P^2IT: Predicting Packet Interarrival Time in Asynchronous Duty-Cycling Sensor Networks

Tong Zhu[1,2], Qiang Ma[2], Junliang Liu[1], Shichao Liu[3], Yuan He[2]
[1]Department of Computer Science and Engineering, HKUST
[2]School of Software and TNLIST, Tsinghua University
[3]Department of Computer Science and Engineering, Dalian University of Technology

***Abstract*—In this paper, we investigate the predictability of packet arrivals in asynchronous duty-cycling wireless sensor networks (WSNs). We conduct statistical analysis on data traces collected in both outdoor large-scale and indoor testbed WSN to show that traditional well-known traffic models, e.g., Poisson process and Self-similarity process, do not fit well for modeling packet arrivals in both networks. According to our observations, some key characteristics such as *sleeping interval* and *sampling rate* have significant impact on the traffic patterns under Low power listening (LPL) models. Hence, we raise a question: could we achieve accurate prediction on the packet arrivals in asynchronous duty-cycling WSNs? To answer this question, we design a novel data-driven predictor P^2IT focusing on two prediction goals: (i) the arrival time of the next packet (ii) the number of arrival packets within a short time interval. We conduct extensive trace-driven experiments to demonstrate that our predictor achieves high accuracies on both prediction goals under various experimental settings.**

I. Introduction

Recent several years have witnessed the prosperous development of Wireless Sensor Networks (WSNs). WSNs serve as the basic infrastructures for various practical applications, e.g. event detection, target tracking. Generally the sensor motes are powered by batteries and deployed in the human-untouched areas where energy replacement is difficult. To solve the conflict between limited energy supplies of sensor nodes and the requirement of long-term deployment, some efforts are put into this field [1]. One of recent research work suggests operating sensor nodes in a duty-cycling work mode [2], [**?**]. In duty-cycling WSNs, radios of sensor nodes are altered between active and dormant states periodically. The duty-cycling operation has been employed in a variety of MAC layer protocols [3], which can be basically classified into *synchronous* and *asynchronous* categories. Synchronous protocols require sensor motes synchronously sleep and wake up, which incur tremendous synchronization overhead. In asynchronous protocols sensor motes do not coordinate their wake-up schedules. To guarantee that a sender and its receiver are both awake when transmitting a packet, Low Power Listening (LPL) mechanism is employed. Before sending a data packet, a sender transmits a preamble whose length covers one sleeping interval of its receiver so it is guaranteed that the receiver wakes up at least once to be aware of the transmission.

Understanding characteristics of asynchronous LPL traffic benefits a large number of WSN applications. Take energy saving for instance, it is observed that a large proportion of energy is wasted on preambles as senders usually use long preambles to capture the wake-up of receivers. We have measured the average preamble length and the average packet length in our testbed network based on TinyOS default LPL MAC. The result shows that a typical preamble is as long as 65

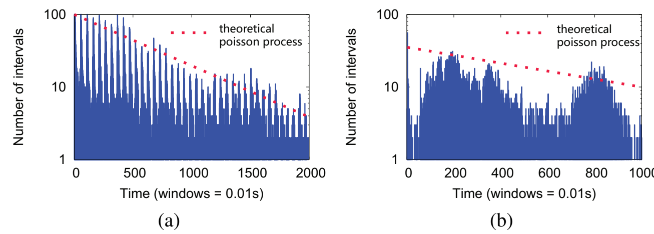

Fig. 1. Log histogram of interarrival times

packet payloads, which implies most of energy are not used to send useful information. If traffic pattern is known, a receiver can predict the arrival time of the next packet. Therefore, it can wake up exactly before the sender starts transmitting. Such manner significantly shortens preamble length and saves energy.

Existing WSN applications, nonetheless, either ignore traffic model or assume packet arrivals can be modeled as Poisson process. We observe that packet arrivals are somehow regular but do not fit any particular well-known model in asynchronous duty-cycling sensor networks, e.g. Poisson process and compound Poisson process. Fig. 1a and Fig. 1b plot the log histogram of interarrival times (between two consecutive packet arrivals) of packets as measured in two nodes. They show diverged packet arrival patterns, but fit neither Poisson process (a straight line) nor compound Poisson process. Other models, like the widely adopted Self-similarity process, also fail to meet the actual pattern. In addition, *sampling rate* and *sleeping interval* have great impact on the traffic across a network, causing various traffic models at different nodes.

Though it is difficult to find a universal and generic traffic model which fits for all the networks, the node-specific traffic pattern can still be discovered and leveraged. After thoroughly investigating the predictability of packet arrivals in asynchronous duty-cycling WSNs, we propose a novel data-driven approach to predict the packet arrivals. Specifically, we focus on two prediction goals: (i) the arrival time of the next packet (ii) the number of arrival packets within a short time interval. First, we analyze the real traces and extract meaningful feature sets. Second, due to the constraint of the resources, we do not have much memory space to store complete history information at sensor nodes. Instead, we select $top-k$ related features by Correlation based Feature Selection (CFS) [4] for the prediction. Third, we employ Bagging [5] method to train the traffic model which has been proven to well fit for the linear or non-linear regression. Trace-

978-0-7695-5104-3/13 $26.00 © 2013 IEEE
DOI 10.1109/MASS.2013.81

driven experiments demonstrate that the prediction accuracy of P^2IT scheme is over 90%. Our major contributions are summarized as follows.

1) We conduct extensive experiments and analysis of data traces collected in both outdoor large-scale WSNs (GreenOrbs) and indoor testbed networks. We show that the traffic patterns in asynchronous duty-cycling sensor networks are quite different from traditional Internet.
2) We explore some key features which have large impact on the traffics under LPL models. The results can be leveraged by protocol designers to analyze the traffic dynamics and packet delivery performance in such asynchronous duty-cycling WSNs.
3) We propose a novel data-driven approach to predict the packet arrivals. We can accurately predict over 90% of cases whether packets will arrive in a short time interval.

The rest of the paper is organized as follows. Section II discusses the foundation of our prediction methods. In Section III, we propose a novel data-driven approach to predict the packet arrivals. Section IV presents the evaluation of our methods by trace-driven experiments. Section V presents the related work. We conclude the paper in Section VI.

II. Prediction Foundations

In this section, we present a novel data-driven approach to accurately predict packet arrivals. Specifically, we summarize the key factors to network traffic under LPL scheme. The prediction goal and error metrics are also stated. Finally we analyze the traffic features.

A. Key factors of LPL traffics

As aforementioned, well-known models cannot faithfully fit for the actual data trace captured in GreenOrbs or testbed networks. We mainly discuss two factors: *sampling rate* and *sleeping interval*.

1) Sampling rate: Fig. 2 and Fig. 3 show the CDF of packet interarrival time of representative nodes from two data traces. In GreenOrbs, each node generates a packet every 10 minutes, while the frequency equals 10 seconds in testbed network. We can see that the nodes exhibit heterogeneous distributions in two networks. The distribution in Fig. 2 is close to exponential distribution, while in Fig. 3 it is close to power-law distribution. Considering that two networks are based on the modified CTP protocol with same parameter except the sampling rate, we conclude that the packet arrival time is bounded by the sampling rate. The previous study [6] has shown that the inter-meeting time in DTN network follows the exponential distribution rather than power-law distribution if boundaries exist. The sampling rate is like a "boundary" in LPL model for packet arrivals and shortens interarrival times. So, most of arrival intervals are less than 600 seconds in GreenOrbs and less than 10 seconds in testbed network. This observation also implies that the packet arrival process fails to fit Poisson process since one key characteristic of Poisson process is that the interarrival time should follow exponential distribution is not satisfied.

2) Sleeping interval: As shown in Fig. 1a, the peak occurs near the multiple of the sleeping interval (512ms). It is because the receiver needs to sleep and wake up alternatively. Thus the sender has to wait until the receiver wakes up, so as to transmit the packets. It leads to that most of packets arrive at the receiver side compactly when the receiver wakes up to poll

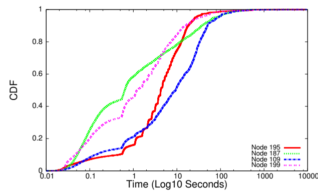

Fig. 2. Distribtuions of packet inter-arrival times of representative nodes in GreenOrbs

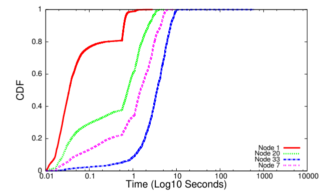

Fig. 3. Distribtuions of packet inter-arrival times of representative nodes in testbed network

the channel. When the sleeping interval closes to 0, it has little impact on the packet arrivals. If the sleeping interval tends to infinite, the interarrival times between packets concentrate to 0. Besides, in Fig. 1b, we can see that there exist two spikes at two different interval times. In this case, we make a conjecture that some subtree nodes may suffer congestions, such that packet delay has to be prolonged. Due to the difficulty to capture those uncertainties, simple homogeneous models cannot be able to predict packet arrivals accurately.

B. Prediction goals and error metrics

To accurately capture packet arrivals, we focus on three predictions: (i) prediction $\hat{P}(t)$ of the next packet arrival time $P(t)$ of a node received a packet at time t, (ii) prediction $\hat{N}_\delta(t)$ of the number of packets received in the time interval $[t, t+\delta]$, and (iii) an indicator function $\hat{I}_\delta(t)$, to denote whether the node will receive any packet in $[t, t+\delta]$ ($I_\delta(t) = 1$), or not ($I_\delta(t) = 0$). Previous two predictions are our main goals and the third one is to verify the accuracy of proposed predictor.

To evaluate our algorithm, we use error metrics employed by [7]. For the next packet arrival time, we measure the relative prediction error $E_P(t) = (\hat{P}(t) - P(t))/P(t)$. The values are in $[-1, \infty)$, with $E_P(t) = 0$ corresponding to a perfect prediction. For $\hat{N}_\delta(t)$, we measure the error $E_{N_\delta}(t) = \hat{N}_\delta(t) - N_\delta(t)$. For $\hat{I}_\delta(t)$, we measure the error E_{I_δ}, the fraction of time $\hat{I}_\delta \neq I_\delta$. $E_{I_\delta} = 0.5$ corresponds to a random predictor.

C. LPL traffic features

A predictor needs to exploit and extract useful features from the history which carry the most information about the traffic patterns to provide future traffic prediction.

We have known that packet interarrival time is mainly affected by *sampling rate* and *sleeping interval* in asynchronous duty-cycling techniques. Combining another important parameter "*delay_after_receive*", the waiting time a node turns to sleep, we propose a metric *radio_on_ratio*, the proportion of time that radio is active within $[t - \tau, t]$, to characterize the history packet arrival information. If we do not consider *delay_after_receive*, *radio_on_ratio* is proportional to the number of packet arrivals over the fixed windows. Otherwise, the receiver's radio may be in idle listening for a non-negligible time interval. In this case, *radio_on_ratio* is expected to capture arrival patterns more accurately as described in section III-C.

Another important feature is average packet interarrival time over the window $[t - \tau, t]$. Many works have shown that dynamic wireless channel conditions lead to bursty packet arrivals, for example Srinivasan *et al.* [8] propose a metric to measure link temporal correlations. To avoid computing

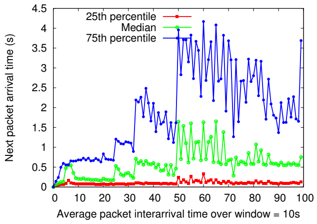

Fig. 4. Relationship between average packet interarrival time and next packet arrival time

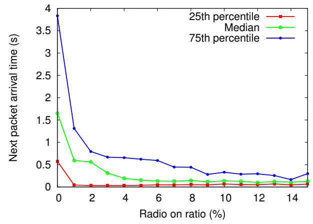

Fig. 5. Relationship between radio_on_ratio and next packet arrival time

overhead and capture burstiness arrivals, we employ a simple feature, average packet interarrival time over $[t-\tau, t]$.

How average packet interarrival time and *radio_on_ratio* impact the next packet arrival time? Fig. 4 shows the median, 25th and 75th percentiles of packet interarrival times as the function of the respective features. We notice that short average packet interarrival time implies that the next packet will come earlier than long ones. Similarly, in Fig. 5, *radio_on_ratio* increases, which means the next packet is likely to arrive early.

The variance of above features, however, is non-negligible. To do a better prediction, we list a set of additional features which may impact packet arrivals. The feature selection will be discussed in section III-C. Besides, we consider *Spatial features* which reflects the routing strategies. Specially, we consider link qualities in the neighborhood over time window $[t-\tau, t]$. For example, if the receiver found that all its link qualities got worse, the next packet arrival should be prolonged. There have been many studies on link estimation on dynamic wireless channel [9]. Considering the feasibility of data fetching, we use Received Signal Strength Indicator (RSSI) to indicate the link quality as many works do [10].

We extract useful features across three dimensions: *Temporal features*, *Spatial features* and *Event-based*. Table I shows the possible features. We emphasize two event-based features: intervals since the most recent packet arrival fluctuation and number of packet arrival fluctuations. First we define the fluctuation of packet arrivals. Assume $T = [t_1, t_2, ..., t_n]$ is packet arrivals, we aggregate the number of arrivals every window size δ, then we get a vector $B = [b_1, b_2, ..., b_{n_b}]$ corresponding time vector $T' = [t_{b_1}, t_{b_2}, ..., t_{b_{n_b}}]$. We define a fluctuation of the packet arrivals is a event that for some i, $|b_i - b_{i+1}| \geq h$. According to our experiences, it is better to set h as the half of the difference between the 75th value and 25th value of vector B. The features related to the fluctuation of the packet arrivals are important to reflect the burstiness of the packet arrivals. They capture the patterns of changes in the past, like parent selection oscillation between two nodes in common data gathering protocol [9].

D. Feature properties

Table I shows the correlation between the next packet arrival time and features extracted from GreenOrbs. We show the correlation values for $\delta = 10s$. As we can see, all the features do not have strong correlations with the prediction goals, which means that we cannot only use one of them to conduct the prediction. Hence, in section III-C, we will study the combination of the features for the prediction.

TABLE I
SET OF CANDIDATE FEATURES

Features (computed over $[t-\tau, t]$)	Correlation with P(r)
Temporal features	
Average interarrival time	0.13
Radio on ratio	0.15
Number of packets received	0.10
Variance of interarrival time	0.02
Maximal interarrival time	-0.04
Minimal interarrival time	-0.03
Spatial features	
Average RSSI value	0.09
Variance of RSSI value	-0.01
Maximal RSSI value	-0.03
Minimal RSSI value	-0.02
Event-based	
Times since the most recent packet arrival fluctuation	-0.06
Number of packet arrival fluctuations	-0.03

III. Novel prediction method for packet arrivals in asynchronous duty-cycling WSNs

We seek a predictor based on an intuitive and detailed model rather than a black box. However, due to high variance of dynamic wireless channel, the packet interarrival times are influenced by many factors, making model building and feature selection challenging. We employ Bagging [5], a state-of-the-art supervised machine learning technique, to bootstrap our modeling efforts.

A. Why choose Bagging

The reasons we resort to Bagging in our approach are three-fold. First, as we discussed in previous sections, it is difficult to estimate packet interarrival using a Poisson process or a linear function estimator such as Logistic Regression or SVM to fit our model. To validate such an conjecture, we compare the testing error on collected dataset using Logistic Regression (LR), SVM and Decision Tree (C4.5). We tune the parameter settings of all these three classifiers to ensure they produce the best regression output on the testing data. As shown in Fig. 6, decision tree achieves the best performance. SVM achieves a marginally better result than LR. Second, Bagging provides a natural ensemble compared to a single decision tree by creating several training sets based on sampling with replacements. It is proven that such an approach can reduce the variance of the estimator when we decompose the error of the estimator along the direction of Bias-Variance Decomposition. One particular feature of the dataset we collected is that the sample variance is huge, as one of the main characteristics of wireless sensor network dataset is traffic imbalance, i.e., different nodes show diverse traffic patterns. Thus, when we are aiming to train a regressor on such a dataset to reduce loss, an approach such as Bagging that focuses on reducing the variance of the learned estimator would be more appropriate. Third, a very low computational complexity is needed in the testing phase. Such a characteristic is important when the computing power is limited, especially when we consider the specific scenario in our settings where we are conducting the testing phases with sensors.

B. Training and test sets

Bagging, like any supervised learning algorithm, trains data set consists of a set of labelled feature vectors. Each feature vector consists of the feature listed in Table I, and with the

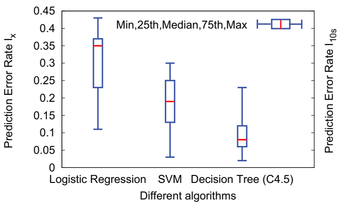

Fig. 6. The performance of different machine learning algorithms

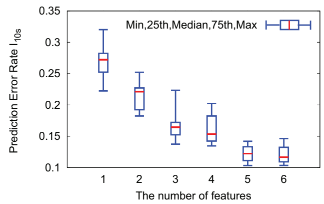

Fig. 7. Impact of number of features

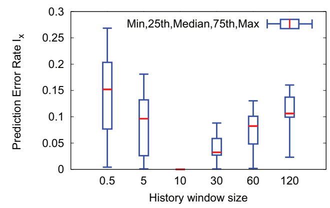

Fig. 8. Impact of history window size

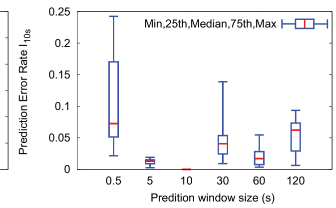

Fig. 9. Impact of prediction window size

next packet arrival time $P(t)$ at some time t. Similar training sets are generated for $N_\delta(t)$ and $I_\delta(t)$.

We take packet arrivals within $[t-\tau, t]$ and combine them with RSSI values recorded in the packets to construct an input feature vector. The important parameter determined in input vector construction is history windows size δ. We have mentioned that *sleeping interval* is one key factor in LPL technique. Besides, we observe that the packet interarrival times always fall in the bins which are the multiple of the *sleeping interval*. Hence, we choose the multiples of sleeping interval lengths as history window sizes, and construct several training sets to identify the predictability of interarrival time.

Training data is extracted from each mote respectively. We randomly split constructed each input vector set into two parts: 66% input instance as the training set and the remaining 34% instances as the test set. We train the Bagging model and apply it to the testing data. All error metrics defined in II-B are compared to assess the performance.

C. Feature Selection

TABLE II
FEATURE IMPORTANCE

Features	Importance
Average interarrival time	100
Ratio on ratio	80
Number of packets received	70
Average RSSI value	50
Number of packet arrival fluctuations	30
Others	≤ 10

Table II shows feature selection results with CFS. As we can see, average interarrival time is most useful since it captures the relative long-range burstiness of the wireless channel and the dynamics of the routing strategies. However, it cannot differentiate the burstiness in a history window of size τ. *Radio_on_ratio* is the second important as we expected. It focuses on the dynamics in a history window. So the first two features compensate each other. Other features also work well as we expected. Considering the computing limitations on the sensor motes, we expect to use the features as few as possible. Fig. 7 shows the prediction error only using the top k features in Table II. The performance is stable when the feature increases to 5. It implies that when we use those top 5 features, the prediction is as accurate as all features included. This helps us to reduce the computation cost on the energy-constraint sensor motes.

IV. Evaluation

In this section, we conduct trace-driven experiments and evaluate our novel method according to the three prediction goals: the next packet arrival time, the number of packet will be received over window $[t, t+\delta]$, whether the packet will arrive over window $[t, t+\delta]$.

1) Parameter configuration: We study two key parameters which have great influence on the prediction accuracy: the history window size τ and the prediction window size δ. Both δ and τ are fixed as 10s to generate the training and test sets when we study each other respectively. The reason for choosing 10s are two-fold. Firstly, in this setting we can achieve a stable prediction goal and differentiate other parameters well. Secondly, the limited resources in WSNs prevent us from recording enough history information.

As shown in Fig. 8, the prediction error decreases when we increase the history window size before 10s and decrease after that. It is because that adding too much history information may not help our predictor. The burstiness of traffic patterns may be smoothed by relative long-range statistics. Note that the average prediction errors stay at a very low level (≤ 0.15).

As shown in Fig. 9, our predictor achieves a high accuracy for various prediction windows size. For a very short range ($\leq 0.5s$), however, the predictor does not work well. It is because that the back-off mechanism of the 802.15.4 MAC protocol randomly chooses a time interval ($2ms$) to avoid the collision. This randomness affects the prediction error when the absolute value of interarrival time is not large.

According to the experiment results, the average accuracy of our predictor is over 90% at various settings, so it can significantly help transmission energy saving.

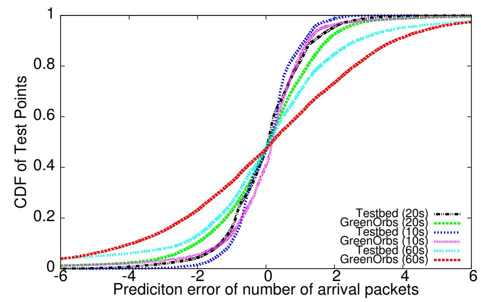

Fig. 10. Distribution of prediction error $\hat{N}_\delta$

2) Number of packet arrivals in next δ interval: Fig. 10 shows the distribution of the error of $\hat{N}_\delta$ in GreenOrbs network and testbed network with different prediction window sizes. Note that nearly above 90% of test points have $-2 \leq E_{\hat{N}_{10s}} \leq 2$ in most curves. In the same network, when the prediction window size increases, the performance

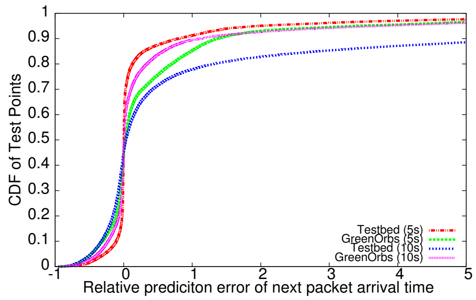

Fig. 11. Distribution of prediction error $\hat{P}$

of prediction decreases. For example, 100% of test points in testbed network with $\delta = 10s$ have $-2 \leq E_{\hat{N}_{10s}} \leq 2$, while only 75% of test points with $\delta = 20s$ fall in $[-2, 2]$. It is reasonable that the history information cannot perfectly capture the dynamics of the wireless channels and routing strategies when the network condition is unstable in a relative long time interval. In addition, over 50% of test points have $|E_{\hat{N}_{10s}}| \geq 2$ with $\delta = 60s$ in GreenOrbs network. It is quite different from indoor testbed network. It is because that outdoor environments are highly dynamic which may incur more traffic burstiness than indoor testbed network.

3) Next packet arrival time: If we can predict accurately when the next packet will arrive, we would like to concentrate our energy on the transmitting data packets instead of idle sensing or transmitting preambles. Fig. 11 shows the distributions of the relative error of $\hat{P}$. In both GreenOrbs and testbed data, with $\delta = 5s$, 50% of test points have the relative error near 0. It shows that our predictor can accurately predict the next packet arrival timing and provide the opportunities for more precise radio control. Different from the prediction of $\hat{N}_\delta$ in the previous section, we notice that the prediction accuracy decreases significantly when increasing prediction window size in testbed network. The result also implies that fast sampling rate leads to the inaccuracy of the prediction of $\hat{N}_\delta$. Hence, for fast sending, the traffic behavior towards to be more randomly, it is difficult to model the traffic pattern well.

V. Related Work

At the early stage, when modeling Internet traffic, packet and connections are always modeled as Poisson process. However, the authors in [11] argue that exponential distributions cannot fit packet interarrival time distribution well. They evaluate 24 wide-area traces and investigate a number of wide-area TCP arrival processes to determine the mismatch between the actual traffic pattern and the theoretical Poisson processes, and find out most connection arrivals are not well-modeled by Poisson process and might be related to the self-similarity features. Leland *et al.* [12] show that the local area network (LAN) traffic is statistically self-similar that there is no natural length of "bursty", so the self-similar properties can capture the long-range burstiness. Recently, more people are aware of self-similarity property in the wide-area traffic. For example, Huang *et al.* [13] study the co-existence problem between WiFi and ZigBee networks and propose a Pareto model for WiFi white space based on the observation of the self-similarity phenomenon. Besides, Liu *et al.* [14] study the optimal real-time sampling rate assignment problem to improve the throughput.

VI. Conclusion

In this paper, we investigate the predictability of packet arrivals in asynchronous duty-cycling WSNs and propose a novel data-driven approach to predict the packet arrivals. We conduct statistical analysis on data traces collected in both outdoor large-scale WSNs and indoor testbed network to show that traditional well-known traffic models, e.g. Poisson process, Self-similarity process do not well fit for the packet arrivals in both practical networks. However, most of protocols assume that the traffic model as Poisson process or even do not consider a traffic model which leads to the mismatch between the theoretical performance and the actual performance. According to our observations, some key characteristics e.g. *sleeping interval*, *sampling rate*, have significantly impact on the traffic patterns when using Low power listening (LPL) techniques. To achieve accurate prediction on the packet arrivals in asynchronous duty-cycling WSNs, we design a simple predictor mainly focusing on two prediction goals : (i) the next packet arrival timing (ii) the number of packets arrivals in the next $[t, t+\delta]$ time window. We conduct extensive trace-driven experiments to demonstrate that our predictor reach high accuracy on both prediction goals under various settings.

VII. Acknowledgement

We would like to thank the anonymous reviewers. This work is supported in part by the National Basic Research Program (973 program) under Grant of 2014CB347800, National Basic Research Program of China (973 Program) under Grants No. 2011CB302705, and NSFC under Grants 61170213.

References

[1] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, "Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks," *IEEE TPDS*, vol. 17, no. 6, pp. 536–547, 2006.
[2] W. Ye, J. S. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, pp. 1567–1576, 2002.
[3] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "Pw-mac: An energy-efficient predictive-wakeup mac protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, pp. 1305–1313, 2011.
[4] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *ICML*, pp. 359–366, 2000.
[5] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
[6] H. Cai and D. Y. Eun, "Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet," in *Proceedings of ACM MOBICOM*, pp. 159–170, 2007.
[7] Ítalo Cunha, R. Teixeira, D. Veitch, and C. Diot, "Predicting and tracking internet path changes," in *Proceedings of ACM SIGCOMM*, pp. 122–133, 2011.
[8] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The beta-factor: measuring wireless link burstiness," in *ACM SenSys*, pp. 29–42, 2008.
[9] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of ACM SenSys*, pp. 1–14, 2009.
[10] T. Liu and A. Cerpa, "Foresee (4c): Wireless link prediction using link features," in *Proceedings of ACM/IEEE IPSN*, pp. 294–305, 2011.
[11] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," in *Proceedings of ACM SIGCOMM*, pp. 257–268, 1994.
[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *Proceedings of ACM SIGCOMM*, pp. 183–193, 1993.
[13] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting wifi white space for zigbee performance assurance," in *Proceedings of IEEE ICNP*, pp. 305–314, 2010.
[14] X. Liu, Q. Wang, W. He, M. Caccamo, and L. Sha, "Optimal real-time sampling rate assignment for wireless sensor networks," *ACM TOSN*, vol. 2, no. 2, pp. 263–295, 2006.