Lars Hanschke, Jan Heitmann, and Christian Renner. Stop Waiting: Mitigating Varying Connecting Times for Infrastructure WiFi Nodes. In Proc. of the 16th GI/ITG KuVS Fachgespräch Sensornetze, FGSN '17, Hamburg, Germany, Sept. 2017.

Stop Waiting: Mitigating Varying Connecting Times for Infrastructure WiFi Nodes

Lars Hanschke
Research Group smartPORT
Hamburg University of Technology
lars.hanschke@tuhh.de

Jan Heitmann
Research Group smartPORT
Hamburg University of Technology
jheitmann@tuhh.de

Christian Renner
Research Group smartPORT
Hamburg University of Technology
christian.renner@tuhh.de

***Abstract*—Increased spreading of the Internet of Things (IoT) in industrial applications but also home automation impairs the problem of a variety of different gateways and interoperability between connected devices. Connecting smart devices without additional infrastructure among each other and to the Internet is possible relying on existing WiFi networks. To ensure prolonged lifetime of battery- or environmentally-powered sensor nodes, multiple demand-planning algorithms exist. Currently, exact demand planning of WiFi nodes is hindered by highly fluctuating connecting times to their central gateway. This leads to wrong estimation of the actual energy consumption and consequently shortened lifetime or increased latency. We show that the high energy demand of long connecting times can be tackled by aborting the connection attempt and retrying later. Additionally, this reduces the variance of connecting times, making prediction of the energy demand more reliable. Our investigations show that power consumption can be reduced by up to 30% dependent on application. This leads to longer lifetime and paves the way for smart infrastructure relying on WiFi sensor nodes.**

I. Introduction

The growing demands on efficiency of industrial environments, e.g. port areas or larger production plants, lead to spreading of the Internet of Things (IoT) in various scenarios. While classical sensor networks require several gateways with different radio standards [1], the established WiFi standard offers seamless integration of Internet access and interoperability between different vendors. Furthermore, many cities offer or plan public WiFi networks, most industry sites maintain their own WiFi infrastructure and nearly every household has its private WiFi network.

While the research on intermittent computing draws considerable attention in literature, e.g. in [2] or [3], many industrial applications require their systems to be always available and down-times are unwanted. Intelligent infrastructure for transport systems as presented in [4] needs to be always available but not necessarily as fast as possible. A smart road sign may be used for monitoring environmental conditions, e.g. monitoring fine dust pollution, but also needs to transmit status information about battery level or if it fell on the ground due to a collision. Since reacting on these events, e.g. prohibiting diesel-fueled cars from entering inner city zones, takes a considerable time, latency of communication is not the biggest concern. However, a minimum degree of availability has to be ensured, which is hardly feasible with intermittent computing. Thus, duty cycle adjusting algorithms may slow down reaction time to ensure perpetual operation. To employ these sensor networks with low maintenance effort, a self-sustained energy resource is needed. Whether this is solar energy as presented in [5] or wind in [6], the limited amount needs careful usage of energy. To ensure prolonged lifetime, the highly fluctuating nature of the energy resource has to be tackled with an exact knowledge of sensor nodes future energy demand. Any source of miscalculation has to be eased to facilitate self-sustained operation while simultaneously matching application requirements of latency and reception ratio.

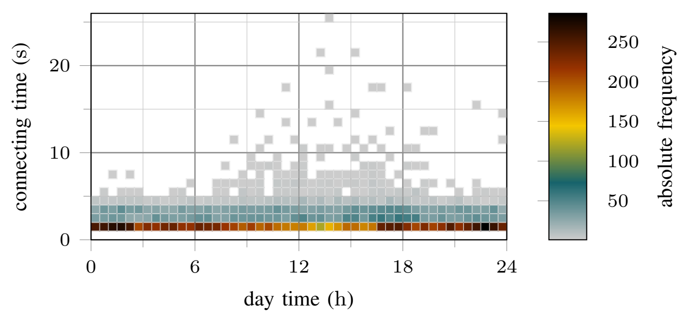

Fig. 1. Connecting times to access point by time of the day; connecting times are influenced by load of surrounding networks producing traffic during regular working times.

With this in mind, we present the issue of varying connecting times with small WiFi sensor nodes relying on these algorithms and show a strategy how to reduce this impact.

II. Analysis

To enable permanent availability of the sensor system, carefully adjusting the duty cycle of the sensor node is mandatory. The classical power saving mode in infrastructure WiFi with star-topology requires periodical wake-up for receiving beacons of the associated access point. As shown in [7], the power consumption of a WiFi sensor node staying connected during sleep mode is heavily influenced by the beacon interval of the access point. E.g, the power consumption of the Arduino Nano rises from $4\,\mathrm{mW}$ to $7\,\mathrm{mW}$ while halving the beacon interval. A typical beacon interval is $100\,\mathrm{ms}$; thus, duty cycling is very limited. Actuators, requiring faster response time, might use this operating mode, but for classical sensingonly applications, e.g. waking up every 60 s gathering sensor data, this approach is inefficient. Consequently, disassociating from the access point, going to sleep and re-associating upon next wake-up promises lower energy demand for longer sleep periods. The time at which re-associating is beneficial varies with the used hardware platform. For the ESP 8266 [8], we obtained through early measurements that re-associating is beneficial with sleep times longer than 10 s.

Association to the access point includes several steps and differs upon used authentication protocol [9]. Furthermore, if the Dynamic Host Configuration Protocol (DHCP) is used, obtaining a valid IP address also takes time. Thus, re-associating has to be carefully balanced against the classical power save mode in WiFi.

The resulting connecting times to the network during our long-term test are shown in Fig. 1. To emulate easy-to-use operation, we used a standard WiFi router with open-source firmware `dd-wrt` and enabled DHCP inside our University building at an operational frequency of 2.4 GHz. During times of high traffic, e.g. in the afternoon, connecting times to the network are considerably higher; in some cases up to 25 s. The resulting energy consumption increases significantly: the whole connecting phase is spent in a power-hungry state. For comparison, this consumes about 5% of the energy stored in a 50 F super-capacitor at 2.7 V.

Furthermore, algorithms adjusting the duty cycling of the sensor node, e.g. [10] or [11], rely on accurate planning of the energy demand. This makes their calculation prone to outliers within short-term prediction horizons. Thus, preventing the node from a fluctuating energy demand is inevitable for correct duty cycling.

To assess the impact of varying connecting times, we present a theoretical model and show a strategy for alleviating this impact. Our simulation results show that the variation of energy demand can be decreased, which leads to more accurate demand planning. The first step of our approach is to assess the point in time at which aborting the connecting process potentially reduces the energy demand of the sensor node.

III. Model Description

The Cumulative Distribution Function (CDF) of connecting times, presented in Fig. 2, shows that connection to the network can be established either quite fast or takes considerably longer, e.g. being connected in at most 1300 ms has a probability of 0.7. Consequently, it is more likely to establish a connection during the first part of the trying period compared to the later section. Finding the border between these two phases is the key to enable the following procedure: aborting the connecting process and trying again afterwards. This is promising, since the overall time spent in the connecting state with active radio module may be reduced.

A. Determining the Lower Bound

Mathematically, we are looking for the time t_a at which aborting the connecting process and retrying at a later point leads to an overall decreased connecting timespan. This decreases the energy consumption of the node.

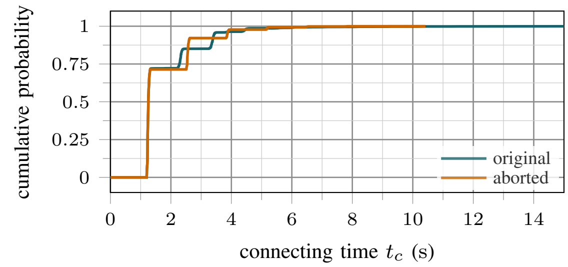

Fig. 2. Comparison of CDFs between originally recorded data and connecting times achieved when aborting and retrying to connect after t_a.

Assuming the connecting times t_c can be described by a random variable C with Probability Density Function (PDF) $f_C(t)$; the expectation value of the connecting time without aborting is defined as

$$\mathbf{E}\left[C\right] = \int_0^\infty t \cdot f_C(t)\mathrm{d}t. \tag{1}$$

If we reached t_a, the expected value of the connecting time additional to t_a is defined as

$$\mathbf{E}\left[C \geq t_a\right] = \frac{1}{1 - F_C(t_a)} \int_{t_a}^\infty (t - t_a) \cdot f_C(t)\,\mathrm{d}t, \tag{2}$$

which can be rewritten as

$$\mathbf{E}\left[C \geq t_a\right] = \frac{1}{1 - F_C(t_a)} \int_{t_a}^\infty t \cdot f_C(t)\,\mathrm{d}t - t_a, \tag{3}$$

with $F_C(t_a)$ denoting the CDF at time t_a. Basically, Eq. (3) is the expectation value of being connected after already waiting for t_a scaled with the probability that this event occurs. The desired time t_a is reached, if the expectation value of aborting and retrying is smaller or equal than trying infinitely. Consequently, we combine Eq. (3) and Eq. (1) so that

$$\frac{1}{1 - F_C(t_a)} \int_{t_a}^{\infty} t \cdot f_C(t)\,\mathrm{d}t - t_a \leq \int_0^{\infty} t \cdot f_C(t)\mathrm{d}t$$

$$\Rightarrow \int_{t_a}^{\infty} t \cdot f_C(t)\,\mathrm{d}t \leq \left(1 - F_C(t_a)\right)\left(t_a + \int_0^{\infty} t \cdot f_C(t)\,\mathrm{d}t\right) \tag{4}$$

holds. This equation can be numerically solved with values obtained from long-term measurements. Note that t_a is the first point in time, at which the energy consumption decreases when aborting the connecting process; thus, t_a is a lower bound for all reasonable aborting times.

B. Impact on Delay

A classical sensor network application gathers sensor data in a constant time interval τ and reports it regularly. Depending on the application, the receiving side has fixed latency requirements, meaning a packet arriving before a critical delay d_c. The accepted amount of outdated packets defines a demanded

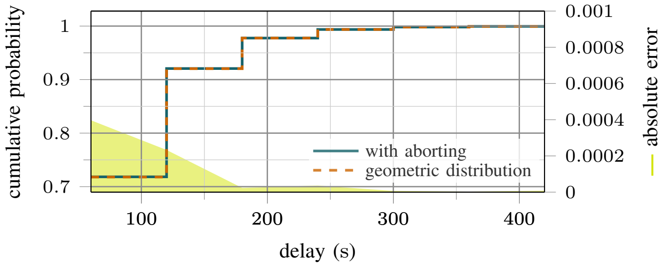

Fig. 3. Cumulative probability of delay; aborting connecting and delaying report to next report attempt leads to geometrically distributed delay.

reception ratio. In case of a fixed reception ratio, the benefit of packets arriving much faster is minor. This can be exploited by delaying a transmission to the next regular reporting time instead of spending a large amount of energy on trying to transmit the packet as fast as possible.

Beginning with the first transmission attempt, the situation is similar to throwing a coin with success probability p. Each of the connection attempts n is expected to have the same success probability p. Since delay is primarily influenced by the number of transmission attempts until one success occurs, it is described by a geometric distribution with PDF

$$f_D(n) = (1-p)^{n-1} \cdot p. \tag{5}$$

In our case, the success probability is determined by the aborting time t_{ab}, which is lower bounded by t_a, so that $t_a \leq t_{\text{ab}}$ holds. Consequently, $p = F_C(t_{\text{ab}})$. Furthermore, the delay introduced by one transmission attempt $n \in \mathbb{N}, n \leq n_{\text{max}}$ increases the delay by $t = n \cdot \tau$. This rewrites Eq. (5) to:

$$f_D(t) = (1 - F_C(t_{ab}))^{(t/\tau - 1)} \cdot F_C(t_{ab}). \tag{6}$$

The overall reporting delay also has a second component. Once the connection to the access point is established, the queued packets have to be transmitted in addition to the newest packet. As stated in [12], the service time of the common Distributed Coordination Function (DCF) can be approximated by an exponential distribution with PDF

$$f_S(t;\lambda) = \lambda e^{-\lambda t} \quad \text{for} \quad t \geq 0. \tag{7}$$

We determined the parameter λ by performing a curve fitting of the observed delay distribution, which we omit here due to space constraints. The resulting reporting delay distribution is displayed in Fig. 3 along with the connecting delay obtained by simulations.

C. Impact on Current Consumption

The primary goal for alleviating the impact of fluctuating connecting times is to decrease the energy consumption of the sensor node. Our model assumes the node to be in one of three different states: deep-sleep, active or associating to a WiFi network, and actively transmitting. This results in three approximated currents I_{ds}, I_{ac} and I_{tx} drawn by the communicating device. The power consumption follows

TABLE I
CURRENT CONSUMPTION ESPRESSIF ESP 8266

state	**variable**	**consumption**	
deep sleep	I_{ds}	26	µA
actively connecting	I_{ac}	69.47	mA
transmitting	I_{tx}	204.10	mA

directly given a constant supply voltage. The actual values in these states can be obtained from measurements and differ between devices, e.g. $I_{\text{ac}} \approx 84.1\,\text{mA}$ and $I_{\text{tx}} \approx 105\,\text{mA}$ for the Arduino MKR1000. Consequently, the resulting overall current drawn during T can be calculated as follows:

$$I^* = \frac{1}{T} \cdot \Big(I_{\text{ds}} \cdot (T - t_{\text{ac}} - t_{\text{tx}}) + (I_{\text{ac}} - I_{\text{ds}}) \cdot t_{\text{ac}} + (I_{\text{tx}} - I_{\text{ds}}) \cdot t_{\text{tx}} \Big). \tag{8}$$

t_{ds}, t_{ac} and t_{tx} denote the time spent in the corresponding states. Note that $T = t_{\text{ds}} + t_{\text{ac}} + t_{\text{tx}}$. The key idea behind aborting and delaying transmission attempts is to reduce the current consumption by reducing time spent in the costly states while simultaneously satisfying the application inherited reception ratio determined by the delay boundary d_c.

IV. EVALUATION

A brief overview of our measurement platform is covered in the first part of the section, whilst the second presents the simulation results of the approach introduced in Section III.

A. Hardware and Setup

The Espressif ESP 8266 [8] offers a 32 bit 160 MHz CPU with 160 kB RAM and is available for less than €3. It is fully compatible with IEEE 802.11b/g/n at a size of $16 \times 26\,\text{mm}$.

We determined the current consumption with an INA139 measurement amplifier with measurement shunts $R_S = 1\,\Omega$ in active states and $R_S = 1\,\text{k}\Omega$ in deep-sleep state. We summarized the measurement results in Table I.

The connecting and reporting delays were recorded during a 16-days long-term test in our university building. The WiFi node attempts to connect to a private access point with no other wireless clients in the network. We used the data to determine the parameters of our model presented in Section III and performed simulations with MATLAB to evaluate our approach to reduce the current consumption of the node.

B. Results

Based on the recorded connecting time data already presented in Fig. 2, the algorithm evaluating Eq. (3) yields a lower bound of aborting times $t_a = 1313\,\text{ms}$ with $F_C(t_a) = 0.72$. Evaluating the approach of aborting after t_a and attempting to connect directly afterwards, yields connecting times with CDF shown in Fig. 2. While the mean value of connecting times can only be reduced from 1790 ms to 1767 ms, the standard deviation can be reduced from 1164 ms to 995 ms.

Delaying the transmission attempt to the next regular sampling time decreases the connecting times significantly: the

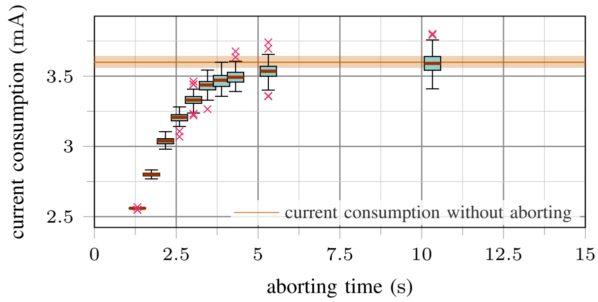

Fig. 4. Mean current consumption w.r.t. different aborting times; aborting reduces time spent in energy-costly state; thus, current consumption decreases by 30% to 4%; current consumption without aborting is shown as median; shaded area indicates upper and lower quartiles.

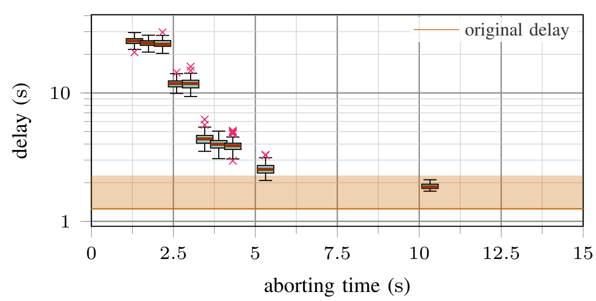

Fig. 5. Packet delay increased with short aborting times and relaxes for longer aborting times; original delay is shown as median; shaded area indicates upper and lower quartiles; note that the upper quartile is more than twice the median.

mean connecting time is $1265\,\mathrm{ms}$ with standard deviation of only $37\,\mathrm{ms}$. This enables demand planning algorithms to be much more reliable.

Figure 4 shows the current consumption of this method with different aborting times beginning with t_a. Using a report interval $\tau = 60\,\mathrm{s}$, the benefit of aborting is clearly visible; e.g. aborting after $1742\,\mathrm{ms}$ saves 22% current.

The downside of this approach is increased packet delay. E.g., for an aborting time of $1742\,\mathrm{ms}$ the median delay is more than $24\,\mathrm{s}$. Since it is mainly influenced by the time difference between two transmission attempts, the delay scales with τ. We plotted the results for $\tau = 60\,\mathrm{s}$ in Fig. 5 for co-existence with Fig. 4, but the situation slightly relaxes for smaller sampling intervals. This shows that power saving mainly affects delay-uncritical applications; but if the main focus is reliable prediction of energy demand, this approach is highly advisable.

As mentioned earlier, many applications require a fixed reception ratio w.r.t. a critical delay d_c at which a packet is discarded, since it is outdated. We depict the needed current consumption for satisfying different reception ratios or percentile of packets with latency smaller than d_c in Fig. 6. Dependent on the critical delay, the power savings can be considerable. E.g., for an application sampling every $30\,\mathrm{s}$ and requiring 90% of the packets to arrive with a delay less than $60\,\mathrm{s}$, current consumption can be reduced by 30%.

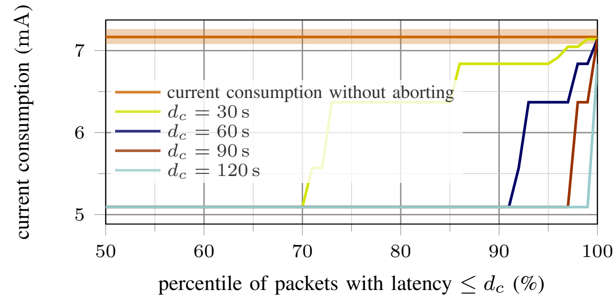

Fig. 6. Current consumption to fulfill demanded reception ratio dependent on delay d_c; relaxing delay requirements or reception ratio highly reduces the current consumption; reporting interval $\tau = 30\,\mathrm{s}$; original current is shown as median; shaded area indicates upper and lower quartiles.

V. Conclusion

We showed that variating WiFi connecting times impact the operation of duty-cycle-adjusting sensor nodes. Our strategy reduces the current consumption for applications with relaxed delay requirements considerably. Additionally, we ease the fluctuation of connecting times, helping demand planning algorithms to work more reliable. This prepares the ground for smart infrastructure using self-sustaining WiFi sensor nodes with low energy budget. Additionally, it allows us to perform real-world tests in harbor environments.

References

[1] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The Internet of Things has a Gateway Problem," in *Proc. HotMobile*, Santa Fe, NM, USA, Feb. 2015.

[2] B. Lucia and B. Ransford, "A Simpler, Safer Programming and Execution Model for Intermittent Systems," in *Proceedings of PLDI*. New York, NY, USA: ACM, 2015.

[3] J. Hester, K. Storer, L. Sitanayah, and J. Sorber, "Towards a Language and Runtime for Intermittently-Powered Devices," *sleep*, vol. 9, 2016.

[4] Free and Hanseatic City of Hamburg — State Ministry of Economic Affairs, Transport and Innovation, " Verkehr 4.0 - ITS-Strategie für Hamburg," http://www.its2021.hamburg/downloads/ITS%20Strategie%20Hamburg.pdf, May 2016.

[5] L. Hanschke, J. Heitmann, and C. Renner, " On the Feasibility of WiFi-Enabled and Solar-Powered Sensors for Smart Ports ," in *Proceedings of the 15th GI/ITG KuVS Fachgespräch "Sensornetze"* , ser. FGSN, Augsburg, Germany, September 2016.

[6] C. Park and P. H. Chou, "Ambimax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes," in *Sensor and Ad Hoc Communications and Networks, SECON'06.*, vol. 1. IEEE, 2006.

[7] L. Hanschke, J. Heitmann, and C. Renner, " Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports ," in *Proceedings of ENSsys*, Stanford, CA, USA, November 2016.

[8] Espressif Inc., *ESP8266EX Datasheet*, Apr. 2016, rev. 5.4. [Online]. Available: https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[9] IEEE 802.11 Working Group, "IEEE Standard for Information Technology — Telecommunications and Information Exchange Between Systems — Local and Metropolitan Area Networks — Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std*, vol. 802, no. 11, 2016.

[10] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time Scheduling for Energy Harvesting Sensor Nodes," *Real-Time Systems*, vol. 37, 2007.

[11] C. Renner, S. Unterschütz, V. Turau, and K. Römer, "Perpetual Data Collection with Energy-Harvesting Sensor Networks," *Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, Nov. 2014.

[12] H. Zhai, Y. Kwon, and Y. Fang, "Performance Analysis of IEEE 802.11 MAC Protocols in Wireless LANs," *Wireless Communications and Mobile Computing*, vol. 4, no. 8, 2004.