Using deployed Wi-Fi Access Points to enhance asynchronous wake up protocols

Daniel Camps-Mur

***Abstract*—Asynchronous wake up protocols determine when a mobile device needs to be awake or sleeping in order to minimize the time needed to discover other devices in range, while operating with a given duty cycle that limits energy consumption. In this letter we introduce a technique that allows asynchronous wake up protocols implemented on devices with a Wi-Fi radio, to use available Wi-Fi APs as external reference in order to improve performance. We present a simulative performance evaluation that demonstrates how, with a minimal impact on power consumption, our technique boosts the contact opportunities delivered by existent protocols.**

I. Introduction

The increase of mobile devices with Wi-Fi interfaces is expected to give raise to novel proximity based applications. For instance, social applications could notify users when they are in proximity of any of their friends or contacts, and stores could broadcast advertisements or coupons that could be picked up by users as they walk by.

These applications require mobile devices to be able to discover other devices with limited contact opportunities, as users may quickly walk out of each other's range. In addition, wireless interfaces like Wi-Fi may also have a significant impact on battery life. Therefore, *asynchronous wake up protocols* have been proposed that, according to a target duty cycle, control when a device needs to be sleeping or awake in order to minimize the time required to discover nearby devices. Most proposed asynchronous wake up protocols, like those in [1] and [2], assume that devices are not synchronized. However, the RBTP protocol [3] was recently proposed that improves upon existing asynchronous wake up protocols by letting smartphones synchronize every few hours with NTP servers over the Internet. Notice though that in order to benefit from RBTP a device needs to have Internet access. Therefore, in this letter we propose a new technique that allows devices to tap on an external reference source in order to enhance performance, but unlike RBTP, our technique only requires a Wi-Fi radio, instead of Internet access, and can thus be implemented in a wider range of devices. Our technique is also inspired by the WizSync protocol proposed in [4], which uses periodic Beacon transmissions coming from Wi-Fi Access Points (APs) to synchronize Zigbee nodes in a sensor network. In particular, our main contributions in this letter are: i) we optimize the concept presented in [4] to allow Wi-Fi devices, instead of Zigbee devices, to use APs as external reference to derive rendezvous slots that improve discoverability, ii) we analyze how a novel protocol proposed in this letter and the protocols proposed in [1] and [2] can benefit from our technique, and iii) we present a simulative performance evaluation that illustrates the performance gains achieved with our approach.

The author is with NEC Network Laboratories, Kurfuersten-Anlage 36, Heidelberg (Germany). E-mail: camps@neclab.eu

II. Using deployed Wi-Fi APs for device rendezvous

A. System Model

Let us consider a set of moving nodes[1] in a given geographical area and time being slotted with slots of duration τ seconds. An asynchronous wake up protocol runs independently in each node, and at each time slot s_k executes a function $f(s_k) \in \{1, 0\}$ to decide if the node remains awake, or if it sleeps. At the beginning of an awake slot a node broadcasts a small advertisement packet that announces its presence to other awake nodes. If during an awake slot a node receives advertisement packets from other nodes we say that a *contact* event has occured. In addition, the clocks between different nodes are not synchronized, which implies that slot boundaries at different devices are not aligned and that clocks can drift over time. Thus, the performance of an asynchronous wake up protocol is characterized by the trade off between energy consumption and the time required to discover a nearby device. Finally, we assume that devices transmit using a standard Wi-Fi radio over a given Wi-Fi physical channel.

B. Deployed APs as external source of synchronization

If nodes are equipped with a Wi-Fi radio, then they are able to receive transmissions from surrounding Wi-Fi APs, which tend to be deployed in high densities in metropolitan areas [6].

In order to be discoverable, Wi-Fi APs transmit Beacon frames and Probe Responses, which contain the MAC address of the AP and a timestamp value with the AP's local time [5], and can be received by *any* Wi-Fi device, even those not connected to that AP. Thus, the core idea proposed in this letter is to use the timestamps included in Beacon frames and Probe Responses as external source of synchronization to enhance the performance of asynchronous wake up protocols. In particular, a Wi-Fi device running an asynchronous wake up protocol may receive Beacons and Probe Responses, upon which the device stores the AP's MAC address, the contained timestamp $\mathbf{t}_{AP}$, and the time according to the device's local clock when the frame has been received $t_{AP_{seen}}$; where time

[1] In this letter we use node and device interchangeably.

is represented in Wi-Fi as a 64-bit counter local to each device that increases with a granularity of microseconds.

Let us now assume that all nodes share a pre-configured time reference, hereafter referred to as $\mathbf{t}_{ref}$ defined as a $n \leq 64$ bit sequence. Thus, when the n least significant bits of the AP's timestamp equal $\mathbf{t}_{ref}$, we say that a *rendezvous slot* has occurred according to the time base of that AP, where rendezvous slots will repeat every $T_{rdv} = 2^n$ microseconds. Therefore, at any time t a device can compute the time lag until the next rendezvous slot for a given known AP as:

$$\Delta_{rdv\_slot}(t) = (\mathbf{t}_{ref} - \mathbf{t}_{AP}|_n + (t - t_{AP_{seen}})) \mod T_{rdv} \quad (1)$$

Where $\mathbf{t}_{AP}|_n$ is the value of the n least significant bits of the AP's timestamp field. Applying this method, nearby devices under the coverage of the *same* AP, can independently derive rendezvous slots where discovery can happen. If a device hears multiple Wi-Fi APs, different rendezvous slots would be defined according to the time base of each AP, therefore a heuristic is needed in order to let devices select the same AP to derive the same rendezvous slots; a simple heuristic that will be studied in this letter is selecting the AP with the lowest MAC address. Note that the proposed mechanism requires no changes in APs or in the 802.11 standard, and can therefore be readily implemented in any Wi-Fi device.

III. Enhancing asynchronous wake up protocols

Next, we illustrate how the presented technique can be applied to improve the performance of the following protocols: i) a random wake up protocol, which is a simple protocol proposed in this letter, ii) the Grid Quorum protocol proposed in [1], and iii) the U-Connect protocol proposed in [2]. It should be understood though that the same techniques can be applied to any asynchronous wake up protocol that follows the description introduced in section II-A.

A. Enhancing a Random wake up protocol

A simple Random wake up protocol can be defined with two parameters: i) a slot size τ, and ii) a wake up period of duration $T = M$ slots. Thus, for each wake up period a node randomly selects one slot within that period to remain awake, whereas it sleeps for the rest of the period, i.e. $f(s_k) = 0,\ \forall k \neq k^*$, and $f(s_{k^*}) = 1$, where $k^* = uniform(0, M-1)$ is computed randomly at the beginning of each period. It is easy to see that the duty cycle of a device operating under this protocol is $d = \frac{1}{M}$, and the probability of two devices discovering each other before N period times is $P\{discovery\ time < NM\tau\} = 1 - (1 - \frac{1}{M})^N$. This protocol is depicted in the upper left diagram of Figure 1.

Let us now describe how this protocol can be enhanced by using the method described in section II-B. First, consider that periodic rendezvous slots with respect to each AP are defined with period $T_{rdv} = M\tau$, hence $\mathbf{t}_{ref}$ is a bit sequence of $n = \lceil \log_2 M\tau \times 10^6 \rceil$ bits. Thus, with the received Probe Responses and Beacon frames, a device creates a list of the Wi-Fi APs in its surroundings with the corresponding timestamps, and the times according to the device's local clock where the timestamps were received. An AP stored in this list is removed after a given timeout without hearing any frame from this AP. At the beginning of each period a device selects its known AP with the lowest MAC address, and instead of picking a random slot it computes the slot where to stay awake as $k^* = \lfloor \frac{\Delta_{rdv\_slot}(t)}{\tau} \rfloor$, where t represents the devices's local time and $\Delta_{rdv\_slot}(t)$ is computed according to Eq. 1. Hence, applying this method two devices under the coverage of the same AP will always select overlapping awake slots at each period time. Notice that if two devices latch on to different APs, they might not be able to find each other. This problem is due to our simple heuristic of selecting only the AP with the lowest MAC, and could be addressed with enhanced heuristics that make a device rotate accross multiple APs. We leave the study of these heuristics as future work, and in this letter we focus on the simple version of the random wake up protocol as studies like [6] and [4] have suggested that in practice a large number of devices may see a common AP. This method is illustrated in the lower left diagram of Figure 1, from where it can be seen that when devices do see the same AP the worst case discovery delay is bounded to M slots. Finally, if a device does not see any AP, or it is not discovering other devices while latched on to a given AP, it returns to the default mode where the awake slot is computed randomly.

B. Enhancing the Grid Quorum protocol

In the Grid Quorum protocol [1], devices operate with a period $T = M$ slots, and $f(s_k)$ is derived by first aligning the M slots in a square grid of side $\sqrt{M}$ (M needs to be a perfect square). Then, each device independently selects a row and a column on that grid, stays awake during the corresponding slots, and sleeps during the rest. This protocol has a duty cycle of $d = \frac{2\sqrt{M}-1}{M}$, a worst case discovery delay of M slots, and is illustrated in the upper middle diagram of Figure 1.

The method described in section II-B can be used to improve the performance of Grid Quorum in the following way. Let $T_{rdv} = \sqrt{M}\tau$ seconds, so that periodic rendezvous slots occur at the same frequency than wake up slots in the Grid Quorum protocol. Thus, as in section III-A, devices maintain a list of the discovered APs and their respective timestamps. Notice that since Grid Quorum only imposes the sequence of awake and sleeping states, a device is free to align its awake states with the rendezvous slots of a given AP. In particular, when a device discovers an AP with a lower MAC address than the APs it already knew, or after a given timeout ($T_{resch} >> T_{rdv}$) to account for clock drift, the time of its next awake slot is rescheduled in order to make it coincide with the next rendezvous slot defined according to the known lowest MAC AP, i.e. the next awake slot is scheduled after a delay $\Delta_{rdv\_slot}(t)$, where t represents the devices's local time and $\Delta_{rdv\_slot}(t)$ is computed according to Eq. 1. If the list of known APs is empty a device operates according to the default Grid Quorum protocol. Notice that this enhanced version of the Grid Quorum protocol does not suffer from the same problem as the Random wake up protocol, because devices latched on to different APs (or not latched on to any AP) are still able to find each other. The reason is that Grid

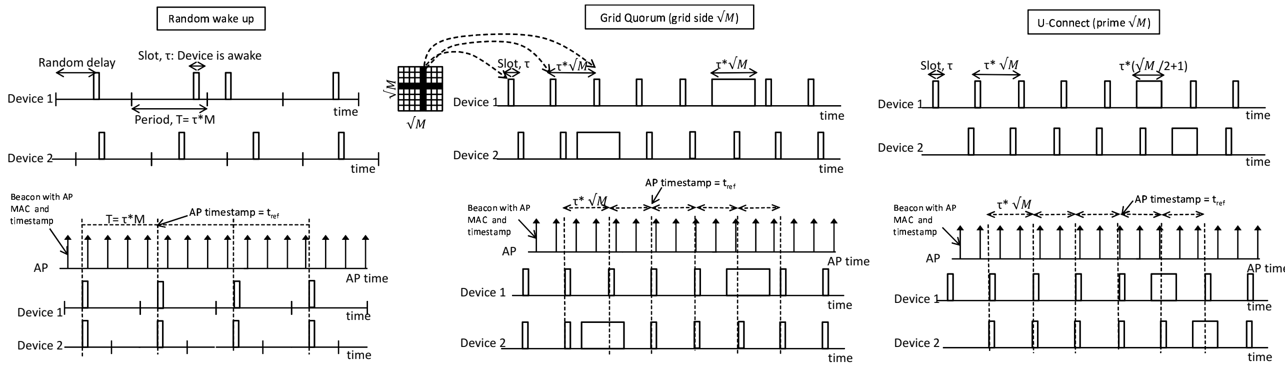


Fig. 1. Upper row: Default protocol operation, lower Row: Enhanced Protocol operation. Left: Random wake up, middle: Grid Quorum, right: U-Connect.

Quorum guarantees that any two devices that are out of synch, as two devices latched on to two different APs, will always find each other within a time $T = M$ slots. Looking at the lower middle diagram of Figure 1 we can see that if devices latch on to the same AP and align their awake slots, then the worst case discovery delay reduces from M slots to $\sqrt{M}$ slots.

C. Enhancing the U-connect protocol

The U-Connect protocol [2] follows the same principle as Grid Quorum and works in the following way. Devices operate with a period $T = M$ slots, being $\sqrt{M}$ a prime number. Thus, in U-Connect $f(s_k) = 1$, if $k \mod \sqrt{M} = 0$ or $0 \leq k \mod M < \frac{\sqrt{M}+1}{2}$, and $f(s_k) = 0$ otherwise. Hence, this protocol results in a duty cycle of $d = \frac{\frac{\sqrt{M}+1}{2}+\sqrt{M}}{M}$, and a worst case discovery delay of M slots. The operation of this protocol is depicted in the upper right diagram of Figure 1, where it can be seen that within each period T there is a longer awake interval that ensures that other devices are discovered regardless of the relative phase of their awake slots.

U-Connect can be optimized following the same principle used for Grid Quorum with $T_{rdv} = \sqrt{M}\tau$, i.e. letting devices align their awake slots with the rendezvous slots defined according to the time base of a given AP. The protocol is depicted in the lower right diagram of Figure 1, where we can see how for devices under the same AP worst case discovery delay reduces to $\sqrt{M}$ slots, while for the rest of devices is still $T = M$ slots. For the sake of space we refer the reader back to section III-B for more details on this method.

IV. Performance Evaluation

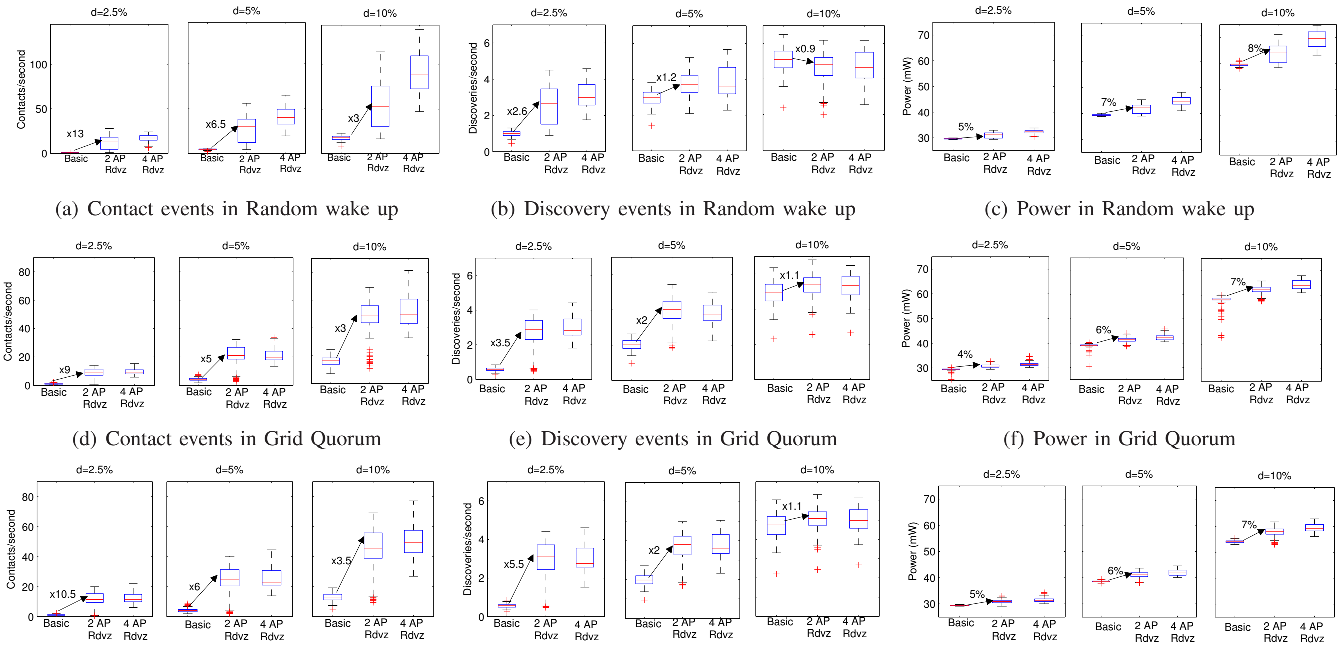
Our evaluation is based on packet level simulations using OPNET [7], to accurately model the protocols involved, and Mobireal [8] to generate mobility models that capture realistic densities of moving pedestrians. In particular, our evaluation scenario consists of 200 devices moving through Osaka downtown at pedestrian speeds, with the same scenario used in [8].

In order to evaluate the protocols under study, we look at three different metrics: i) *contact* events (see section II-A), ii) *discovery* events, where a discovery event is a contact event with a device that has not been seen in the last 10 seconds[2], and iii) the average power consumed by a device while running each protocol. Our power model computes the time spent transmitting, receiving, sensing the channel and sleeping, and weights them by the power consumed by a commercial Wi-Fi chipset in each state. In addition, we evaluate the performance of each protocol with three different duty cycles, i.e. $d = \{2.5\%, 5\%, 10\%\}$ and three different configurations: the basic protocol (labeled *Basic*), and the protocol enhanced with our proposed technique considering 2 or 4 deployed APs (labeled *2/4 AP Rndvz*), randomly deployed in our scenario. Note that four was found to be the most common number of overlapping APs in the same channel in [6]. In addition, the slot size is set to $\tau = 25$ms, $T_{resch} = 60$s, the list of known APs is refreshed every 10s, the transmission rate is fixed to 6Mbps, Probe Requests are sent every 5s, and for each duty cycle $d = \{2.5\%, 5\%, 10\%\}$ the M parameter in each protocol is configured as: i) in Random wake up $M = \{40, 20, 10\}$, ii) in Grid Quorum $\sqrt{M} = \{80, 40, 20\}$, and iii) in U-Connect $\sqrt{M} = \{61, 31, 17\}$. Finally, each experiment simulates 3000s and is repeated three times.

Figure 2 depicts for the Random wake up (figures 2(a), 2(b), 2(c)), Grid Quorum (figures 2(d), 2(e), 2(f)) and U-Connect protocols (figures 2(g), 2(h), 2(i)): i) contact events per second (left column), ii) discovery events per second (middle column) and the average device power consumption (right column). Results are presented as boxplots of each metric's cumulative distribution function. In addition, each figure is composed of three subfigures, one for each considered duty cycle ($d = \{2.5\%, 5\%, 10\%\}$), and each subfigure contains three boxplots corresponding to the basic protocol performance and the performance with our proposed technique with two or four APs in the same channel.

Looking at Figure 2 we can see how our technique boosts the number of contact events for all protocols, e.g. a 13 fold increase in the case of the Random wake up protocol with duty cycle $d = 2.5\%$ (signaled with an arrow). The number of discovery events also increases substantially especially with small duty cycles, e.g. 5 fold increase in U-Connect when $d = 2.5\%$. This performance increase confirms that in our scenario many devices are able to see a common AP and align their awake slots, which increases the number of contact events of devices in range, and decreases the delay required to discover a new device (see Fig. 1). In addition, the number

[2]Discovery events are considered to evaluate if the contact events in a given protocol are with different devices or with the same device.



(a) Contact events in Random wake up (b) Discovery events in Random wake up (c) Power in Random wake up

(d) Contact events in Grid Quorum (e) Discovery events in Grid Quorum (f) Power in Grid Quorum

(g) Contact events in U-Connect (h) Discovery events in U-Connect (i) Power in U-Connect

Fig. 2. Performance of the Random wake up, Grid Quorum and U-Connect protocols, when using deployed APs as external source of synchronization.

of contact events increases more than the number of discovery events because, while under the coverage of the same AP two devices will enjoy contact opportunities in each awake slot, whereas discovery events can only occur when a new device appears, which is a more seldom event. The gains achieved by our technique decrease though when the allowed duty cycle increases, because traditional protocols improve with larger duty cycles. We can also see in Figure 2 how the performance of our technique generally tends to improve when the number of deployed APs increases, since in that case it is more likely that two devices will be under the coverage of the same AP. The effect of the number of APs though is very dependent on the particular deployment and we leave as future work the study of other realistic deployments. In addition, the performance of different protocols in terms of contact and discovery events differs because, although configured with the same duty cycle, the Random protocol schedules awake slots more often than the other protocols. Finally, as future work we identify the following optimizations: i) better AP selection heuristics to reduce the probability of devices latched on to different APs not finding each other, and ii) jitter the transmission times within a rendezvous slot in order to reduce collisions, which could be problematic in dense scenarios with hidden nodes and devices using slow transmission rates.

Regarding power consumption, we can see in figures 2(c), 2(f) and 2(i) how applying our technique results in a penalty of less than 8% in terms of power, mainly due to the additional Probe Requests and Responses. This penalty could be reduced by increasing the interval between Probe Requests, which could however impact the discovery of APs. Nevertheless, we believe that the significant gains in terms of contact events provided by our technique should in most implementations compensate for a small penalty in power consumption.

V. Conclusions

In this letter we have proposed a technique that allows devices with a Wi-Fi radio to derive common rendezvous slots by means of passively overhearing the transmissions of already deployed Wi-Fi APs. Our technique does not require any modification to existing 802.11 standards, and as we have shown, it can be applied to significantly boost the number of contact and discovery events experienced under several asynchronous wake up protocols, with a minimal impact on power consumption.

References

[1] Tseng, Yu-Chee; Hsu, Chih-Shun; Hsieh, Ten-Yueng; , *Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks,* INFOCOM, 2002 Proceedings IEEE, vol. 1, no., pp.200-209, 07 November 2002.

[2] Kandhalu, Arvind, Karthik Lakshmanan, and Ragunathan Raj Rajkumar. *U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol.* Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. ACM, 2010.

[3] Dong Li, Prasun Sinha,; *RBTP: Low Power Mobile Discovery Protocol through Recursive Binary Time Partitioning,* IEEE Transactions on Mobile Computing, 26 Nov. 2012.

[4] Tian Hao; Ruogu Zhou; Guoliang Xing; Mutka, M.; *WizSync: Exploiting Wi-Fi Infrastructure for Clock Synchronization in Wireless Sensor Networks*, Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd , vol., no., pp.149-158, Nov. 29 2011-Dec. 2 2011

[5] 802.11-2012 - *IEEE Standard for Information technology–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*.

[6] Castignani G., Loiseau L., Montavont N., *An Evaluation of IEEE 802.11 Community Networks Deployments*, Information Networking (ICOIN), 2011 International Conference on. IEEE, 2011.

[7] OPNET, http://www.opnet.com.

[8] K. Maeda et. al., *Getting Urban Pedestrian Flow from Simple Observation: Realistic Mobility Generation in Wireless Network Simulation*, in Proceedings of the 8th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM2005), 2005.