Wake-Up Transceiver Architectures with Symbol Time Estimation Schemes for ElectroMagnetic NanoNetworks

FINAL YEAR PROJECT
ESCOLA TÈCNICA SUPERIOR D’ENGINYERIA DE TELECOMUNICACIÓ DE BARCELONA

Raül Gómez Cid-Fuentes
Advisor: Ian F. Akyildiz
Introduction

Transceiver Architecture for EM Nanonetworks

Symbol Time Estimation

Wake-Up Receiver

Conclusions and Open Issues
Table of Contents

Introduction
- Transceiver Architecture for EM Nanonetworks
- Symbol Time Estimation
- Wake-Up Receiver
- Conclusions and Open Issues
Nanotechnology is enabling the control of matter at an atomic and molecular scale:

At this scale, novel nanomaterials show new properties not observed at the microscopic level which can be exploited to develop new devices and applications.

Fig. 1 - Nanosensor device. [1]

Graphene: a one-atom-thick planar sheet of bonded carbon atoms in a honeycomb crystal lattice.

A prime candidate to become the silicon of the 21st century due to:

- Very High Electron mobility → Supporting fast operating frequencies

- Thermoelectric current effect → Self cooling and heat reabsorption

Fig. 2 - Graphene atomic structure.

Graphene can be used to manufacture novel nano-antennas with atomic precision.

- New antenna theory has been required to model the quantum effects that affect the propagation of EM waves in graphene.

- Using a 1 um x 10 nm graphene-based nano-antenna we can radiate in the Terahertz Band (0.1 – 10 THz)

- Which coincides with the expected operating frequency of graphene devices.

The Terahertz Band (0.1-10 THz) is strongly affected by molecular absorption from different types of molecules (specially water vapor).

For communications over a few tens of meters, this limits the potential of the band to a single transmission window at 300 GHz.

For the expected distances in nanonetworks (below 1 meter), the Terahertz Band offers huge bandwidths, almost 10 THz.

TS-OOK (Time Spread On/Off Keying Mechanism)

- A new communication scheme based on the asynchronous exchange of femtosecond-long pulses.
- Allows very simple and energy efficient nano-transceiver architectures.
- Femtosecond-long pulses are already being used for nanoscale sensing and imaging.
- It provides almost orthogonal channels for different users.

Fig. 3 – TS-OOK modulation scheme. Not in scale

Promising Terahertz sources can be classified into:

- RF NEMS: Oscillation beyond 1 Terahertz will be possible [5]. This technology leads to full graphene circuits.
- STNO: Future low-voltage, room temperature Terahertz Oscillators [6].

In any case, the oscillation frequency of these sources depend on the energy supplied.

*The Energy constraints will provide bad Terahertz Sources*

---


Our Work

- The timing and energy constraints limit the performance of nanonetworks and present a challenge to guarantee the communication among nanodevices.
  - **Timing:** There are frequency drifts among nanodevices
  - **Energy:** A nanodevice can send just a few hundred of bits every minute

- We provide the bridge between the antenna and the nanodevice which consists of three main contributions:
  - A transceiver architecture designed to improve the Symbol Error Rate in the Terahertz channel for pulse-based modulations, which simplifies synchronization schemes built on top.
  - A symbol time estimation built on top of the transceiver architecture to guarantee the successful reception of the symbols.
  - An asynchronous synchronization scheme to detect new transmissions based on a Wake-Up receiver module.
**Table of Contents**

- Introduction
- *Transceiver Architecture for EM Nanonetworks*
- Symbol Time Estimation
- Wake-Up Receiver
- Conclusions and Open Issues
Goal:

- We present a very simple transceiver architecture that:
  - Supports pulse-based modulations in the Terahertz band.
  - Simplifies future synchronization designed on top.

Properties:

- Simple architecture → Suited for nanodevices.
- Outperforms previous architectures in terms of pulse detection capabilities.
- Simplifies the symbol time estimation designed on top of this architecture.

Fig. 5 – Transceiver block diagram architecture
**Transmitter Architecture for EM Nanonetworks**

![Diagram of Transmitter Block Diagram Architecture]

- **Transmitter**
  - **Encoder:**
    - Buffer or memory
    - Codification schemes
  - **Pulse Generator:**
    - Converts the logical values into voltage
  - **Bitrate:**
    - Decides when the next symbol is sent
  - **Output Amplifier**
    - Matches antenna
    - Provides enough power

---

Fig. 6 – Transmitter block diagram architecture
Fig. 7 – Receiver block diagram architecture

**Receiver**
- Terahertz Front-End
  - Dual to Output amplifier
- Power Detection
  - Calculates the input power
- Low pass filter
  - It approximates an ideal integrator

**Peak detector**
- It fixes its output value to “1” when its input is above the threshold. Continuous comparison.

**Decoder**
- Decodes the received packet

**Synch**
- Switches On and Off the receiver
Ideal Non-Coherent Receiver

Fig. 9 – Architecture of an ideal non-coherent receiver

Main Challenges:
- The receiver should operate at 10 THz
- Time-spread modulations, the pulse time is 1000 times shorter than.
- Estimating the time of arrival with an error of some femtoseconds is very challenging

Solution:
- The expected time of arrival can be larger than the pulse time

We propose to use a the maximum function instead of the addition:

But:
- Do we have to implement \( N \) integrators?
- What if the pulse is received in the middle of two of this intervals?

Fig. 10 – Example of the noise effect in typical symbol detectors
- As soon as the integration time is increased, the noise is averaged with the signal.
- This effect drops the performance of the receiver.

Fig. 11 – Example of the noise effect in the proposed symbol detector
- Better Signal to Noise ratio
- But:
  - \( \frac{S}{N} \approx \frac{0.9}{0.25} \approx 3.6 \)
Receiver Architecture for EM Nanonetworks with Continuous-time integration

If we use \( N \to \infty \) Integrators, we convert the system into a linear system with input-to-output relationship:

\[
x(t) = \int_{t-T_p}^{t} u(\tau)^2 \, d\tau
\]

We seek for the maximum of this function over a time \( T \)

\[
\hat{s}[n] = \begin{cases} 
1 & \text{if } \max_{t\in(0,T)} x(t) > V_{th} \\ 
0 & \text{otherwise}
\end{cases}
\]

However, since there is no ideal continuous-time integrator we propose the use of a second order low-pass filter.

Fig. 12 – Comparison between the integrator (left) and second order low-pass filter (right) impulse responses (arbitrary units)

Fig. 13 – Receiver architecture block diagram
Detection of logical “0”

We discretize $x(t)$ into $N$ independent random variables $X_i$ with probability density function:

$$f_n(y) = \frac{1}{2^{v/2} \Gamma\left(\frac{v}{2}\right)} y^{(v-2)/2} e^{-y/2}, \quad y \geq 0$$

where $Y = 2X / N_0$

Chi-square distribution

Thus, the probability density function of $\max X = \max \{X_1, \cdots, X_N\}$:

$$f_{\max,n}(y, N) = NF_n(y)^{N-1} f_n(y)$$
Detection of logical “1”

We discretize $x(t)$ into:

- $N_n$ random variables of noise with probability density function:
  \[ f_n(y) = \frac{1}{2^{v/2} \Gamma(v/2)} y^{(v-2)/2} e^{-y/2}, \quad y \geq 0, \quad f_{\max,n}(y, N) = NF_n(y)^{N-1} f_n(y) \]

- $N_s$ random variables of signal with probability density function:
  \[ f_s(y) = \frac{1}{2} \left( \frac{y}{\lambda} \right)^{(v-2)/4} e^{-\left(y+\lambda\right)/2} I_{(v-2)/2}(\sqrt{y\lambda}), \quad y \geq 0 \]

where:
- $Y = 2X/N_0$
- $\lambda = 2E / N_0$

Thus, the probability density function of $\max X = \max \{X_1, \cdots, X_N\}$:

\[ f_{\max,sn}(y, N_s, N_n) = F_{\max,s}(y, N_s) f_{\max,n}(y, N_n) + f_{\max,s}(y, N_s) F_{\max,n}(y, N_n) \]

Where:

\[ f_{\max,s}(y, N) = NF_s(y)^{N-1} f_s(y) \]
Model Validation

Assumptions:

- Path loss and noise from [12]. These values are expressed in terms of the distance
- TS-OOK modulation scheme. Almost orthogonal channels, so we do not consider collisions
- The transmitter encodes logical “1” with second derivative 1 pJ femtosecond-long gaussian pulse
- The receiver is perfectly synchronized

We validate the expressions for “1”s and “0”s in the Terahertz channel for a distance of 66mm.

<table>
<thead>
<tr>
<th>$T$</th>
<th>$N$</th>
<th>$N_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3\ T_p$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$30\ T_p$</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>$300\ T_p$</td>
<td>110</td>
<td>2</td>
</tr>
</tbody>
</table>

Table. 1 – Relation between the time interval and number of random variables to model the symbol detection

Fig. 15 – Model Validation. Numerical over simulation results
Symbol Error Rate Estimation

- We compare the SER estimation of our symbol detector to the SER estimated in a usual receiver architecture.

The SER has a log-log dependence with the width of the time interval:

\[ n = \frac{T}{T_p} \]

Fig. 16 – Comparison between the SER provided by the proposed receiver and current receiver in terms of the distance for different time intervals

Fig. 17 – Comparison between the SER provided by the proposed receiver and current receiver in terms of the time interval width for a distance of 66 mm
Symbol Error Rate Estimation

We propose the following model

\[ SER_n = n^{0.45} \times SER_{n=1}, \quad SER(r) = r^{0.45} \times SER_{n=1} \]

Then, we obtain the value in origin (\(n = 1\)) using the model of ideal symbol detectors in [9].

Maximum Bitrate

- The use of second-order low-pass filters instead of ideal integrators adds InterSymbol Interference (ISI)
- This ISI affects the receiver only if pulses are not spread in time

Fig. 18 – Comparison between the SER provided by the proposed receiver and current receiver in terms of the distance for different time intervals

Fig. 19 – SER in terms of bitrate

Table of Contents

- Introduction to Nanonetworks
- Transceiver Architecture for EM Nanonetworks
- Symbol Time Estimation Scheme
- Wake-Up Receiver
- Conclusions and Open Issues
## Symbol Time Synchronization Scheme

### Goal:
- We propose a simple frequency estimation scheme that:
  - Is built on top on the transceiver architecture
  - Guarantees the successful reception of the packets
  - Is evaluated in terms of Packet Error Rate estimation

### Properties:
- It uses special properties from the receiver architecture
- Low overhead. This symbol time estimation needs less than 10 pulses to synchronize
- Simple algorithm

---

Fig. 20 – Context of the symbol time synchronization block
Symbol Time Synchronization Scheme

Motivation:

RF NEMS and STNO are expected to provide Terahertz oscillation in the nanoscale but they are energy dependent[5],[6],[13].

Thus, we expect the operating frequency of different nanodevices to be different.

PLL synchronization is discouraged in carrierless pulse based communications[14].

The transceiver architecture proposed provides very interesting synchronization options.

References:


Symbol Time Synchronization Scheme

Frequency Synchronization properties of the receiver:

- Usual receivers:
  - Graphs showing frequency synchronization

- Our Receiver:
  - Graphs showing frequency synchronization
Symbol Time Synchronization Scheme

Frequency Synchronization properties of the receiver:

- Usual receivers:

- Our Receiver
Frequency Synchronization properties of the receiver:

- Slotting a time interval into $K$ sub-intervals, the relation between the error probabilities for logical “0”s and “1”s are:

  \[ P_{e|s=0} = 1 - (1 - p_0)^K \approx Kp_0 \]

  \[ P_{e|s=1} = p_1 (1 - p_0)^{K-1} \approx p_1 \]

- We successfully receive a logical “0” if every subinterval is decoded as “0”

- We receive an error if in the reception of a logical “1” if there is an error in the “1” and the rest time intervals are kept as “0”

Fig. 21 – Property of subinterval slotting
**Symbol Time Synchronization Scheme**

**Frequency Estimation**

- To estimate the frequency we count number of periods between pulses:

  ![](image1)

- As shown in the example: Receiver 1 detects a $T_s$ of 5 sampling periods
- Receiver 2 detects a $T_s$ of 4 sampling periods

- We refer this as Relative Frequencies

- To improve the performance in the estimation, we have information *a priori* about the received frequency

  ![](image2)

**Expected time, $K$ subintervals**

- The receiver is in Standby
- **Fig. 22 – Standby - Reading time**
**Symbol Time Synchronization Scheme**

**Frequency Estimation:**
- There is an error in this estimation. The receiver can count only an integer numbers of periods.
- We propose the use of a synchronization preamble:

\[ N_0 = 11, \; N_i = \{9, 9, 10, 9\} \]

Using this estimation, there is always an error that the receiver must be able to handle.

\[ \hat{N} = \frac{\sum_{i=1}^{N_{synch}} N_i}{N_{synch}} = N + \epsilon \]

\[ |\epsilon| < \frac{1}{N_{synch} - 1} \]

Fig. 22 – Example of the frequency estimation process
Adaptive Frequency Correction

During the transmission, the receiver must be able to cope handle the estimation errors and possible frequency drifts

“1”s Provide synchronization information

“0”s Provide uncertainty

Fig. 22 – Example of the adaptive frequency correction algorithm
Optimum Number of Subintervals

- It must guarantee that the next pulse is inside the time interval.
- It must be kept as small as possible to reduce the error probability.

\[ K_{i+1} = \left[ (n_{\text{zeros}} + 1)(\hat{N}_s + \epsilon) + 1/2 \right] - \left[ (n_{\text{zeros}} + 1)(\hat{N}_s - \epsilon) - 1/2 \right] \]

The average number of subintervals is:

\[ \overline{K} = \sum_{n=0}^{\infty} p_n E[k_n] = 2 \left( 1 + \frac{\epsilon_{\text{max}}}{P_{s=1}} \right) \]

Where:
- \( p_n \): probability of receiving \( n \) consecutive “0”
- \( E[k_n] \): average number of subintervals when the receiver has received \( n \) consecutive “0”
- \( \epsilon \): maximum error accepted

Then, there are \( 2K-1 \) zero subinterval per each one subinterval, thus we approximate the Packet Error Rate as:

\[ PER = 1 - \left( 1 - P_{e|s=1} / 2 - P_{e|s=0} / 2 \right)^{N_{\text{bits}}} \approx \frac{\overline{K}}{2K-1} \text{SER}(2K-1) \]

\[ P_{e|s=1} \approx \text{SER}(2K-1) \]
Symbol Time Synchronization Scheme

- **Preamble Evaluation**
  - There is a probability that the error is kept inside the maximum error accepted.
  - This maximum error depends on the number of pulses for synchronization.
  - Probability estimated in terms of the channel degradation.

- **Frequency correction evaluation**
  - We have simulated the adaptive algorithm proposed.
  - We observe that unbalancing probabilities we obtain a minimum in the Packet Error Rate estimation.

![Fig. 22 – Probability of no synchronization in terms of the SER and the synchronization preamble length](image1)

![Fig. 23 – Evaluation of the frequency correction. PER in terms of the maximum error and unbalancing parameter](image2)
Symbol Time Synchronization Scheme

**Frequency correction Evaluation**
- We evaluate the expression for the Packet Error Rate in terms of the channel degradation and we compare the results with the simulation results for the algorithm.
- Appropriately unbalancing probabilities.

**Benefits of this frequency correction**
- We compare the Packet error rate with:
  - Ideal synchronization
  - Non frequency correction
- We outperform in one order of magnitude.

![Fig. 24 PER comparison. Numerical model vs. Simulation](image1)

![Fig. 25 PER comparison. Numerical model vs. Simulation](image2)
How many pulses must be sent to synchronize frequencies?

- A few number of pulses increases the PER, increases the time interval but reduces overhead.
- Large number of pulses improves PER, reduces time interval but increases overhead.
- We define:

\[
t_{\text{put}} = \frac{\left( N_{\text{bits}} - N_{\text{synch}} / P_{s=1} \right) \left( 1 - \text{PER}_{\text{synch}} \right) P_{\text{synch}}}{N_{\text{bits}} \left( 1 - \text{PER}_{I} \right)}
\]

- Less than 10 pulses are needed to synchronize if the adaptive algorithm is being used.
- Alternatively, without the algorithm some tens of pulses are needed.
Table of Contents

- Introduction
- Transceiver Architecture for EM Nanonetworks
- Symbol Time Estimation Scheme
- **Wake-Up Receiver for EM Nanonetworks**
- Conclusions and Open Issues
Goal:

- We provide an asynchronous synchronization scheme to detect new transmissions that:
  - is based on a wake-up receiver
  - We evaluate its functionality over the ALOHA protocol

Properties:

- Asynchronous synchronization
- It is capable of rejecting packets before the receiver wakes up if the receiver is not the target of this packet

Fig. 27 – Context of the Wake-Up module
Motivation:

- Due to power restrictions, a receiver node can only decode some tens of packets of 200 bits each minute.
- The rest of the time, the receiver must be sleeping to save energy.
- It is too expensive (in energy) for the receiver to decode any packet not targeted to it.
- Due to clock drifts, duty cycled synchronization schemes do not apply.
Duty cycled synchronization schemes

- Nodes wake up periodically to sense the channel, in case any node is transmitting.
- When a node is transmitting, it sends a synchronization preamble. If the receiver decodes the packet, the receiver switches to reception and the transmitter sends the packet.
- Suitable for carrier communications.
- Power Consumption proportional to:

\[ P = \frac{T_1}{T_1 + T_2} \]

EM Nanonetworks

- TS-OOK: Carrierless
- **We consider frequency drifts**
- Some tens of nanosecond long packets per minute.
- The energy constraints limit the duty cycle to be very reduced.
- Maximum drifts of nanoseconds allowed.

![Example of duty cycled synchronization schemes](image)

Fig. 28 – Example of duty cycled synchronization schemes

Wake-Up Receiver

We need an asynchronous scheme to synchronize the nanodevices.

A wake-up receiver needs to constantly sense the channel but using less power [16].

The wake-up signal must be easier to decode.

In particular, authors in [16] they use a second frequency to synchronize.


Fig. 29 – Comparison between duty cycled and wake-up synchronization schemes
Wake-Up Signal
- The medium is shared with other users.
- The pulses are spread in time.
- The receiver cannot try to synchronize every pulse it detects.
- The Wake-Up signal cannot be a preamble of pulses.

We propose the use of pulse bursts.
Detection of a Pulse Burst

- We model this pulse burst as $N_B$ independent pulses.
  - This detection can be done with power detectors, detecting a minimum power during a minimum time.
- To provide robustness, we suppose that not all of the pulses are needed to detect a burst.

\[
P_D = \sum_{i=0}^{N_B-N_b} \binom{N_B}{N_b+1}(1-p_d)^{N_b+i}p_d^{N_B-N_b-i}
\]

- Additionally, it is also valid for when a neighboring node starts a transmission.

Effect of noise and Interference

- We model noise and interference as Poisson arrival.
  \[
  \lambda = \lambda_n + \lambda_i
  \]
  \[
  \lambda_n = \frac{p_0}{T} \\
  \lambda_i = N\lambda_{TX}
  \]
- We model the behavior of the wake-up module in presence of noise as a M/D/c/c queue.

Fig. 30 – M/D/c/c queue model
Orthogonal Burst Preamble

As the number of neighboring nodes increases, the number of false alarms is increased.

To be energy consistent, the nanodevice has to wake-up only if this is the target of this packet

We propose time orthogonality between two consecutive pulses

Fig. 31 – Example of Orthogonal Burst Preamble
Protocol Description

- We propose to build this synchronization scheme on top of the ALOHA protocol.

A nanodevice sends a packet whenever it needs to send it.

- The receiver acknowledges the packet by using a burst acknowledgment (BACK).

- If the transmitter does not receive the BACK, it sends again the packet.

Fig. 32 – Protocol description. Current states and power consumption

Fig. 29 – Receiver state diagram
**False alarm**

- We refer as a false alarm as starting the reception due to neighboring nodes, interference or noise.

---

**Fig. 33 – False alarm probability in terms of the node density**

- **When the pulse burst is short:**
  - The false alarm is mainly affected by noise.

- **When the pulse burst is large:**
  - The false alarm is mainly affected by interferences and neighboring nodes.

- **When using orthogonal preambles,**
  - the node is not affected.

**Fig. 34 – False alarm probability in terms of the packet size**
**Loss Probability**
- Losing a packet due to the protocol depends on the number of neighboring nodes.
- However, this loss probability is very low. The system is highly scalable.

**Energy Consumption**
- We model the energy consumption in terms of the stateflow.
- The energy to receive a pulse is fixed to 0.1 pJ while the power in wake up is fixed to 0.7 pW.

---

*Fig. 35 – Loss probability in terms of the node density*

*Fig. 36 – Energy consumption in terms of the node density*
Table of Contents

- Introduction
- Transceiver Architecture for EM Nanonetworks
- Symbol Time Estimation
- Wake-Up Receiver
- Conclusions and Open Issues
Conclusions and Open Issues

Conclusions:

- We provide a bridge between the antenna and the future network protocols. For this:
  - We propose a low complexity transceiver architecture, which provides better performance in terms of Symbol Error Rate and simplifies the frequency synchronization designed on top.
  - We propose a low complexity frequency synchronization scheme to guarantee the successful packet delivering. This is evaluated in terms of Packet Error Rate.
  - We propose an asynchronous synchronization scheme based on a wake-up receiver for nanodevices to enable the communication among nanodevices.
Open Issues:

- Simulation and implementation of the transceiver architecture over a specific technology.

- Integration of the transceiver architecture results and frequency estimation in a network simulator.

- Network protocols designed built on top of our Wake-Up transceiver architecture.
Thank you very much for your attention!