IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 9, 2021, accepted October 12, 2021, date of publication October 15, 2021, date of current version October 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120348

Towards Optimizing WLANs Power Saving: Context-Aware Listen Interval

AHMED SAEED AND MARIO KOLBERG, (Senior Member, IEEE)
Department of Computing Science and Mathematics, Faculty of Natural Sciences, University of Stirling, Stirling FK9 4LA, U.K.

Corresponding author: Ahmed Saeed (ahmed.saeed@stir.ac.uk)

ABSTRACT Despite the rapid growth of Wireless Local Area Networks (WLANs), the energy consumption caused by wireless communication remains a significant factor in reducing the battery life of power-constrained wireless devices. To reduce the energy consumption, static and adaptive power saving mechanisms have been deployed in WLANs. However, some inherent drawbacks and limitations remain. We have developed the concept of Context-Aware Listen Interval (CALI), in which the wireless network interface, with the aid of a Machine Learning (ML) classification model, sleeps and awakes based on the level of network activity of each application. In this paper we develop the power saving modes of CALI. The experimental results show that CALI consumes up to 75% less power when compared to the currently deployed power saving mechanism on the latest generation of smartphones, and up to 14% less energy when compared to Pyles' *et al.* SAPSM power saving approach, which also employs an ML classifier.

INDEX TERMS 802.11, energy consumption, power save mode (PSM), NS2, wireless local area network (WLAN), Wi-Fi.

I. INTRODUCTION

IEEE 802.11 Wireless Local Area Networks (WLANs) are in pervasive deployment, and considered one of the most rapidly growing technologies in the world [1]. In an infrastructure-based WLANs, wireless devices, through a Wireless Network Interface Controller (WNIC), transfer data wirelessly with an Access Point (AP) [2]. However, energy consumption and consequently battery depletion of WLAN devices, remains one of the major issues [3], [4].

To reduce the amount of energy consumed by the WNIC, a number of power saving approaches have been devised including Static Power Save Mode (SPSM) [5], Adaptive PSM (APSM) [9], and Smart Adaptive PSM (SAPSM) [10]. However, several limitations with these approaches have been reported in the literature [6]–[11].

Unlike other power saving approaches, we have developed the concept of a Context-Aware Listen Interval (CALI), where the WNIC, with the aid of a Machine Learning (ML) classification model, sleeps and awakes based on the level of network activity of each application.

In this paper, we develop the power saving modes of CALI. CALI's power saving modes optimize the sleep and awake cycles of the WNIC in accordance with applications' network interactivity. The following contributions are reported in this paper:

- We have created four scenarios of mobile applications' network traffic based on our previously defined four CALI power saving modes: Buffering, Dynamic Listen Interval (DLI), Low, and Awake. These traffic scenarios reflect the traffic of applications with a diverse set of network behavior.
- Three sets of energy parameters reported in major previous studies have been adopted.
- The CALI power saving modes are evaluated by comparing the levels of energy consumption with existing benchmark power saving approaches, including APSM and SAPSM using the three sets of energy parameters.
- Simulation results, using the NS2 simulation engine, are reported observing CALI's energy consumption while varying individual energy parameters between the max and min used in the three parameter sets.

In our previous work [12], we have conducted a comprehensive study of classifying smartphone applications' network traffic and proposed the framework for the context-aware network traffic classification approach based on ML classifiers for optimizing WLAN power saving.

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

The remainder of this paper is organized as follows: Section II reviews related work, including deployed power saving protocols in WLANs. This is followed by a review of power saving approaches proposed in the literature. Section III describes the CALI framework, and the experimental setup employed in this study. Experimental results and their analysis are presented in Section IV. Section V draws conclusions and identifies future research directions.

II. RELATED WORK

This section reviews the deployed power saving protocols in WLANs, in particular SPSM and APSM including their comparative drawbacks. This is followed by a critical review of power saving approaches proposed in the scientific literature.

A. REVIEW OF KEY DEPLOYED POWER SAVING PROTOCOLS

1) STATIC PSM

In the WLAN Infrastructure Basic Service Set (IBSS), the 802.11 standard defines SPSM to reduce the amount of energy consumed by the WNIC when the wireless devices are connected to an AP.

The WNIC of a wireless device in SPSM operates in two modes: awake mode and sleep mode. In the awake mode, the radio transceiver of a wireless device is on and ready to receive and transmit consuming significant amount of power. While in sleep mode, the transceiver is off, meaning that the wireless device cannot receive or transmit in order to conserve power [5].

In SPSM, the AP announces the presence of any buffered packets intended to a wireless device via a Traffic Indication Map (TIM) in a beacon frame. Thus, the wireless device stays in sleep mode and periodically wakes up during its listening interval (multiples of the beacon interval) to listen to the TIM in the beacon frame. If the TIM does not indicate packets for the wireless device at AP, the wireless device immediately goes back into sleep mode to save power.

In the case a TIM indicates the existence of buffered packets at AP, the wireless device remains awake and generates the Power Save Poll (PS-Poll) frames to retrieve the buffered packets from the AP. Upon receiving PS-Poll frames, the AP transmits the buffered packets to the wireless device, one packet at a time and receives its corresponding acknowledgment until all buffered packets are received successfully and the AP finally indicating the existence of no more packets by setting the value of the More Data field to zero [13], [14].

The SPSM conserves energy by allowing a wireless device to sleep and waking up periodically. Nevertheless, SPSM suffers from latency issues for the following two reasons: firstly, when a wireless device generates the PS-Poll frames in order to retrieve the buffered packets one at a time from AP [6], [7]. Secondly, when a delay of 100-300ms is introduced when the WNIC is off during the beacon intervals and there are incoming packets for a wireless device buffered at AP [8]. These issues impact on the performance of both, real-time applications such as VoIP and interactive applications such as web browsers.

2) ADAPTIVE PSM

APSM has been deployed within the most recent generation of mobile devices to overcome the delay of the WNIC being off during the beacon interval and the delay caused by the PS-Poll frames. In APSM, a wireless device adaptively switches between sleep and awake mode depending on the network traffic [9]. In APSM, by default, a wireless device remains in SPSM [15]. To switch from SPSM mode to the awake mode, the wireless device notifies the AP by sending a null data frame with the power management bit set to zero. When the AP receives the null frame, it stops buffering packets for the wireless device. To switch back into SPSM mode, the wireless device sends a null data frame with the power management bit set to one, so that the AP resumes buffering packets for the wireless device [7], [16].

APSM operates based on a threshold, i.e., ingress and egress packets between a timer start and expiry are counted. If the counted packets exceed the threshold, the WNIC switches to the awake mode. On the other hand, if the counted packets are below the threshold, the WNIC remains in SPSM mode [10].

Latency related issues found in SPSM are eliminated in APSM. However, the WNIC of a wireless device does not take into consideration the type or the importance of network traffic, it switches between two modes based on network activity thresholds alone. This may lead to the WNIC being switched into awake mode unnecessarily, receiving low priority traffic consuming energy which could be better used for more important traffic [7], [10]. Moreover, the WNIC remains in awake mode for an idle timeout period before being fully switching back to SPSM [11].

B. STATE-OF-THE-ART LITERATURE REVIEW

This subsection reviews a number of key power saving approaches proposed in the literature.

In [17] He and Yuan propose a time division multiple access approach based on MAC protocol, called scheduled PSM. In this approach, the beacon interval is divided into an equal number of slices by an AP. The slices can be assigned to a single wireless device or multiple wireless devices. The TIM was restructured to hold slice assignment information. Scheduled PSM eliminates channel contention, as each wireless device wakes up on its designated time slot to retrieve the buffered data from the AP, and sleeps during its non-allocated time slots to save power. This approach conserves energy as the channel is contention free, but time slots will be wasted if a wireless device does not wake up at its designated time slot. Also, this approach suffers from additional delay: data frames arriving at the current beacon interval will only be scheduled for transmission to a wireless device in the next beacon interval. Finally, all the time slots are identical in size, which may not be appropriate for small frames or light traffic.

Opportunistic Power Saving Mode (OPSM) is proposed in [18]. The application of OPSM is limited to a specific scenario: wireless devices are engaged in web browsing to download short files with a short duration of inactivity or think time in between downloads. The authors of [18] observed that the throughput share of an individual wireless device decreases in SPSM when multiple wireless devices are associated with a single AP and download files simultaneously. Therefore, to gain the maximum throughput and reduce energy consumption, only one wireless device is permitted to download a file at a time in OPSM. During this time other wireless devices remain in sleep mode. One additional bit has been added to the beacon header indicating whether the AP is currently serving another wireless device. To avoid a number of wireless devices from initiating a file download simultaneously on completion of the service of the current wireless device, wireless devices wait for a random period of time before initiating their file download.

Rozner *et al.* [7] introduced a Network Assistant Power Management solution (NAPman). The authors conducted a variety of experiments to show that current implementations of PSM strategies in wireless devices and APs are not efficient due to competing background traffic which increases the energy consumption of a wireless device and decreases the network capacity due to unnecessary retransmissions. To mitigate these issues, NAPman employs virtualization and an energy-aware scheduling algorithm for AP based on the First Come First Serve (FCFS) policy that applies only to packets of wireless devices that are awake at a given time. By leveraging AP virtualization, contention among wireless devices is mitigated, as several virtual APs from one physical AP are created. Each wireless device is connected to its own dedicated copy of a virtual AP. As NAPman relies on virtualization, one physical AP can only support a limited number of virtual APs. This causes disruption when the number of assigned wireless devices to virtual AP exceeds the threshold limit.

In [19] Omori *et al.* present a power saving approach that utilizes Network Allocation Vector (NAV) periods set by the Request to Send (RTS) and Clear to Send (CTS) handshake mechanism. The proposed approach allows other wireless devices to sleep when they overhear the CTS or RTS during the NAV duration. Moreover, the NAV duration is extended which allows multiple bidirectional burst transmission between a device and an AP. In their previous work [20] the authors of this approach utilized NAV duration by allowing the burst transmission in an unidirectional manner for incoming packets from AP only.

Studies [8] and [21] explore conserving power by utilizing multiple radios of wireless devices. Authors of [8] introduced Bluesaver, which employs Bluetooth and WiFi combined at an AP and wireless device. The wireless device switches between WiFi and Bluetooth radios. The wireless device receives and sends packets over Bluetooth when it is within range of the Bluetooth radio of the AP. When a higher data rate is required or a wireless device is out of range of the Bluetooth radio of the AP, it switches to WiFi radio. However, this approach requires an additional Bluetooth adaptor at the AP.

Zhang and Li [21] developed a WiFi-ZigBee message delivery scheme, which delegates some of WiFi operations to ZigBee radio. In this case, the WiFi radio of a wireless device is turned off, and instead, low power ZigBee radio is utilized to discover the presence of WiFi networks. It then listens to incoming beacon frames from the AP to detect the presence of any buffered packets intended to a wireless device. However, the developed scheme in [21] requires an external chipset on smartphones.

Other researches [22]–[24] focused on decreasing the radio's clock rate to conserve energy. SloMo [22] proposed a transceiver that enables a wireless device to operate at a lower clock rate during transmitting and receiving. E-Mili [23] allows the WNIC to operate at a lower clock rate during idle listening and transits to the full clock rate during data transmission and reception. In [24] the authors proposed Sampless WiFi, which enables the wireless device to recover under-sampled packets via multiple transmissions.

Li *et al.* [25] proposed DLI to reduce the energy consumption caused by unnecessary wakeups. In this scheme, the listen interval of a wireless device is incremented by 1 each time a wireless device wakes up during its listen interval and finds the presence of no packets buffered at AP. Moreover, a wireless device reverts its listen interval to 1 when it finds the presence of buffered packets at the AP. The proposed scheme conserves power in comparison with SPSM by adjusting longer listen intervals, but an additional delay will be added if packets of interactive applications are buffered at an AP during the increased listening interval.

Attempting to eliminate the issues related to APSM, authors of [10] proposed SAPSM, which is based on categorizing smartphone applications as either low or high priority apps using an ML classifier. Consequently, the traffic of applications, which have been tagged as high priority, switches the WNIC into awake mode. Conversely, network traffic of low priority applications keeps the WNIC in SPSM conserving energy. To train the ML classifier and set applications' priority, a study was conducted. In this study, participants interacted with a range of applications that have diverse levels of network interactions. Initially, all applications are configured with SPSM, and based on the participants' experience with the selected application, the priority of each application was determined. The priority is set to high if the observed delay by a participant is unacceptable. In contrast, it is set to low if the observed delay has not impacted the participants' experience. The Support Vector Machine (SVM) classification model that was used in the study has achieved an accuracy of 88.1%.

However, no additional priority levels or modes have been proposed for this approach to cater for applications with, for instance, very low levels of background interactivity or applications using buffer streaming. Instead, the approach operates the WNIC in SPSM for all low priority applications.

Li *et al.* [26] introduced a similar approach to SAPSM, which is also based on prioritizing smartphone applications into low and high priorities. Authors of this approach conducted measurements of smartphone applications' usage. Based on these measurement results, two features that reflect network interactivity: the receiving rate and the screen touch rate were extracted. Finally, based on these two features, a prioritization scheme that classifies applications' network traffic into low or high priorities was presented. For high priority applications the network traffic will be operating in the awake mode, and for low priority applications the network traffic will remain operating in SPSM. The proposed scheme in [26] was only evaluated against a user study. Moreover, no further priority or mode was considered for applications that are capable to operate with extended periods of WNIC listening intervals.

III. FRAMEWORK AND EXPERIMENTAL SETUP

A. FRAMEWORK

1) CONTEXT-AWARE LISTEN INTERVAL (CALI)

Unlike other power saving approaches reported in the literature, CALI proposes the concept of a context-aware listen interval, where the WNIC, with the aid of an ML classification model, sleeps and awakes based on the level of network activity of each application.

In ML, classification is defined as a learning method that maps or classifies instances to corresponding class labels which were predetermined in a given dataset. According to Han *et al.* [27] data classification is a two-step process; the first step is learning, where a classification model is built from a given dataset, the data from which a classification model is learned called a training set. The second step is the classification, where a model is used to predict class labels for previously unseen data. The dataset, which is used to test the classifying accuracy of the learned model is called the test set.

We have constructed a real-world dataset based on the network traffic of nine smartphone applications, which reflects a diverse array of network behavior and interaction. For high levels of network interactivity, both Google Hangouts and Skype audio and video calls were selected. For traffic with intermittent interactions, Gmail and Facebook were chosen. For applications with the lowest level of interactions New Star Soccer (NSS) and New Star Cricket (NSC) were considered. Network interactions of these applications mostly occur during fetching advertisements. For the network traffic that reflects applications with audio buffering capabilities, the XiiaLive internet radio application was employed.

All applications were downloaded through Google Play, including the Network Log application, which was used to capture the network traffic and to extract the set of six input features form the network activities of each application. These features are: 1- receiving data rate in Kbytes/sec, 2- transmitting data rate in Kbytes/sec, 3- total received Kbytes, 4- total transmitted Kbytes, 5- total number of received packets, and 6- total number of transmitted packets. These features are statistical-based and unique for specific types of applications. Additionally, inspection into the packet content is not required to extract these features, hence statistical features have low computational overhead and are applicable for both encrypted and unencrypted traffic [28], [29]. Moreover, these features reflect the applications' network interactivity better than non-network features like touch screen rate, as regularly touching the screen, does not always mean that network traffic is occurring. For instance, video games are highly interactive in terms of user and screen, but practically non-interactive in terms of network interaction.

We have manually labeled instances of the nine applications according to the levels of traffic interactivity in the background of each application. Fig. 1 shows the receiving data rate in Kbytes/sec of the first 50 instances which reflect varying levels of network interaction.

Fig. 2 illustrates the flowchart of CALI, where instances of real-time network traffic of each application were captured, and then manually labeled to the right output or class. We have labeled all instances of applications with a high level of interaction as high, instances of applications with an intermittent level of interaction were labeled as varied, whereas instances of applications with the lowest level of interaction were labeled as low. Finally, instances of audio streaming application with buffering capability were labeled to buffering.

After labeling the input samples of the captured traffic of each application, an ML classifier learns to map the input features of each sample to an output class from the training set, constructing an ML classification model.

The next step is the classification, where an ML classification model is used to predict class labels for previously unseen data. Test set is used to test the classifying accuracy of the learned model.

In [12] we have employed eight commonly used ML classifiers to classify the traffic of the nine applications using the WEKA data mining tool [30]. The applied ML classifiers were: Multilayer Perceptron (MLP), Naïve Bayes, Decision Tree C4.5, SVM, Bayes Net, Radial Basis Function (RBF), Random Forest, and K-nearest neighbor (KNN). To validate the accuracy of each ML classifier in mapping the inputs to the correct output class 10-fold cross-validation was employed. Results in our previous paper [12] showed that more than 99% of accuracy can be achieved with these algorithms and indicated that ML classifiers can be usefully applied for classifying smartphone applications' network traffic based on different levels of interaction.

2) CALI POWER SAVING MODES

In order to optimize the sleep and awake cycles of the WNIC in accordance with the applications' network activity, we have defined four CALI power saving modes. These power saving modes enable additional power saving opportunities and have been devised based on the classified output traffic of the captured samples from a varied range of

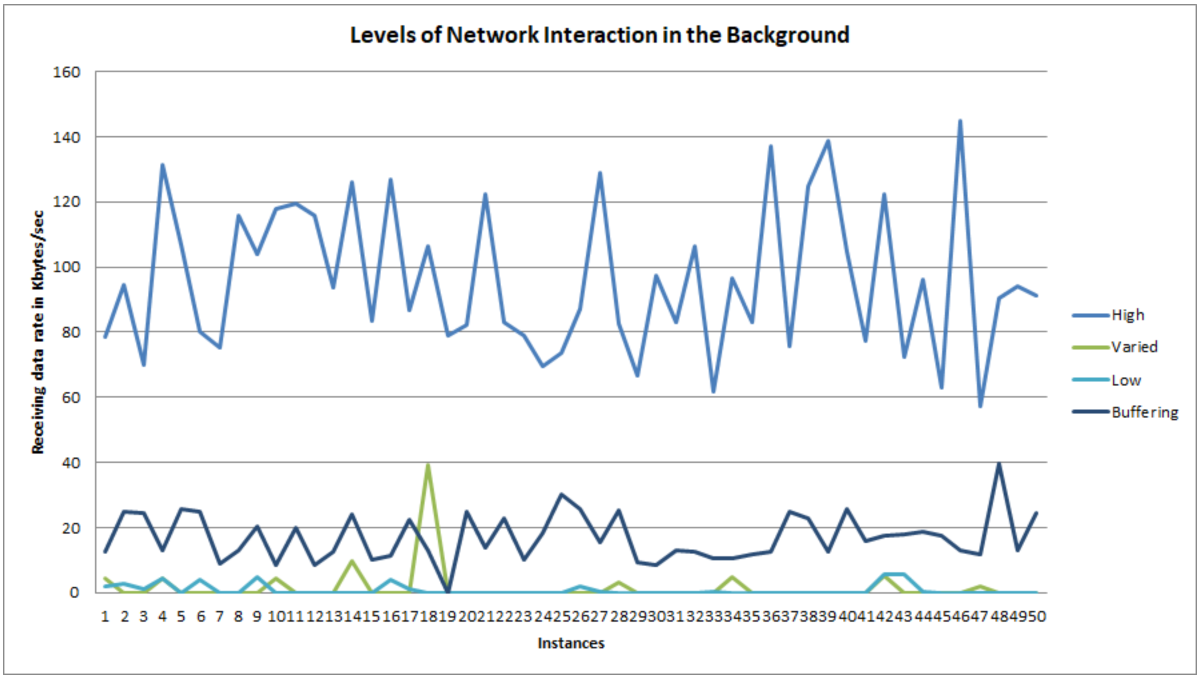


FIGURE 1. Arrays of network behaviour characterized by levels of traffic interaction.

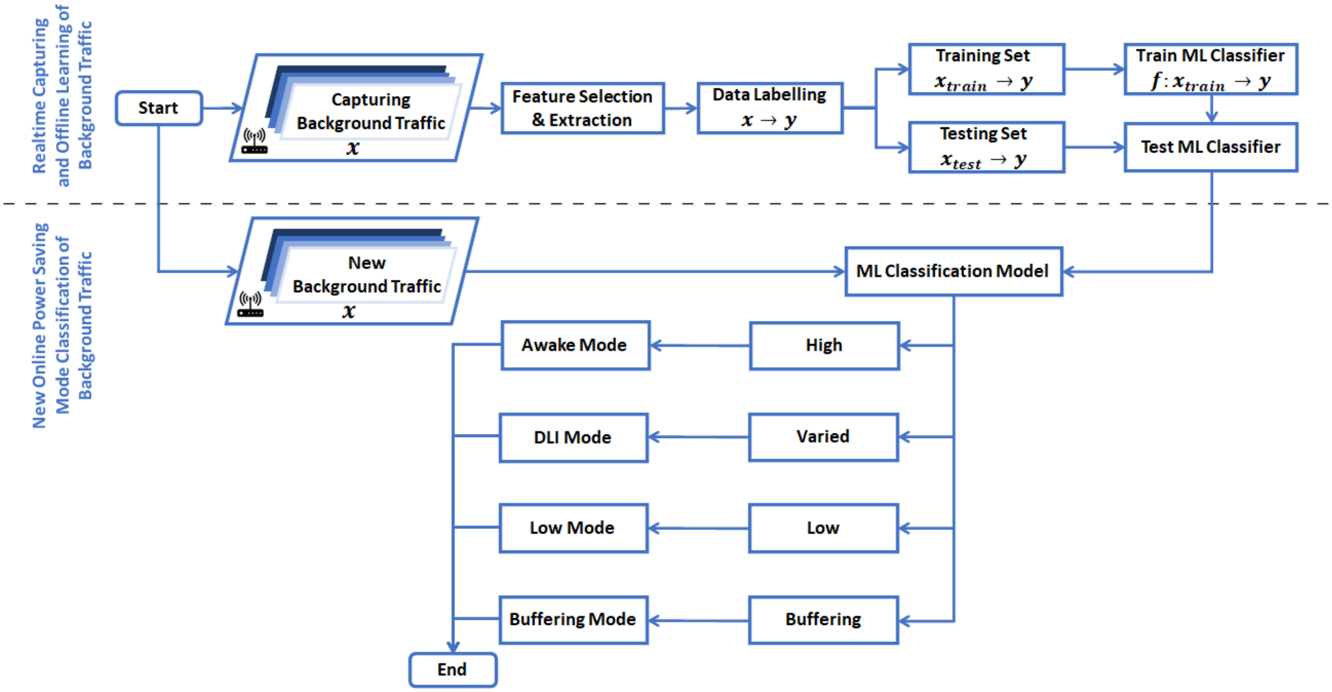

Realtime Capturing and Offline Learning of Background Traffic
Start
Capturing Background Traffic x
Feature Selection & Extraction
Data Labelling $x \rightarrow y$
Training Set $x_{train} \rightarrow y$
Testing Set $x_{test} \rightarrow y$
Train ML Classifier $f: x_{train} \rightarrow y$
Test ML Classifier
New Online Power Saving Mode Classification of Background Traffic
New Background Traffic x
ML Classification Model
High
Varied
Low
Buffering
Awake Mode
DLI Mode
Low Mode
Buffering Mode
End

FIGURE 2. Context-aware listen interval.

smartphone applications' network traffic that reflect a diverse array of network behaviour and interactions. Hence, the ML classification model classifies the new unseen samples into one of the classes, and the WNIC will be adjusted to operate into one of CALI power saving modes. Moreover, CALI handles applications, which it cannot map to one of the four modes by reverting the WNIC to operate in SPSM mode. That means, the worst possible performance is that of SPSM, but if one of the four modes applies, a significant performance improvement with respect to power saving is achieved.

a: AWAKE MODE

When the ML classification model classifies the new unseen samples of highly interactive applications to the output class high. Consequently, the WNIC is set to operate in awake mode.

b: DLI MODE

The ML classification model classifies the traffic samples of applications with varied levels of interactivity to the output class varied. The WNIC will be adjusted to operate in DLI mode. We have considered employing the DLI methodology introduced in [25]. So, the listen interval is incremented by 1 at each time a wireless device wakes up during the listen interval and finds no packets buffered at the AP. The listen interval reverts back to 1 when interactions occur. To prevent the listen interval from growing excessively we set an upper bound of $10 = 1000$ms for the listen interval. Applications such as Gmail and Facebook have intermittent network interactions and do not always receive data. Therefore, assigning the background traffic of these applications to the awake mode would not be efficient.

c: LOW MODE

The ML classification model classifies the traffic of applications with the lowest level of interactions to the output class low. Consequently, the WNIC will be switched to operate on low mode, with an extended value of the listen interval. This is beneficial as network interactions of these applications mostly occur during fetching advertisements.

d: BUFFERING MODE

The ML classification model classifies samples of audio streaming applications with buffering capability to the output class buffering. The WNIC will be set to operate in buffering mode. The buffering mode was defined for applications that allow users to stream audio over the Internet, according to [31] these applications are capable to buffer several seconds of audio stream. For such applications, switching off the WINC for short periods of time does not impact on the playback streaming quality.

B. EXPERIMENTAL SETUP

Due to a rapid revolution of wireless technology, new enhancements are required to be tested and analyzed in a rapid and cost-effective manner. Analytical modeling, real deployment, and simulation are the most commonly used methods in communication networks for evaluating the performance of a proposed system or framework [32]. Analytical methods are based on simplified models, on the other hand, real deployment is complex, costly, and time-consuming. Alternatively, simulation allows network scenarios to be easily built, modified, and analyzed. Simulation allows parameterization, in which a system could be modeled with any level of detail required [32], [33].

Consequently, for our experimentation we have used the network simulator NS2 [34]. NS2 has been widely used to measure performance parameters in wired and wireless networks. To support the power management functions in WLAN, we used the NS2 extension proposed in [35], which has been applied in several studies including [36] and [37]. This NS2 extension provides PSM mechanisms, such as the PS-Poll, AP buffer, and TIM. Furthermore, it includes an energy model which uses four energy parameters: txPower, rxPower, idlePower, and sleepPower. During the experimentation we adjusted the listen interval of CALI based on the type of application's network traffic.

To experiment with the four CALI power saving modes, we configured four corresponding traffic scenarios (Buffering, DLI, Low, and Awake) using a Tcl script.

The buffering scenario uses traffic from the XiiaLive internet radio application using a random station with a 128kbps stream.

For the DLI scenario, the traffic of 30 emails in Gmail and receiving 30 Facebook posts at random intervals was employed.

For the low scenario, NSS was run several times. We observed that the duration of one game is about 110 seconds, after that time an advertisement will be loaded.

Finally, for the awake scenario, traffic of 10min Skype video call was used.

As smartphones' applications spend longer in receiving packets than transmitting, the downlink receiving traffic has been considered in our simulation. From the dataset, we have used the following features as inputs to configure the four corresponding traffic scenarios of CALI power saving modes: 1- receiving data rates, 2- number of received bytes, and 3- number of received packets. The simulation environment is based on Ubuntu 10.04.4 LTS with a simulation duration of 600 seconds and initial energy of 1000 J.

To explore the behavior of the CALI power saving modes, three sets of energy parameters reported in major previous studies have been adopted. Each set consists of 6 energy parameters; Set 1 has been widely employed in studies including [25], [38], [39]. Set 2 reflects the energy parameters of Wavelan WNIC [5], [40], whereas Set 3 reflects the energy parameters of Intel WNIC [41], [42]. The six parameters are:

txPower: the power consumption during packet transmission.

rxPower: the power consumption during packet reception.

idlePower: the power consumption when a WNIC is awake and not transmitting or receiving packets.

transitionPower: the power consumption when a WNIC transits from the sleep to idle state and vice versa. This must be twice of idelPower [38].

transitionTime: The amount of time required when a WNIC transits from sleep to idle state and vice versa.

sleepPower: The power consumption when a WNIC is in sleep state.

The three sets of energy parameters are shown in Table 1.

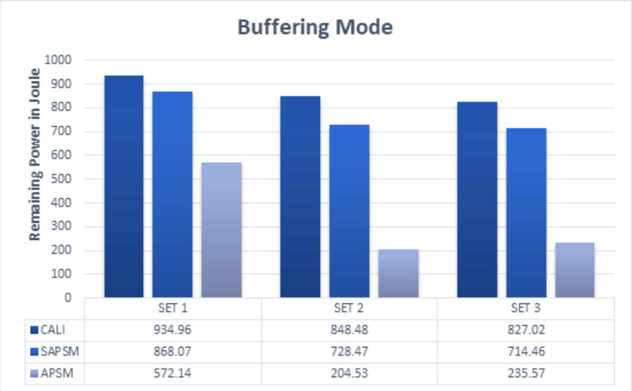
TABLE 1. Sets of energy parameters.

Parameter	Set 1 Value	Set 2 Value	Set 3 Value
txPower	1.4 W	1.675 W	1.44 W
rxPower	0.9 W	1.425 W	1.34 W
idlePower	0.7 W	1.319 W	1.27 W
transitionPower	1.4 W	2.638 W	2.54 W
transitionTime	0.002 S	0.002 S	0.002 S
sleepPower	0.06 W	0.177 W	0.22 W

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section evaluates the performance of CALI power saving modes by comparing the levels of energy consumption of CALI with existing power saving approaches. We selected APSM as the most current approach deployed in smartphones and SAPSM as a recent technique also employing ML.

Fig. 3 shows the energy consumption of CALI, SAPSM, and APSM in buffering mode for the 3 sets of energy parameters. We set the listen interval of CALI to $10 = 1000$ms. The listen interval value has been determined to not affect audio quality in several experiments with the audio streaming application XiiaLive. We found that the added delay did not impact the playback streaming quality as was also noted in [10] and [31]. For all 3 sets of energy parameters, CALI consumes less energy in comparison to SAPSM and APSM. In Set 2, CALI consumes 14.14% less energy compared to SAPSM and 75.89% when compared with APSM. For all 3 sets of energy parameters, APSM consumes more energy in comparison to SAPSM and CALI. This is due to the behavior of APSM with this type of traffic, as the WNIC remains awake and always on. When the values of rxPower and idlePower increased in Set 2, more power was consumed using APSM compared to Set 1 and Set 3.

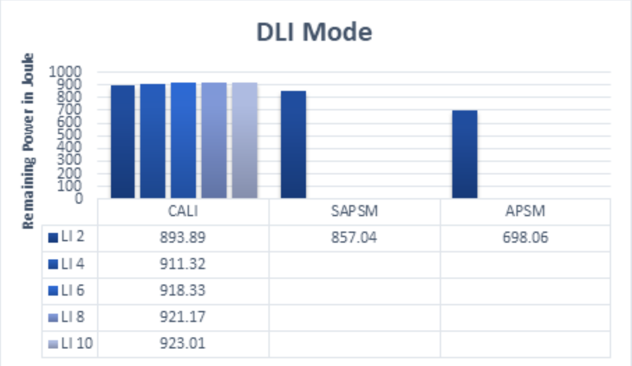


	SET 1	SET 2	SET 3
■ CALI	934.96	848.48	827.02
■ SAPSM	868.07	728.47	714.46
■ APSM	572.14	204.53	235.57

FIGURE 3. Comparison of CALI, SAPSM, and APSM in buffering mode against the 3 sets of energy parameters.

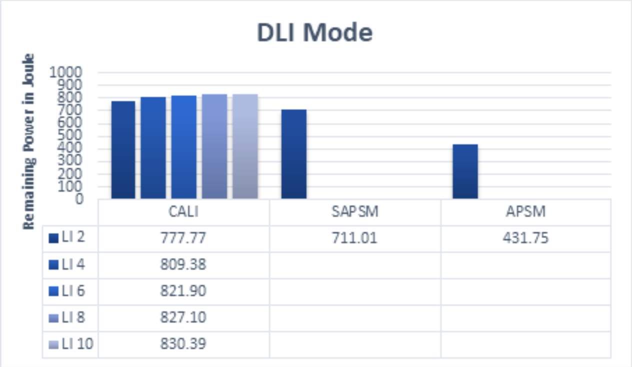
Figs. 4 to 6 show the levels of energy consumption of CALI, SAPSM, and APSM in DLI mode for the 3 sets of energy parameters. Recall that for DLI mode, the listen interval of a wireless device is incremented by 1 at each time a wireless device wakes up during its listen interval and does not find any packets buffered at the AP, and reverting to 1 when interactions occur. We adjusted the listen interval of CALI to 2,4,6,8, and 10, for applications with varied levels of network activity (Gmail and Facebook), as these applications have intermittent network interactions and not always receive data. Based on 30 emails and 30 Facebook posts, CALI consumes less energy in comparison to SAPSM and APSM for all 3 sets of energy parameters. Fig. 5 shows CALI consumes 8.58% to 14.37% less energy compared to SAPSM when the listen interval is set to between 2 and 10. This increases to between 44.48% and 48.00% less energy in comparison with APSM. In contrast, APSM consumes more energy than SAPSM and CALI in all 3 sets of energy parameters.

Although these applications run in the background non-interactively and do not always receive data, SPSM could add an approximate delay of 100-300ms of delay when the WNIC is off during the beacon intervals, but buffered packets are available at the AP. This added delay could reach 1000ms in the case of CALI when the listen interval is increased to 10.



	CALI	SAPSM	APSM
■ LI 2	893.89	857.04	698.06
■ LI 4	911.32		
■ LI 6	918.33		
■ LI 8	921.17		
■ LI 10	923.01		

FIGURE 4. Comparison of CALI, SAPSM, and APSM in DLI mode against set 1 of energy parameters.



	CALI	SAPSM	APSM
■ LI 2	777.77	711.01	431.75
■ LI 4	809.38		
■ LI 6	821.90		
■ LI 8	827.10		
■ LI 10	830.39		

FIGURE 5. Comparison of CALI, SAPSM, and APSM in DLI mode against set 2 of energy parameters.

The levels of energy consumption of CALI, SAPSM, and APSM in low mode are shown in Fig. 7. For all 3 sets of energy parameters CALI consumes less energy than SAPSM and APSM. In the experiments the listen interval of CALI was set to 20. Besides after the playing time of 110 seconds

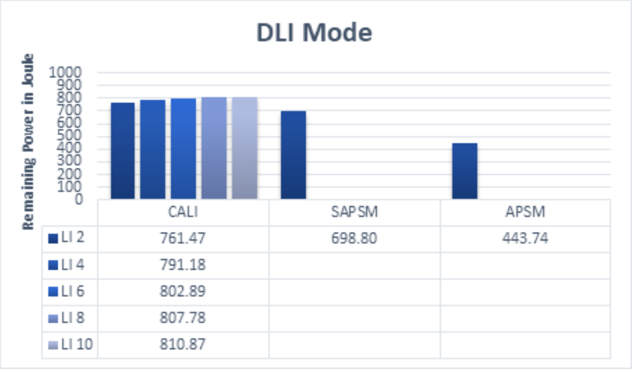


FIGURE 6. Comparison of CALI, SAPSM, and APSM in DLI mode against set 3 of energy parameters.

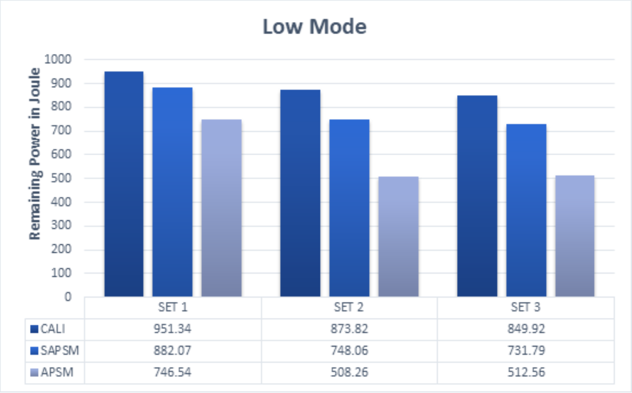


FIGURE 7. Comparison of CALI, SAPSM, and APSM in low mode against the 3 sets of energy parameters.

when the network traffic to load the advertisements occurs, we also observed a small level of network interaction during playing time. While small this was sufficient to switch APSM to awake mode. In Set 2, CALI consumes 14.39% less energy compared to SAPSM and 41.83% when compared to APSM.

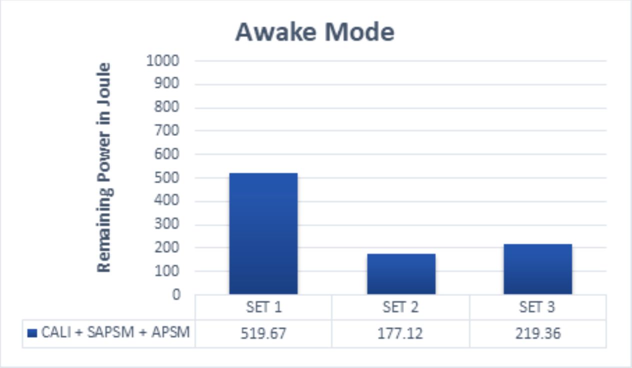


FIGURE 8. Comparison of CALI, SAPSM, and APSM in awake mode against the 3 sets of energy parameters.

Fig. 8 shows the levels of energy consumption of CALI, SAPSM, and APSM in awake mode for the 3 sets of energy parameters. As awake mode reflects applications with higher levels of network traffic, the WNIC is always on. Therefore, in all 3 sets of energy parameters, the levels of energy consumption of CALI, SAPSM, and APSM are identical.

Further investigation was carried out observing the behavior of CALI, as we varied the values of individual energy parameters between their max and min across the three sets. We chose each individual energy parameter and gradually increased its value from the minimum as in Set 1 to match the max value as in Set 2. The values for the other energy parameters were kept unchanged.

Figs. 9 to 13 show the energy consumption of CALI in buffering mode as the value of the individual power parameters were varied.

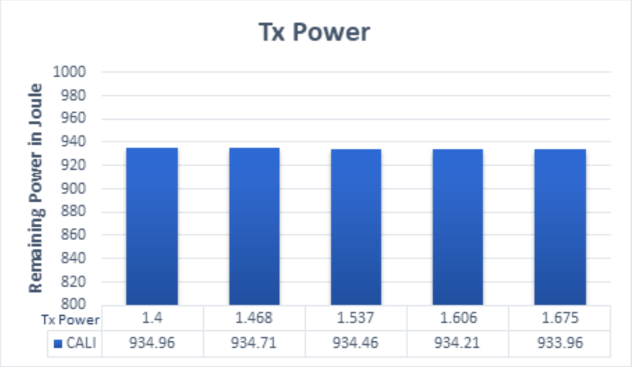


FIGURE 9. Levels of energy consumption of CALI in buffering mode against the value variations of txPower energy parameter.

Fig. 9 shows the energy consumption of CALI in buffering mode for changing values of txPower 1.4W (Set 1), to 1.675W (Set 2). In this context, txPower reflects the energy consumption of the acknowledgment packets sent by the wireless device to an AP upon receiving the destined packets.

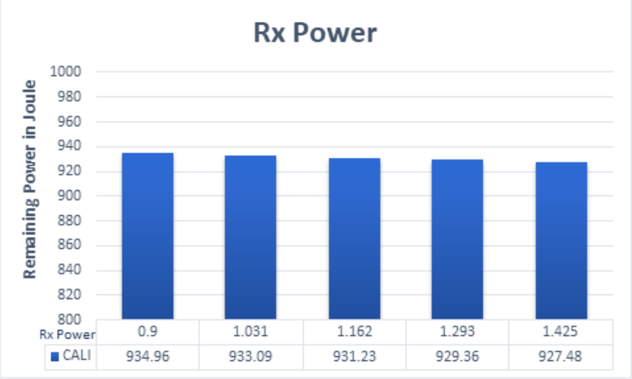


FIGURE 10. Levels of energy consumption of CALI in buffering mode against the value variations of rxPower energy parameter.

Fig. 10 illustrates levels of energy consumption of CALI in buffering mode when incrementing rxPower from 0.9W (Set 1), to 1.425W (Set 2). rxPower reflects the energy consumption of the wireless device while receiving packets from an AP.

As mentioned before, the value of transitionPower must be twice of idelPower. Therefore, we have incremented the values of transitionPower along with the value of idelPower. Levels of energy consumption of CALI in buffering mode when incrementing transitionPower and idelPower from values in Set 1 to values in Set 2 are shown in Fig. 11.

The transitionTime value identical in all 3 sets of 0.002s. In order to further analyze its impact on energy consumption, we have varied transitionTime between 0.005s and 0.0008s. The impact of increasing and decreasing the transitionTime on energy consumption of CALI in buffering mode is shown in Fig. 12.

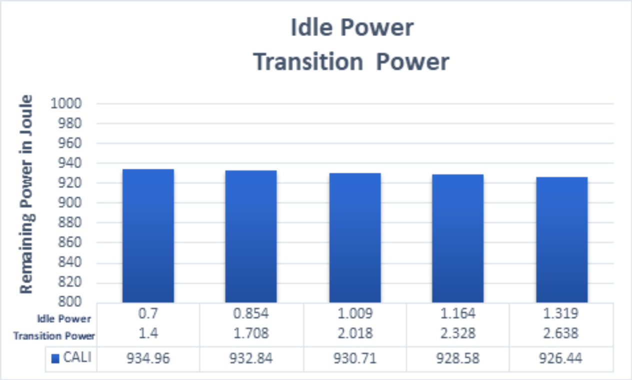


FIGURE 11. Levels of energy consumption of CALI in buffering mode against the value variations of idelPower and transitionPower energy parameters.

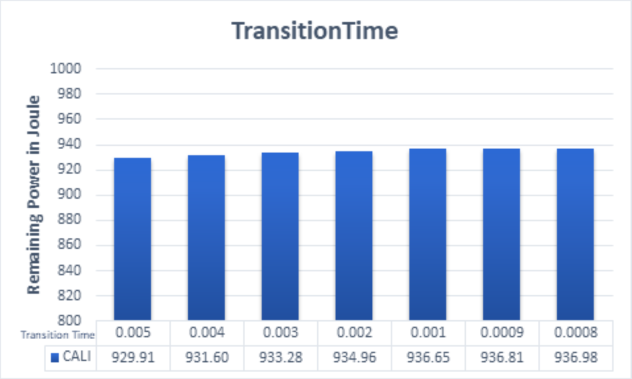


FIGURE 12. Levels of energy consumption of CALI in buffering mode against the value variations of transitionTime.

Fig. 13 shows levels of energy consumption of CALI in buffering mode while increasing sleepPower from 0.06W (Set 1), to 0.177W (Set 2). As can be expected, we observe that the value of sleepPower parameter has a major impact on the levels of energy consumption of CALI in comparison to the other energy parameters.

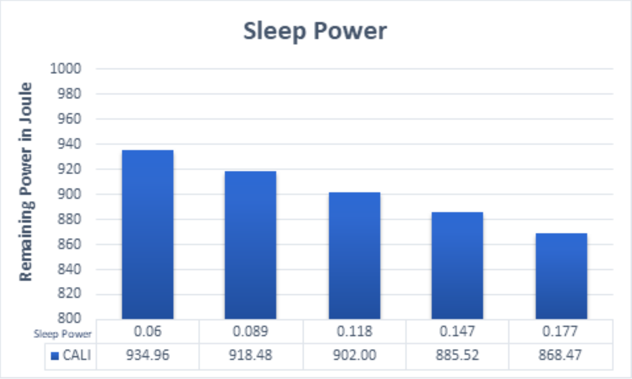


FIGURE 13. Levels of energy consumption of CALI in buffering mode against the value variations of sleepPower energy parameter.

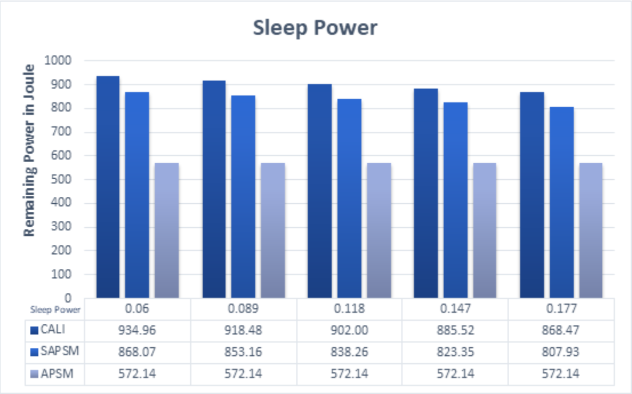


FIGURE 14. Levels of energy consumption of CALI, SAPSM, and APSM in buffering mode against the value variations of sleepPower energy parameter.

Fig. 14 shows the levels of energy consumption of CALI, SAPSM, and APSM in buffering mode when increasing sleepPower from 0.06W (Set 1), to 0.177W (Set 2). CALI consumes less energy than SAPSM and APSM. The power consumption of APSM remains static, as the WNIC remains awake and thus the value of sleepPower has no impact on energy consumption.

V. CONCLUSION AND FUTURE WORK

Regardless of the rapid growth and popularity of WLANs, the energy consumed by WNIC during wireless communication remains crucial to power-constrained wireless devices. To reduce the amount of energy consumed by WNIC, SPSM and APSM have been deployed in WLANs. However, several limitations with these approaches have been reported in the literature. Attempting to address some of these limitations, authors of [10] proposed SAPSM. SAPSM is based on categorizing smartphone applications into low and high priority apps using an ML classifier. However, no additional priority or mode has been proposed, e.g., for applications with very low levels of network interactivity or applications using buffer streaming.

Unlike other power saving approaches reported in the literature, our approach is based on the concept of context-aware listen interval. With this approach, the WNIC, with the aid of an ML classification model, sleeps and awakes based on the level of network activity of each application. In this paper, we have developed the four power saving modes of CALI using experimentation employing the NS2 simulator. The simulation results have demonstrated their efficacy in substantially reducing energy consumption.

Our approach relies on an ML classification model to optimize energy efficiency of power-constrained wireless devices. Therefore, the computational cost of training and testing the ML classifier is crucial [43], [44]. In [12] we have demonstrated high accuracy and low computational cost for building a classification model. Clearly, this is a one-off cost during deployment. Additionally, the cost of our approach at runtime is minimal as the WNIC simply operates in one of the CALI power saving modes, once the classification of the traffic is completed.

In future work, we envision to implement our approach in a smartphone testbed.

REFERENCES

[1] C. Deng, X. Fang, X. Han, X. Wang, L. Yan, R. He, Y. Long, and Y. Guo, "IEEE 802.11be Wi-Fi 7: New challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2136–2166, Jul. 2020.

[2] J. Saldana, J. Ruiz-Mas, J. Fernandez-Navajas, J. L. S. Riano, J.-P. Javaudin, J.-M. Bonnamy, and M. L. Dizes, "Attention to Wi-Fi diversity: Resource management in WLANs with heterogeneous APs," *IEEE Access*, vol. 9, pp. 6961–6980, 2021.

[3] X. Xie, W. Yang, and K. Tian, "Delay-aware power saving mechanism for 802.11 wireless LANs via NDN," in *Proc. 1st IEEE Int. Conf. Hot Information-Centric Netw. (HotICN)*, Shenzhen, China, Aug. 2018, pp. 154–159.

[4] M. F. Tuysuz, "Towards providing optimal energy-efficiency and throughput for IEEE 802.11 WLANs," *Int. J. Commun. Syst.*, vol. 31, no. 13, p. e3725, Jun. 2018.

[5] F. Wu, W. Yang, J. Ren, F. Lyu, P. Yang, Y. Zhang, and X. Shen, "Named data networking enabled power saving mode design for WLAN," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 901–913, Jan. 2020.

[6] N. Ding, A. Pathak, D. Koutsonikolas, C. Shepard, Y. C. Hu, and L. Zhong, "Realizing the full potential of PSM using proxying," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2821–2825.

[7] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: Network-assisted power management for WiFi devices," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 91–106.

[8] A. Pyles, D. T. Nguyen, X. Qi, and G. Zhou, "Bluesaver: A multi-PHY approach to smartphone energy savings," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3367–3377, Jun. 2015.

[9] V. Bernardo, B. Correia, M. Curado, and T. I. Braun, "Towards end-user driven power saving control in Android devices," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (Lecture Notes in Computer Science), vol. 8638. Cham, Switzerland: Springer, 2014, pp. 231–244. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-10353-2_20#aboutcontent, doi: 10.1007/978-3-319-10353-2_20.

[10] A. J. Pyles, X. Qi, G. Zhou, M. Keally, and X. Liu, "SAPSM: Smart adaptive 802.11 PSM for smartphones," in *Proc. ACM Conf. Ubiquitous Comput.*, Sep. 2012, pp. 11–20.

[11] X. Xia, S. Li, Y. Zhang, B. Li, Y. Zheng, and T. Gu, "Enabling out-of-band coordination of Wi-Fi communications on smartphones," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 518–531, Apr. 2019.

[12] A. Saeed and M. Kolberg, "Towards optimizing WLANs power saving: Novel context-aware network traffic classification based on a machine learning approach," *IEEE Access*, vol. 7, pp. 3122–3135, 2019.

[13] *IEEE 802.11-2012: Wireless LAN Medium Access Control MAC and Physical Layer PHY Specifications*, Standard IEEE 802.11, 2012.

[14] S. Baek and B. D. Choi, "Performance analysis of power save mode in IEEE 802.11 infrastructure WLAN," in *Proc. Int. Conf. Telecommun.*, Jun. 2008, pp. 1–4.

[15] A. J. Pyles, Z. Ren, G. Zhou, and X. Liu, "SIFI: Exploiting voip silence for WiFi energy savings insmart phones," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 325–334.

[16] H. Kwon, S. Kim, Y. Son, C. Yang, S. Byeon, and S. Choi, "AWARE: Adaptive Wi-Fi power save operation coexisting with LTE-U," in *Proc. IEEE 16th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Nov. 2019, pp. 208–216.

[17] Y. He and R. Yuan, "A novel scheduled power saving mechanism for 802.11 wireless LANs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1368–1383, Oct. 2009.

[18] P. Agrawal, A. Kumar, J. Kuri, M. K. Panda, V. Navda, and R. Ramjee, "OPSM-opportunistic power save mode for infrastructure IEEE 802.11 WLAN," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Cape Town, South Africa, May 2010, pp. 1–6.

[19] K. Omori, Y. Tanigawa, and H. Tode, "Power-saving for wireless stations using RTS/CTS handshake and burst transmission in wireless LANs," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2017, pp. 708–711.

[20] K. Omori, Y. Tanigawa, and H. Tode, "A study on power saving using RTS/CTS handshake and burst transmission in wireless LAN," in *Proc. 10th Asia–Pacific Symp. Inf. Telecommun. Technol. (APSITT)*, Colombo, Sri Lanka, Aug. 2015, pp. 1–3.

[21] Y. Zhang and Q. Li, "Exploiting ZigBee in reducing WiFi power consumption for mobile devices," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2806–2819, Dec. 2014.

[22] F. Lu, G. M. Voelker, and A. C. Snoeren, "SloMo: Downclocking WiFi communication," in *Proc. USENIX NSDI*, 2013, pp. 255–258.

[23] X. Zhang and K. G. Shin, "E-MiLi: Energy-minimizing idle listening in wirelessNetworks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1441–1454, Sep. 2012.

[24] W. Wang, Y. Chen, and L. Wang, "Sampleless Wi-Fi: Bringing low power to Wi-Fi communications," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1663–1672, Jun. 2017.

[25] Y. Li, X. Zhang, and K. L. Yeung, "DLI: A dynamic listen interval scheme for infrastructure-based IEEE 802.11 WLANs," in *Proc. IEEE 26th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Hong Kong, Aug. 2015, pp. 1206–1210.

[26] Y. Li, G. Zhou, G. Ruddy, and B. Cutler, "A measurement-based prioritization scheme for smartphone applications," *Wireless Pers. Commun.*, vol. 78, no. 1, pp. 333–346, Sep. 2014.

[27] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Amsterdam, The Netherlands: Elsevier, 2011.

[28] N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *Proc. 10th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Oct. 2015, pp. 43–48.

[29] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1988–2014, 2nd Quart., 2019.

[30] (Jun. 2021). *Weka Website*. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[31] M. Anand, E. B. Nightingale, and J. Flinn, "Self-tuning wireless network power management," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM)*, San Diego, CA, USA, Sep. 2003, pp. 176–189.

[32] M. Jeruchim, P. Balaban, and K. Shanmugan, "Simulation of communication systems: Modeling, methodology and techniques," *Information Technology: Transmission, Processing and Storage*. New York, NY, USA: Springer, 2006.

[33] A. R. Khan, S. M. Bilal, and M. Othman, "A performance comparison of open source network simulators for wireless networks," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng.*, Penang, Malaysia, Nov. 2012, pp. 34–38.

[34] *The Network Simulator-ns-2*. Accessed: Sep. 20, 2020. [Online]. Available: https://www.isi.edu/nsnam/ns/

[35] M. Fujinami, Y. Miyamoto, and T. Murakami. *Wireless LAN Power Management Extension for ns-2*. Accessed: Oct. 8, 2020. [Online]. Available: http://nspme.sourceforge.net/

[36] A. Ksentini and Y. Hadjadj-Aoul, "QoE-based energy conservation for VoIP over WLAN," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Paris, France, Apr. 2012, pp. 1692–1697.

[37] D. Liu, H. Wang, G. Zhou, W. Mao, and B. Li, "Arbitrating traffic contention for power saving with multiple PSM clients," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7030–7043, Oct. 2016.

[38] Y. Li, X. Zhang, and K. L. Yeung, "A novel delayed wakeup scheme for efficient power management in infrastructure-based IEEE 802.11 WLANs," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, New Orleans, LA, USA, Mar. 2015, pp. 1338–1343.

[39] Y. Xie, X. Luo, and R. K. C. Chang, "Centralizing the power saving mode for 802.11 infrastructure networks," in *Energy Technology and Management*. London, U.K.: IntechOpen, 2011, pp. 3–24. [Online]. Available: https://www.intechopen.com/chapters/20631, doi: 10.5772/18325.

[40] D. Jung, R. Kim, and H. Lim, "Power-saving strategy for balancing energy and delay performance in WLANs," *Comput. Commun.*, vol. 50, pp. 3–9, Sep. 2014.

[41] F. Wu, W. Yang, J. Ren, F. Lyu, P. Yang, Y. Zhang, and X. Shen, "Cutting down idle listening time: A NDN-enabled power saving mode design for WLAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[42] S. Sur, T. Wei, and X. Zhang, "Bringing multi-antenna gain to energy-constrained wireless devices," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2015, pp. 25–36.

[43] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," *Soft Comput.*, vol. 20, no. 1, pp. 343–357, 2016.

[44] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018.

AHMED SAEED received the bachelor's degree in computer and network engineering and the master's degree from the University of Westminster, London, in 2012 and 2013, respectively. He is currently pursuing the Ph.D. degree in computer science and mathematics with the University of Stirling, U.K. His research interests include optimizing the energy consumption of WLANs, computer networks, cybersecurity, and the application of artificial intelligence for traffic classification.

MARIO KOLBERG (Senior Member, IEEE) received the Ph.D. degree from the University of Strathclyde, U.K.

He is currently a Senior Lecturer of computing science and the Associate Dean with the University of Stirling. He has published more than 60 papers in leading journals and conferences. His research interests include peer-to-peer overlay networks, wireless sensor networks, and the Internet of Things. He is a member of a number of international conferences program committees on networking and communications. He is the Editor of the bi-annually published Consumer Communication Networking Series within the *IEEE Communications Magazine*. He is also on the Editorial Board of the *Peer-to-Peer Networking and Applications* (Springer) journal and has a long-standing involvement with the IEEE CCNC conference series.