the minimum data transfer time, which implies that the data is being sent with the highest possible packet arrival rate (i.e., as much as the service rate of the channel allows). Therefore, we assume here that the node under study is almost saturated. In [19], it has been shown that the service time of IEEE 802.11 networks near saturation can be approximately modeled as a Geometric random variable. Thus, the queuing model of the node under study in this case follows a D/Geo/1 queuing system with a sufficiently large buffer, which is case for the currently existing hardware.

The data transfer time for the node under study can be obtained according to the D/Geo/1 queuing model [20] as

$$D_T \approx \left(\frac{N_r - 1}{\mu_u \rho_u}\right) + \delta\left(\frac{1}{\lambda_{us}} - 1\right)\frac{\rho_u}{2(1-\rho_u)} + \frac{1}{\mu_u}, \quad 1 > \rho_u \geq 0.98 \quad (10)$$

where λ_{us} is the packet arrival rate per slot, μ_u is the service rate, and ρ_u is the utilization factor for the node under study (should be close to 1).

B. The Energy Estimation Algorithm

The following steps summarize the proposed algorithm that can be used to estimate the energy consumption for a certain amount of data in conjunction with the aforementioned mathematical model.

Step 1: The algorithm starts by tuning the WiFi NIC to the operating channel of a certain AP.

Step 2: Energy detection is performed by the receiver while in the idle mode. This allows the receiver to determine the time during which the sensed carrier power level exceeds the detection threshold out of some window of time. The channel utilization ratio can be calculated from this step.

Step 3: The values of N and ρ can be selected such as the utilization u (calculated using (8)) matches the utilization measured in Step 2. This can be done by assuming certain value of N and ρ then solving for p using (4) and for p_s using (4) after setting ρ equal to 1. This step can be done using a lookup table of p that is calculated at different values of N and ρ.

Step 4: The amount of time that is required to transfer arbitrary amount of data F_s can be obtained using (10) where μ_u is calculated from

$$\mu_u = \frac{1}{\rho(N-1)\left[T_s + \frac{T_c}{2}\frac{p_u}{1-p_u}\right] + \overline{W}(p_u)\delta + T_s + \frac{T_c}{2}\frac{p_u}{1-p_u}} \quad (11)$$

where p_u is the solution of

$$p_u = 1 - \left(1 - \frac{\rho}{\overline{W}(p_u)}\right)^{N-1}\left(1 - \frac{\rho_u}{\overline{W}(p_u)}\right). \quad (12)$$

Similar to Step 3, the value of p_u can be evaluated by a lookup table that contains p_u at different values of N and ρ for ρ_u close to one.

Step 5: By using the measured transceiver's power consumption values, i.e., the transmit power P_{TX}, receive power P_{RX}, and idle power P_{idle}, the average power can be calculated as in (13). P_{avg} in (13) is calculated based on the fraction of transmission, reception, and idle time with respect to the cycle time $S_t(N, p_u)$.

The energy that is expected to be consumed by the WiFi transceiver for transmitting F_s amount of data can be obtained by

$$E_t = P_{avg} D_T \quad (14)$$

where D_T can be obtained by (10).

IV. Experimental Setup

In this section, the experimental setup is introduced. First, we present the used software and hardware tools. After that, we describe how the entire setup works.

A. Software Tools

Two software tools are used. One is used to generate background traffic. The tool is called *PSping* [21]. It can generate Internet control message protocol data (ICMP) packets with variable packet size and variable inter-arrival time. Meanwhile, we use *iperf* to generate the test data and measure the file transfer throughput.

In order to monitor the wireless channel, we use the software package CommView-for-WiFi [22]. It is a network monitor and analyzer software for IEEE 802.11 a/b/g/n/ac networks. It is mainly used as a sniffer in order to monitor all the data sent over a certain WiFi channel. .

B. Hardware Equipment

In order to build the WiFi network, we use an ASUS Dark Knight router. It is a dual band router that can support IEEE 802.11 a/b/g/n. The stock firmware is changed by re-flashing the router to Tomato firmware. In fact, Tomato firmware allows the user to access more configuration details that are not available through stock firmware such as controlling the maximum data rate. The router has been configured to work on Channel 6, which is one of the commonly used radio channels in the 2.4 GHz band.

For network connectivity, three types of USB WiFi adapters are used. The Think Penguin legacy (IEEE 802.11 b/g) 2.4 GHz WiFi USB adapters are used with the PCs that generate the background traffic. Dell dual-band 802.11/a/b/g legacy USB adapters are used by the sender-receiver pair under study to send the test data. Since we are using a sniffer software to capture the traffic sent over the channel, the advanced dual-band USB adapter Alfa AWUS036ACM 802.11 a/b/g/n/ac is used. It is recommended by the sniffer software as it can sniff on any WiFi standard currently operational in any WiFi network in range.

We mimic the online measurements of channel utilization by using a software-defined radio (SDR). The SDR gives the ability to control the energy detection of the radio channel without the need to deal with a complicated open-source WiFi adapter driver software. Here, we use BladeRF. The BladeRF is a fully bus-powered USB 3.0 SDR with a small form factor. It operates in 300 MHz - 3.8 GHz radio frequency range. It can transmit and receive independently using 12-bit samples with a maximum sample rate of 40 M sample/second.

We have two kinds of traffic sources that are connected to the network. One traffic source is employed to load the network with some amount of background traffic. This traffic is created using normal Intel Core-i5 Windows-based PCs. The other traffic source generates the test data. In our experiment, this traffic source is implemented using two single-board computers (SBCs). The SBCs are Roboard RB-110, which is a complete computer system. It has a Vortex 86Dx CPU running at 1000 MHz with 256 MB RAM. Both SBCs run Linux Ubuntu server.

C. The Lab Setting

The lab setting consists of a number PCs that are connected to a WiFi router using Think-Penguin legacy Wi-Fi adapters. These PCs create constant-bit-rate background traffic over Channel 6 on the 2.4 GHz band using the PSPing tool. The packet generation rate is kept fixed at its maximum value, however, the packet size is varied in order to change the channel utilization. On the other hand, the test data is transferred between the two SBCs while the background traffic is being transmitted.

Channel monitoring is performed using the sniffer software, which gives a comprehensive output file that can be easily exported to any spreadsheet software. Using the information provided in the file such as the start time, the end time, and the frame transmission duration, the channel utilization percentage can be obtained for any window of time.

The BladeRF is employed to collect channel utilization information by the aid of a Matlab code, which uses energy detection in order to determine the channel utilization percentage over the same window of time used by the sniffer software. Since our setup only mimics the real-life WiFi network interface card (NIC), the sniffer software is used to calibrate the radio amplifier gain parameters of the BladeRF SDR to match the output channel utilization of the sniffer. In

$$P_{avg} = \frac{1}{S_t(N, p_u)} \left(\begin{array}{l} P_{TX}\left(T_{RTS} + T + \frac{RTS}{2}\frac{p_u}{1-p_u}\right) + P_{RX}\left(T_{CTS} + T_{ACK}\right) \\ + P_{idle}\left(\overline{W} + DIFS\left(1 + \frac{p_u}{2(1-p_u)}\right) + 3SIFS + \rho N\left(T_s + \frac{T_c}{2}\frac{p_u}{1-p_u}\right)\right) \end{array} \right). \tag{13}$$

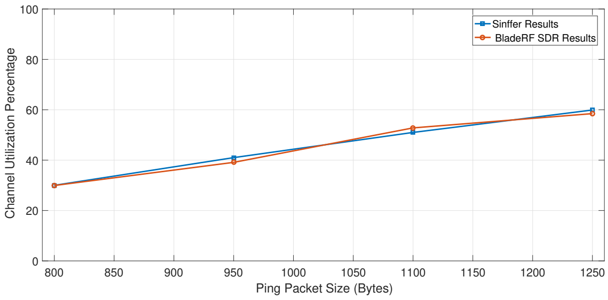


Fig. 2. The channel utilization as recorded by the sniffer and the SDR.

a real scenario, the WiFi NIC is able to determine whether the channel is busy or idle based on received the signal level. In Section V-A, a comparison between the channel utilization obtained by the sniffer and the one obtained by the BladeRF SDR is introduced.

V. Experimental and Analytical Results

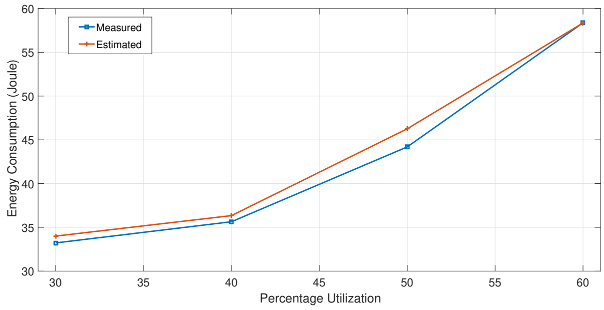
Here, we present a comparison between the analytical and the experimental results in order to show the effectiveness of the proposed algorithm.

A. Energy Detection for Channel Utilization

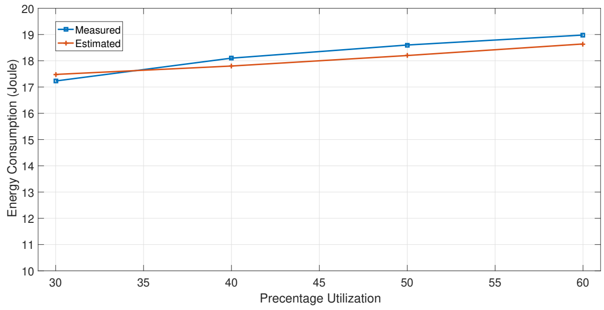
For testing the feasibility of using energy detection to determine the channel utilization, we use the sniffer software to capture all the frames sent over a specific WiFi channel. At the same time, we connect the SDR to the computer running the sniffer software and measure the percentage of time the power received by the SDR exceeds some threshold value. As mentioned in Section IV-C, we generate a background network traffic by sending ping packets from a number of PCs to the AP. In order to change the channel utilization, the generated traffic volume is varied by sending ping packets with the same rate but with different sizes. Each packet size corresponds to a different channel utilization percentage. The result of this experiment is depicted in Figure 2. Evidently, the percentage channel utilization as calculated from the sniffer software output is close to the output of the SDR as Figure 2 reveals.

B. Algorithm Validation with Fixed Channel Rate

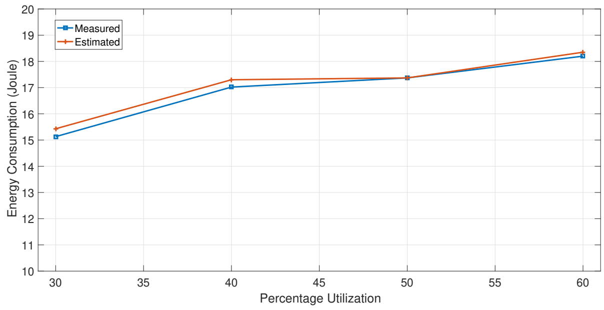
In order to validate the proposed algorithm, an experiment is conducted in a practical setting and repeated at least 20 times. In this experiment, two RB110 SBCs are connected to the same WiFi network that contains a number of PCs acting as traffic generators. One of the SBCs acts as a client and whereas the other acts as a server using the Iperf tool. The energy consumption is measured by the aid of a power supply that feeds the SBCs. The power values of the transmitting,



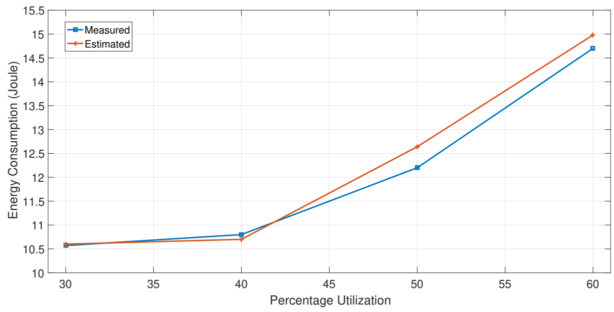
(a) Energy consumption versus utilization for 6 Mb/s channel rate.



(b) Energy consumption versus utilization for 12 Mb/s channel rate.



(c) Energy consumption versus utilization for 18 Mb/s channel rate.



(d) Energy consumption versus utilization for 24 Mb/s channel rate.

Fig. 3. Energy consumption versus channel utilization for different data rates.

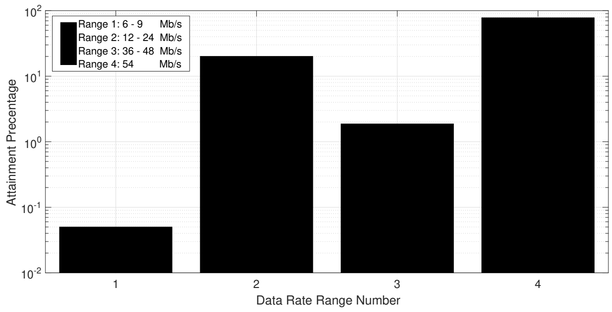


Fig. 4. The attainment history of different data rates.

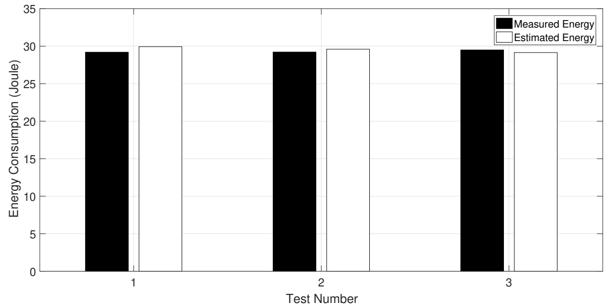


Fig. 5. Results of proposed algorithm testing without rate limitation.

the receiving mode, and the idle mode of the SBCs are also measured and recorded.

The traffic generators are set to create four different channel utilization levels, namely, 30%, 40%, 50% and 60%. Consequently, a file of 15 MB in size is transferred from one SBC to the other at a relatively low data rate ($\leq$ 24 Mb/s), which can be restricted in the configuration of the AP and the WiFi adapters. This is to guarantee that the rate is not adaptively changing from the assigned value, which normally happens in indoor scenarios with relatively short line-of-sight distance between sender and receiver. A check by using the sniffer software is performed in order to make sure that the packets of the transmitted data are transferred with the same data rate. The traffic volume generated by the Iperf tool at each utilization level is set at the highest value that does not cause any packet loss during file transfer. This process is repeated for different transmission rates (6, 12 , 18 and 24 Mbps).

Figure 3 shows the energy consumed in sending a 15 MB file using different channel transmission rates with different channel utilization percentages. The figure 3 reveals that the measured energy consumption closely matches the estimated (predicted) amount using the proposed algorithm for different data rates and different channel utilization percentages. From Figure 3, it is clear that increasing the transmission rate leads to less energy consumption since the file takes less time to be transmitted.

C. Algorithm Validation in an Adaptive Rate Scenario

For further validation of the proposed algorithm in a more realistic scenario, three tests have been conducted in a different premises. The tests are different from one another in terms of the location of the node under study, the receiving node, and the used AP. In these tests, a 60 MB is sent from the node under study to another node while other nodes are creating background traffic to achieve the same channel utilization. However, in all the three tests, no data rate limitation has been assigned to any of the nodes. Since the channel rate is varying, the proposed algorithm requires the average channel rate to be able to calculate (2) and (1). In fact, the average data rate can be obtained from the communication history of the node under study with the AP. This makes the algorithm mainly effective in case of stationary nodes (e.g., many IoT or machine-to-machine applications) or in case of limited mobility (e.g., business campuses where users slightly move in their office space). A sample of this data rate history is shown in Figure 4, which reveals the attainment percentage of different data rate ranges. The figure is generated based on collecting statistics about the data rate that the node under study used to communicate with another node via an AP.

For all the three tests, the channel utilization is measured by the SDR and used by the algorithm in order to predict the energy consumption. Consequently, the actual energy consumption is recorded and compared as in Figure 5, which depicts a good match in all tests.

VI. Conclusion

The paper offers an algorithm that forecasts the energy consumption by a WiFi-enabled device in case some amount of data is transmitted via certain AP. The algorithm relies on an IEEE 802.11 analytical model that can estimate the data transfer time by using the percentage channel utilization, which can be measured by the WiFi transceiver during idle time. The analytical model takes into account the equivalent network background traffic load generated by a number of competing nodes. This allows the WiFi-enabled device to make an accurate decision regarding the association with the AP that minimizes its energy consumption either for normal data transfer or for different offloading scenarios. The proposed algorithm is validated by hardware experimentation in a lab environment that mimics a realistic scenario. The test environment allows either a fixed or an adaptively varying channel rate. The experimental results show that the algorithm is effective in prediciting the energy that will be consumed in transmitting some amount of data through a certain WiFi network.

References

[1] M. Ismail and W. Zhuang, "Network cooperation for energy saving in green radio communications," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 76–81, October 2011.

[2] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107

[3] J. A. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, Jan 2005.

[4] S. Robinson, "Cellphone energy gap: Desperately seeking solutions," Strategy Analytics, Tech. Rep., 2009.

[5] K. Pentikousis, "In search of energy-efficient mobile networking," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 95–103, January 2010.

[6] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 627–640, April 2015.

[7] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, April 2013.

[8] M. Altamimi, A. Abdrabou, K. Naik, and A. Nayak, "Energy cost models of smartphones for task offloading to the cloud," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 384–398, Sept 2015.

[9] L. Sun, R. K. Sheshadri, W. Zheng, and D. Koutsonikolas, "Modeling wifi active power/energy consumption in smartphones," in *2014 IEEE 34th International Conference on Distributed Computing Systems*, June 2014, pp. 41–51.

[10] S. Hao, D. Li, W. G. J. Halfond, and R. Govindan, "Estimating mobile application energy consumption using program analysis," in *2013 35th International Conference on Software Engineering (ICSE)*, May 2013, pp. 92–101.

[11] L. Zhang, B. Tiwana, R. P. Dick, Z. Qian, Z. M. Mao, Z. Wang, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Oct 2010, pp. 105–114.

[12] M. Dong and L. Zhong, "Self-constructive high-rate system energy modeling for battery-powered mobile systems," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 335–348.

[13] M. M. Carvalho, C. B. Margi, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Modeling energy consumption in single-hop ieee 802.11 ad hoc networks," in *Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969)*, Oct 2004, pp. 367–372.

[14] X. Wang, J. Yin, and D. P. Agrawal, "Analysis and optimization of the energy efficiency in the 802.11 dcf," *Mobile networks and applications*, vol. 11, no. 2, pp. 279–286, 2006.

[15] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 169–180.

[16] A. Abdrabou and W. Zhuang, "Stochastic delay guarantees and statistical call admission control for IEEE 802.11 single-hop ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3972–3981, 2008.

[17] *IEEE Std 802.11g/D1.1*, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 2001.

[18] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[19] A. Abdrabou and W. Zhuang, "Service time approximation in ieee 802.11 single-hop ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 305–313, 2008.

[20] M. Xie and M. Haenggi, "Towards an end-to-end delay analysis of wireless multihop networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 849 – 861, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S157087050800111X

[21] *PsPing v2.1*, Microsoft, Jun. 2016. [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/psping

[22] *CommView for WiFi*, Tamosoft LTD, 2018. [Online]. Available: https://www.tamos.com/products/commwifi/