Maximum-Likelihood Detection for Energy-Efficient Timing Acquisition in NB-IoT

Harald Kröll[†], Matthias Korb[†], Benjamin Weber[‡], Samuel Willi[‡] and Qiuting Huang[‡]

[†]ACP AG, Zurich, Switzerland

[‡]Integrated Systems Laboratory, ETH Zurich, Switzerland

{kroell,mkorb}@newacp.ch {weberbe,huang}@iis.ee.ethz.ch,

Abstract—Initial timing acquisition in narrow-band IoT (NB-IoT) devices is done by detecting a periodically transmitted known sequence. The detection has to be done at lowest possible latency, because the RF-transceiver, which dominates downlink power consumption of an NB-IoT modem, has to be turned on throughout this time. Auto-correlation detectors show low computational complexity from a signal processing point of view at the price of a higher detection latency. In contrast a maximum likelihood cross-correlation detector achieves low latency at a higher complexity as shown in this paper. We present a hardware implementation of the maximum likelihood cross-correlation detection. The detector achieves an average detection latency which is a factor of two below that of an auto-correlation method and is able to reduce the required energy per timing acquisition by up to 34%.

I. INTRODUCTION

Various estimates predict tens of billions devices connected to the Internet in 2020 in what is called the Internet of Things (IoT). IoT does not only take place in our homes or in areas which are covered by WiFi and other low-range networks, but also in remote places which are only covered by cellular or satellite networks. Cellular network coverage is almost ubiquitous and does not depend on proprietary end-user infrastructure.

To realize an IoT in which the requirements for low-power, low-cost, and extended-coverage IoT devices will be met, the 3GPP consortium agreed on an LTE-Release-13 extension called Narrow Band (NB)-IoT or LTE Cat-NB1 [1]. On the downlink and uplink side NB-IoT mainly reuses LTE technology. However, cell search and timing acquisition procedures have undergone major adaptions to fit into the narrow 200 kHz bandwidth and to meet coverage extension requirements.

The energy efficiency of an NB-IoT device preferably implemented as a system-on-chip is of great importance to achieve years of battery life as aimed for emerging cellular IoT standards. Besides the power amplifier for the uplink, which holds the lions share of overall power consumption, it is well known that the downlink baseband signal processing consumes only a fraction of the RF-transceiver power in receive mode [2]. This appears because RF-transceivers are dominated by analog integrated circuits whose power consumption especially does not scale as well with the CMOS technology feature size as it scales for the digital integrated baseband circuits. Therefore, NB-IoT has undergone various simplifications to allow energy-efficient implementations. Significant

bandwidth reduction to 200 kHz was the main simplification of NB-IoT compared to the minimal bandwidth requirement of 1.4 MHz in LTE. But, the RF-transceiver power consumption is rather proportional to the carrier frequency and to sensitivity requirements than bandwidth. While adjacent channel leakage ratio was reduced by 5dB compared to 1.4MHz LTE [4], the maximum carrier frequency is only slightly reduced from 2.6 to 2.2 GHz. Thus, the RF-transceiver is still dominating the downlink power consumption. However, power consumption of digital baseband processing scales well with bandwidth, which is useful for NB-IoT timing acquisition.

Besides data decoding timing acquisition is the most complex baseband task along the downlink path [3]. Hereby energy-efficient timing acquisition is important because timing acquisition has to be done frequently, mainly for two reasons: Firstly, NB-IoT is designed for the exchange of short messages, thus devices are in deep sleep mode most of the time and wake up e.g. every hour for a short period of time to receive and transmit a few hundred bytes. To ensure years of battery life, circuits providing accurate timing are turned off during deep sleep mode, which requires timing acquisition after every wake-up. Hereby timing acquisition has a relatively large share on the short reception interval, which requires an energy-efficient implementation. Secondly, NB-IoT is likely to be used on vehicles and drones where devices are prone to timing synchronization loss due to their relatively high mobility and the absence of handover capability in NB-IoT.

For timing acquisition a periodically transmitted a priori known Narrowband Primary Synchronization Sequence (NPSS) has to be detected [5]. The latency of a successful timing acquisition (NPSS detection) is the relevant performance metric, because it determines how long the RF-transceiver, which consumes the major part of the power, has to be turned on to receive data. Therefore, using low-complexity NPSS detectors which achieve suboptimal performance can be disadvantageous for the overall downlink energy efficiency.

Contributions: We present a maximum-likelihood (ML) NPSS detector which achieves an average timing acquisition latency of 140 ms (in-band deployment, TU1.2 channel, SNR = -12.6 dB). Our ML detector is based on cross-correlation metrics which are computed in frequency domain via the overlap-save method. The detector has high computational complexity but allows to reduce the required energy by up to 34% per timing acquisition for state-of-the-art RF-transceivers.

Fig. 1. NPSS and NSSS resource mapping onto an NB-IoT frame.

Fig. 2. ML function over normalized frequency offset.

II. TIMING ACQUISITION IN NB-IoT

The first step after power-on (or after a wake-up from a sleep cycle) of an NB-IoT device is the detection of an NB-IoT capable base-station. In case such a base-station exists, the receiver does not know which OFDM symbol of the frame is currently transmitted. On top of that, the frequency relation between the base-station and the local receiver clock is also unknown. In NB-IoT as well as in other LTE device categories, the detection of a suitable base-station and the estimation of the timing and frequency offset is based on two periodically transmitted sequences: the NPSS and the Narrowband Secondary Synchronization Sequence (NSSS). While the NPSS is transmitted repeatedly every sub-frame of length 10 ms, the NSSS is repeated in every second sub-frame as shown in Fig. 1. For NB-IoT the transmitted NPSS is identical in every sub-frame for all base-stations. In contrast, the NSSS depends on the base-stations cell ID and is scrambled with a frame-dependent sequence code.

The NPSS is used to verify the existence of an NB-IoT capable base-station. Additionally, it enables the estimation of the frequency offset and timing offset with respect to the sub-frame boundary. The NSSS is then used to detect the frame boundary and cell ID.

The NPSS is defined in frequency domain as a Zadoff-Chu sequence of length 11 for each sub-carrier index n given by

$$S[n] = e^{\frac{-j5\pi n(n+1)}{11}} c[l], \quad n = 0 \ldots 10 \qquad (1)$$

where $c[l]$ is an element of the code cover vector

$$\boldsymbol{c} = [1, 1, 1, 1, -1, -1, 1, 1, 1, -1, 1],$$

with l being the symbol index in a sub-frame. This sequence is mapped to 11 subsequent OFDM symbols each consisting of 12 OFDM sub-carriers holding one copy of the NPSS.

After zero-padding each of the 11 copies to 128 symbols, time-domain conversion, and cyclic-prefix insertion of either length 9 or 10, the NPSS results in 1,508 time domain samples.

With a sub-frame length of 10 ms and a sampling rate of 1.92 MHz 19,200 samples need to be captured in order to get exactly one copy of the NPSS. As the sub-frame boundary is unknown, the NPSS can start at any of the 19,200 samples. One task of the receiver is to estimate the beginning of the NPSS to acquire sub-frame boundary timing information. In addition, an NB-IoT device has a random frequency offset because the crystal oscillator on the device is not yet tuned

after power-on or after wake-up from a sleep cycle. This heavily affects the detection complexity because the device needs to analyze various frequency-offset candidates within a specified boundary, as well. To reduce the complexity it is possible to perform a coarse frequency and timing offset estimation on a down-sampled version of the received signal. For example in [6] the coarse estimation is done via auto-correlations at a sampling frequency of 240 kHz. Then, one sub-frame consists of only 2,400 samples.

III. ML TIMING ACQUISITION WITH CORRELATIONS

There are two main algorithms to perform timing acquisition, namely auto-correlation and cross-correlation. While auto-correlation is the only option if the transmitted, periodic sequence is unknown, for NPSS detection both algorithms can be applied as the transmitted sequence is known to the receiver. Auto-correlation approaches are in general more hardware efficient than cross-correlation approaches. But, since the auto-correlation algorithm does not exploit the fact that the transmitted sequence is known, its performance is sub-optimal. In fact, cross-correlation detectors are ML detectors [7]. This is the reason, why many applications like radar systems or GPS receivers use a cross-correlation for signal detection [8]. In this paper we focus on low latency rather than low complexity. Thus, the ML detector [7] (Page 244), which projects the received signal vector onto each of the N_{f} possible frequency candidates, is a viable option for NPSS detection.

The NPSS ML detector correlation metrics are given by

$$C\left(\boldsymbol{r} \mid \theta, f_{\mathrm{o}}\right) = \sum_{k=\theta}^{\theta+189} r[k+\theta]s^*[k]e^{-j2\pi f_{\mathrm{o}}k/f_s}, \qquad (2)$$

where the received signal vector

$$\boldsymbol{r} = [r[0]\ r[1]\ \ldots\ r[189-1]]^{\mathrm{T}}$$

has a sampling rate of $f_s = 240$ kHz and $s[k]$ for $k = 0 \ldots 188$ is the time domain NPSS sequence given in Eq. (1) at 240 kHz.

The ML function $C(\boldsymbol{r}' \mid \theta = 0, f_{\mathrm{o}})$ for a distortion-free received signal vector $\boldsymbol{r}'$ over the frequency offset f_{o} is plotted in Fig. 2. The ML frequency- and timing-offset estimation $\hat{f}_{\mathrm{o}}$ and $\hat{t}_{\mathrm{o}}$ can then be calculated according to

$$(\hat{f}_{\mathrm{o}}, \hat{t}_{\mathrm{o}}) = \arg\max_{f_{\mathrm{o}}, t_{\mathrm{o}}} \left\{C(\boldsymbol{r} \mid f_{\mathrm{o}}, t_{\mathrm{o}})\right\}.$$

Fig. 3. Signal processing scheme of the ML NPSS detector which includes fractional-frequency and coarse-timing offset estimation. The right sub-figure shows correlation computation with the overlap-save method.

Hereby different time offset hypotheses t_o, which correspond to sub-frame boundaries, have to be evaluated by cross-correlating the received samples in the correlation window with the known NPSS.

In addition, correlations are performed for every N_f frequency offset hypothesis f_o, which defines the range of frequency offsets the detector shall support. The minimum frequency grid spacing is defined by the 240 kHz sampling rate and the FFT and IFFT size which trades off computational complexity, memory requirements, and processing delay for estimation accuracy. Larger FFT sizes with smaller grid spacings improve frequency offset estimation accuracy but have a longer delay and require more memory. An FFT size of 1,024 results in a grid spacing of 234 Hz which is sufficient for NPSS detection and was therefore chosen in this work. In addition, the width of the correlation peak of Fig. 2 allows to take every fourth grid point only, while still covering 93% of the peak amplitude. N_f trades off the minimal observed height of a correlation peak against computational complexity and memory size. It is a design parameter, which can be chosen to match the accuracy of the underlying crystal oscillator. Choosing $N_f = 31$ leads to a frequency-offset range of $31 \cdot 4 \cdot 234\,Hz$ which allows to compensate ± 14.5 kHz.

In every 10 ms frame we receive $N_s = 2,400$ samples and the length of the NPSS in time domain is $N_p = 189$ samples. In total $N_s \cdot N_f$ cross-correlations of length N_p are required as shown in the left part of Fig. 3. Considering $N_f = 31$ different frequency candidates a total of 74,400 cross-correlations need to be performed every 10 ms, which is impractical for NB-IoT devices.

However, the computational complexity can be significantly reduced when using an overlap-save (OLS) method [9]. This method is well established especially for discrete convolutions but it can also be applied to cross-correlations. By applying OLS to the NPSS cross-correlation, the input stream is divided into overlapping sequences of length N as illustrated in the right part of Fig. 3. The number of overlapping samples depends on the NPSS length and is chosen to be $N_O = 188$. Afterwards, the block-wise cross-correlation with the different frequency-offset candidates is performed, which can be done in frequency domain. The main benefit of this method is that a cross-correlation in time domain is replaced by a point-wise multiplication in frequency domain. Additionally, the generation of the NPSS reference signals in frequency domain gets simplified: Different frequency offsets relate to cyclic shifts which can be easily implemented in hardware.

IV. COMPLEXITY AND PERFORMANCE

As the proposed cross-correlation-based algorithm is an ML detector, the complexity is expected to be significantly higher compared to the low-complexity auto-correlation method. On average for each received block of size $N - N_O$ a single N-point FFT, N_f point-wise complex multiplications of a vector of length N, and N_f N-point IFFT operations need to be performed. Choosing an FFT size of $N = 1,024$ the number of real additions and multiplications can be estimated to 135.0 and 135.4 MOPS, respectively leading to an overall computational complexity of 270.5 MOPS. Thus, the computational effort per sub-frame of the ML detector is roughly 10x higher than the auto-correlation timing acquisition [6].

Fig. 4.   Timing acquisition latency of cross-correlation NPSS detector.

TABLE I
NPSS DETECTOR CHARACTERISTICS IN TWO CMOS TECHNOLOGIES.

CMOS technology	SMIC 130 nm	GF 28 nm
Synthesized Cell Area	$3.34\,\mathrm{mm}^2$	$0.22\,\mathrm{mm}^2$
Voltage	1.2 V	1.0 V
kGE	735	600
est. P_{ML}	38 mW	2.5 mW

The performance in terms of timing-acquisition latency is shown in Fig. 4 for in-band deployment which has the most demanding SNR requirement of -12.6 dB and beyond. For the simulations the TU1.2 channel model was used and the threshold was set to achieve a false-alarm rate of 1%. The OLS detector achieves a latency of 400 ms, whereas the auto-correlation detector of [6] takes 620 ms to achieve a 90% hit rate. The average detection latency is 140 ms which is roughly a factor of two below the value of [6].

V. HARDWARE IMPLEMENTATION

A block diagram of the cross-correlation NPSS detector is shown in Fig. 5. The main computational elements are the FFT and IFFT blocks with a required throughput of $2.87/10\,\mathrm{ms} = 287.1/\mathrm{s}$ and $890.0/\mathrm{s}$ FFT and IFFT computations or 1.5 and 45.6 million radix-2 operations per second, respectively. Even for the more demanding IFFT it is possible to reuse a single radix-2 instance for all IFFT operations when assuming typical VLSI clock frequencies. So, for the FFT as well as for the IFFT block a single radix-2 in-place architecture is sufficient.

The FFT is designed to include a RAM holding 1,360 complex samples which is larger than N. The reason for this is two-fold: Firstly, the FFT operates on 1,024 complex

words, but the unaltered 188 overlap samples need to be stored for the next FFT computation, as well. Secondly, during the FFT operation further inputs $r[k]$ need to be stored in the memory. Furthermore, a single-port RAM has been chosen, which minimizes the storage area. The introduced memory-bandwidth bottleneck limiting the throughput to one radix-2 operation every 4 clock cycles is tolerable due to the very low throughput requirements of the FFT.

In contrast such an architecture would not be sufficient to meet the throughput requirement of the IFFT. Here, the memory bandwidth has to be $4\times$ higher to support a throughput of one radix-2 operation every cycle. Thus, the memory in the IFFT block is split into four banks each still being a single-port RAM to minimize storage area. Memory access conflicts are avoided by assuring that every two subsequent radix-2 operations do not access the same register banks. After processing the FFT, the correlations in frequency domain, and the $N_f = 31$ IFFTs for each received block of length N the results are non-coherently combined with previous correlation results. The size of the memory holding the intermediate, non-coherently combined correlation results is reduced by down-sampling the correlation results by a factor of 2 as proposed in [6]. After the processing of a sub-frame, a peak-detection is used to decide, whether the NPSS sequence was found. Rather than using a simple peak-to-average ratio an analysis of the four largest correlation results is considered which improves the detection probability when combining correlation results of multiple sub-frames. Also, the existence of side-peaks (Fig. 2) requires a more sophisticated peak detection as a simple peak-to-average ratio would lead to many false detections.

We implemented the detector in VHDL and performed synthesis experiments in SMIC130 and GF28 CMOS technology targeting a clock frequency of 62 MHz. The key characteristics of the detector are give in Table I.

With $N_f = 31$ a correlation RAM with 334 kbit is required. This is the largest memory in the design and occupies 54% of the entire area. However since this memory is only used for NPSS detection it can be easily shared with other building blocks. The implementation also includes the fine frequency- and timing-offset estimation as proposed in [6].

The power consumption of the detector was estimated by using Cadence® tools from post-synthesis netlist and the value change dump file to 38 mW (1.2V, TT, 25C) and 2.5 mW (1.0V, TT, 25C) for the 130- and 28-nm technology, respectively.

Fig. 5.   Block diagram of NPSS detector.

VI. ENERGY EFFICIENCY

The energy of timing acquisition is given by the power of the detector and the RF-transceiver in receive mode times the latency t. Given the energy of the ML approach and the auto-correlation (AC) approach for a certain RF-transceiver power P_{RF} we compute the savings according to

$$\Delta E \left[\%\right] = 100 \left[1 - \frac{(P_{RF} + P_{ML})t_{ML}}{(P_{RF} + P_{AC})t_{AC}} \right].$$

For the AC timing acquisition we account for a power of $P_{AC} = \frac{P_{ML}}{10}$ because the arithmetic load is about $10\times$ below the arithmetic load of the ML approach. However it shall be denoted that this factor is dependent on the implementation.

In Fig. 6 the energy saving $\Delta E \left[\%\right]$ per timing acquisition is plotted over the power consumption of the RF-transceiver P_{RF} [W] for the latency of the ML detector ($t_{ML} = 400$ ms) and the AC detector ($t_{AC} = 620$ ms) in [6].

Fig. 6. Energy savings per timing acquisition for the ML detector with 400 ms latency over the auto-correlation detector 620 ms latency with different power consumption values.

Even though the AC detectors show a lower power consumption for NPSS detection (due to their reduced number of additions and multiplications) they do not improve overall energy efficiency because of higher latency. The dotted line shows the maximum possible savings of 35.5%.

The power consumption of RF-transceivers is dependent on multiple factors whose analysis lie beyond the scope of this paper, therefore we consider a broad range of values for RF-transceiver power consumption. The grey rectangle in Fig. 6 indicates the region of interest for NB-IoT dedicated RF-transceivers which lies below the power consumption of conventional LTE and GSM transceivers due to the simplifications made in NB-IoT. Power consumptions of state of the art conventional LTE and GSM transceivers are indicated by the vertical lines in Fig. 6 indicate the power consumption of two reported RF-transceivers [11], [12].

VII. CONCLUSION

The fact that the RF-transceiver dominates downlink power consumption in an NB-IoT device creates design space for dedicated hardware implementations which can execute exhaustive baseband algorithms. Following this guideline we have shown that the computationally complex ML approach for NB-IoT timing acquisition can lead to significant energy savings in NB-IoT devices. The savings were achieved by the low latency of our detector which due to algorithmic transforms based on the OLS method and by targeting a dedicated VLSI implementation shows a relatively low power consumption. We were able to reduce the energy required for a single NPSS detection by 34% for 28 nm CMOS technology and from 9% up to 21% even in a rather mature 130 nm CMOS technology. Future research will address area reductions especially by sharing memory resources with other hardware building blocks.

REFERENCES

[1] Wang, Y-P. Eric, et al. "A Primer on 3GPP Narrowband Internet of Things (NB-IoT)." arXiv preprint arXiv:1606.04171 (2016).

[2] Lin Zhong, Power Consumption by Wireless Communication, Lecture ELEC518, 2011, http://www.ruf.rice.edu/~mobile/elec518/lectures/3-wireless.pdf

[3] Intel Cooperation, R1-156524: On device complexity for NB-IoT, 6.2.6.1, 3GPP TSG RAN WG1 Meeting Nr. 83, Anaheim, USA, 16-20 Nov., 2016

[4] 3GPP TS 36.101 V14.0.0, Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception, July 2016

[5] 3GPP TS 36.211 V13.2.0, Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation, June 2016

[6] Qualcomm Inc., R1-161981: NB-PSS and NB-SSS Design, 2.2.5, 3GPP TSG RAN WG1 NB-IoT Ad-Hoc Meeting, Sophia Antipolis, France, 22-24 March, 2016

[7] John G. Proakis, "Digital Communications", McGraw-Hill series in electrical and computer engineering : communications and signal processing, (2001)

[8] Akopian, David. "Fast FFT based GPS satellite acquisition methods." IEE Proceedings-Radar, Sonar and Navigation 152.4 (2005): 277-286.

[9] Nussbaumer, Henri J. Fast Fourier transform and convolution algorithms. Vol. 2. Springer Science & Business Media, 2012.

[10] Wang, Andrew Y., and Charles G. Sodini. "On the energy efficiency of wireless transceivers." 2006 IEEE International Conference on Communications. Vol. 8. IEEE, 2006.

[11] A. Mirzaie, A. Yazdi, Z. Zhou, E. Chang, P. Suri, and H. Darabi, "A 65 nm CMOS quad-band SAW-less receiver for GSM/GPRS/EDGE," in Symp. VLSI Circuits, 2010, pp. 179180.

[12] L. Sundstrm et al., "A receiver for LTE Rel-11 and beyond supporting non-contiguous carrier aggregation," 2013 IEEE International Solid-State Circuits Conference, San Francisco, 2013, pp. 336-337.