
Inga

Inga


Norwegian Sea
Iceland
Sweden
Finland
Norway
Denmark
United Kingdom
Ireland
Belarus
Ukraine
Austria
France
Romania
Italy
Greece
Turkey
Portugal
Syria

Inga

http://invinets.com/

invinets

THERMOMESH REQ. QUOTE COMPANY CONTACT EN PL

THERMOMESH IS A NETWORK MEASURING TEMPERATURE AND HUMIDITY

IT VISUALIZES HOW THESE PARAMETERS EVOLVE IN SPACE AND TIME
SUCH PRECISE INFORMATION GIVES YOU MEASURABLE BENEFITS

VIEW TECHNICAL SPECIFICATIONS

REQUEST QUOTE

WE ARE HAPPY TO CALL THE FOLLOWING ENTITIES OUR CUSTOMERS

onet.pl
UNIWERSYTET WARSZAWSKI
PAN POLSKA AKADEMIA NAUK
TRAMWAJE WARSZAWSKIE
JMP flowers
INSTYTUT SPORTU
ZKM Gdańsk
ZAMEK KRÓLEWSKI NA WAWELU
EUROSER
ANIMEX
Nord Capital
mza
DAMCO
VPPlant
Raben


invihets
Made in Poland
4251C


invihets
Made in Poland
4251C


invihets
Made in Poland
4251C


invihets
Made in Poland
4251C

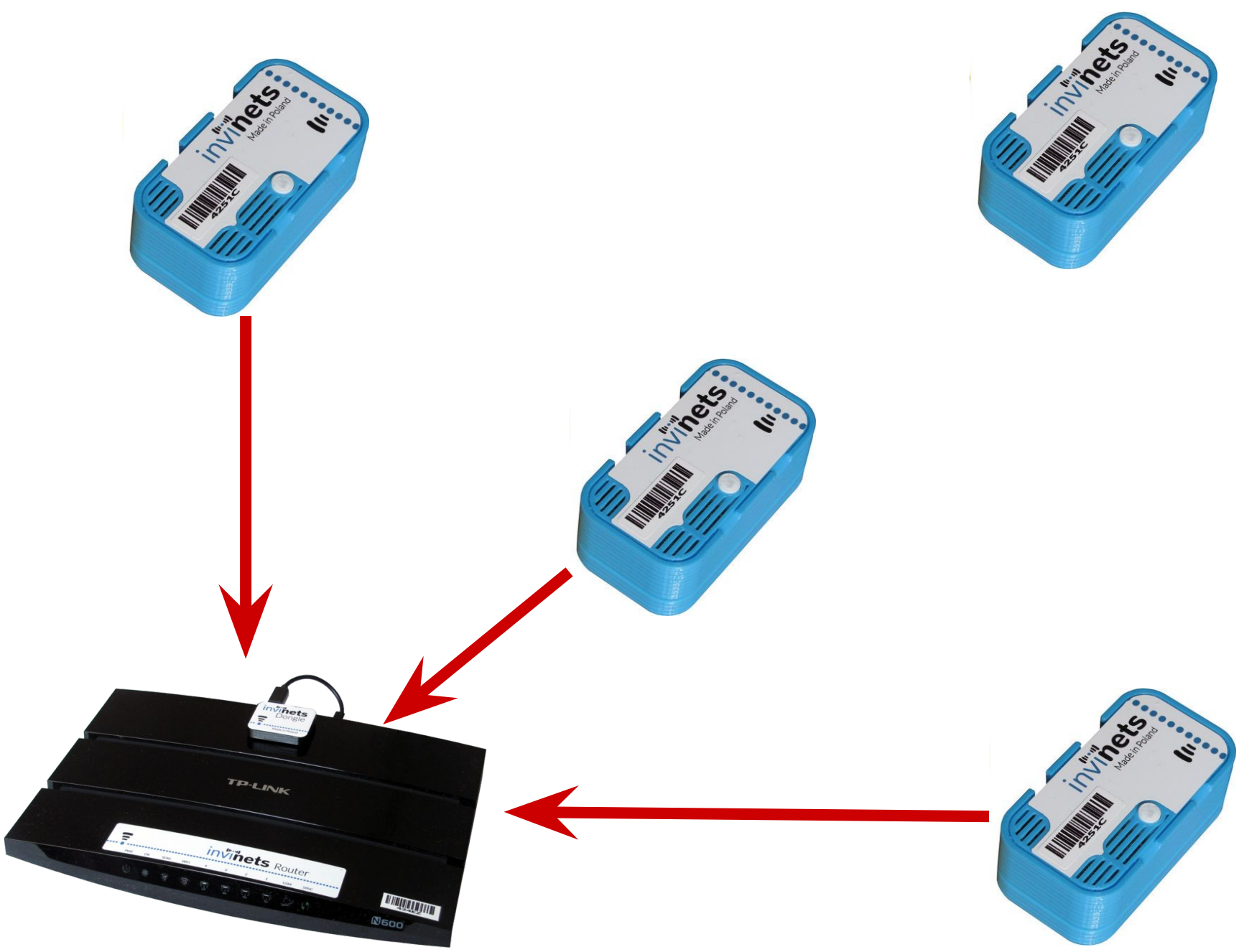

invinets
Made in Poland
4251C

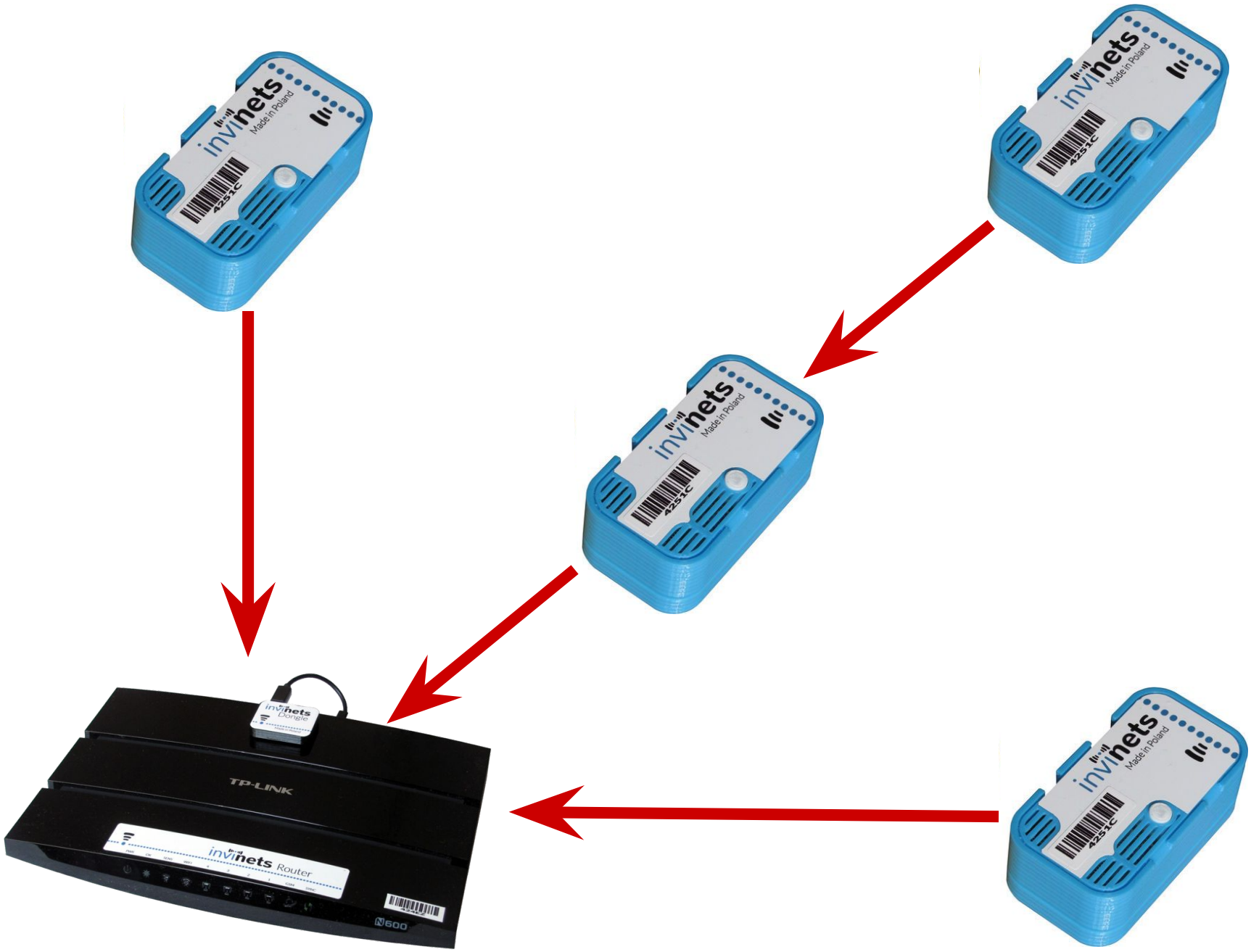

invinets
Made in Poland
4251C


invinets
Made in Poland
4251C


invinets Dongle
TP-LINK
invinets Router
N600


invinets
Made in Poland
4251C


invihets
Made in Poland
4251C
invihets Dongle
TP-LINK
invihets Router
N600


invinets
Made in Poland
4251C
TP-LINK
invinets Router
N600

Customers wish:

★ to add or replace some nodes

Customers wish:

★ to add or replace some nodes

The company wishes:

★ to keep customers satisfied
at the possibly lowest cost

Customers wish:

- ★ to add or replace some nodes
- ★ to keep sensitive data protected

The company wishes:

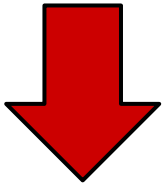
- ★ to keep customers satisfied at the possibly lowest cost

Motes need to be easily configurable by the customers,

without external devices nor additional hardware interfaces.

Motes need to be easily configurable by the customers,

without external devices nor additional hardware interfaces.



crosstalk-based communication = cross-technology communication (**CTC**)

Motes need to be easily configurable by the customers,

without external devices nor additional hardware interfaces.

crosstalk-based communication = cross-technology communication (**CTC**)

IEEE 802.15.4

IEEE 802.11

Ad Hoc 802.11-802.15.4 Crosstalk-Based Communication in Practice

Inga Rüb
Szymon Acedański
Konrad Iwanicki

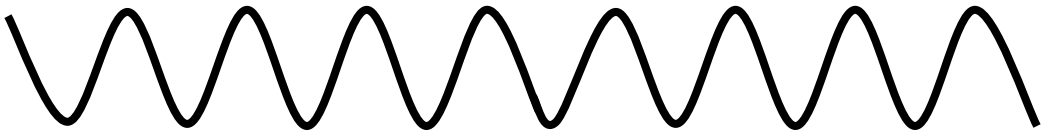
Ad Hoc WiFi-ZigBee Crosstalk-Based Communication in Practice

Inga Rüb
Szymon Acedański
Konrad Iwanicki

The Idea of CTC

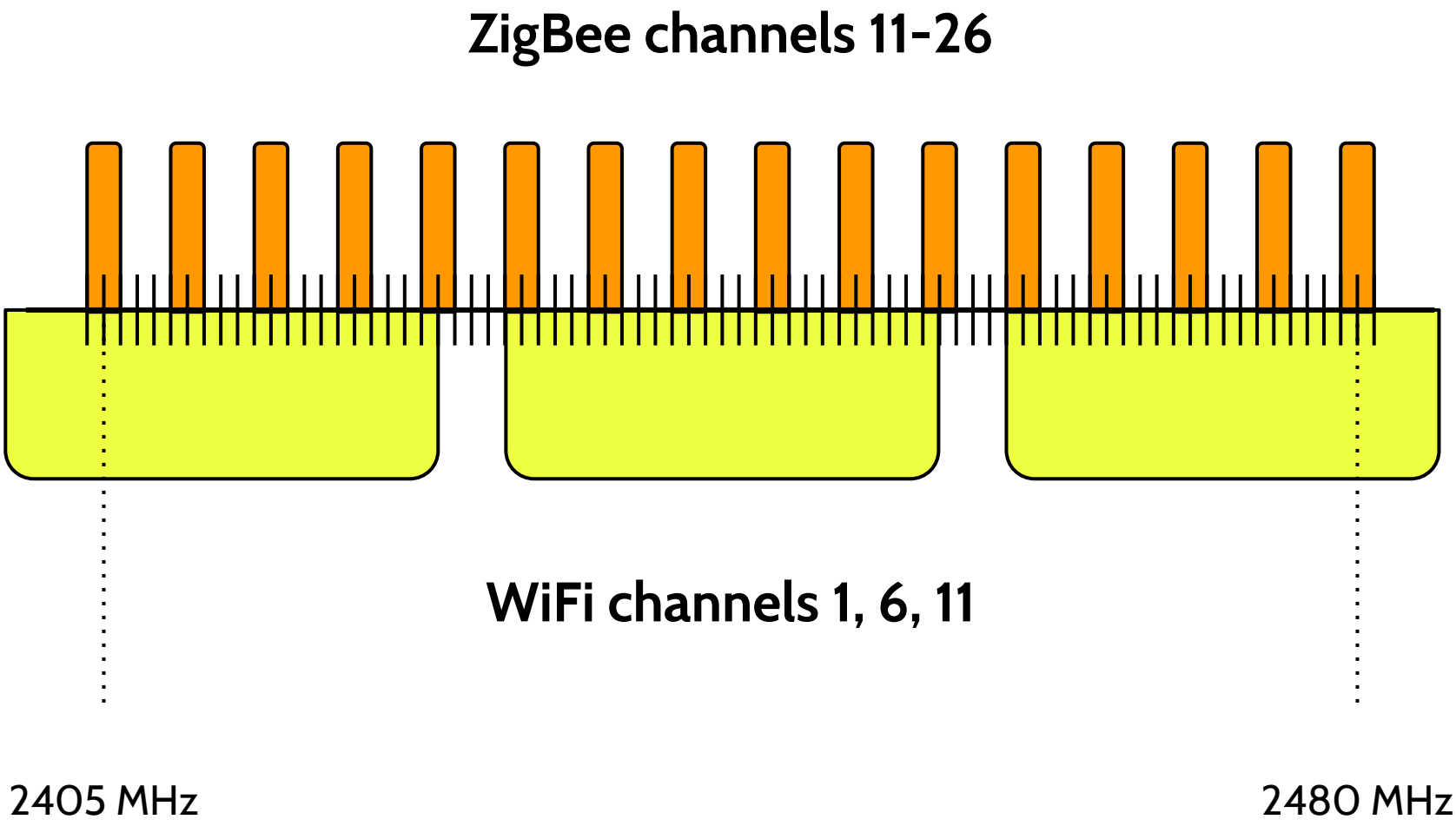
ZigBee radio detects **RSSI** (**Received Signal Strength Indicator**) for a given channel.

The Idea of CTC

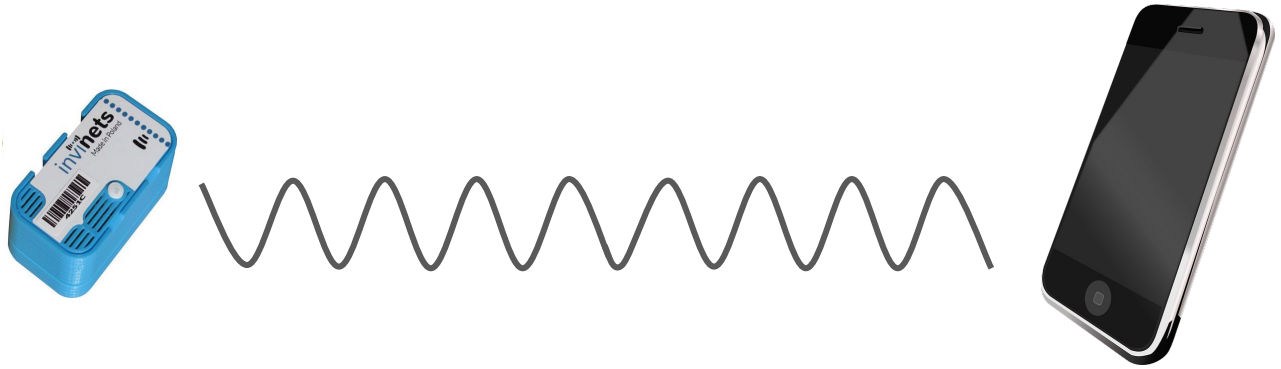


ZigBee radio detects **RSSI** (**Received Signal Strength Indicator**) for a given channel.

The Idea of CTC

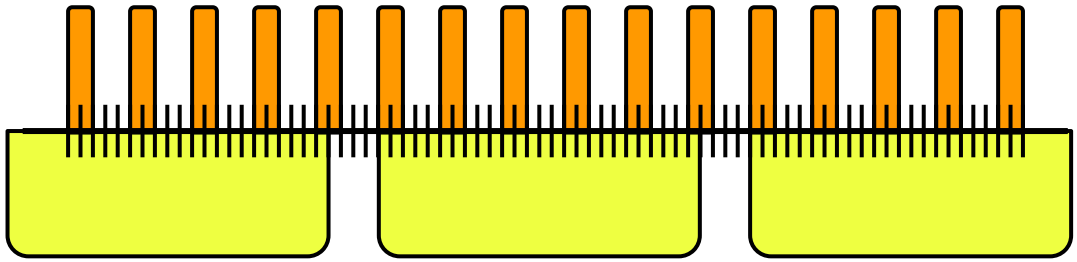


The Idea of CTC



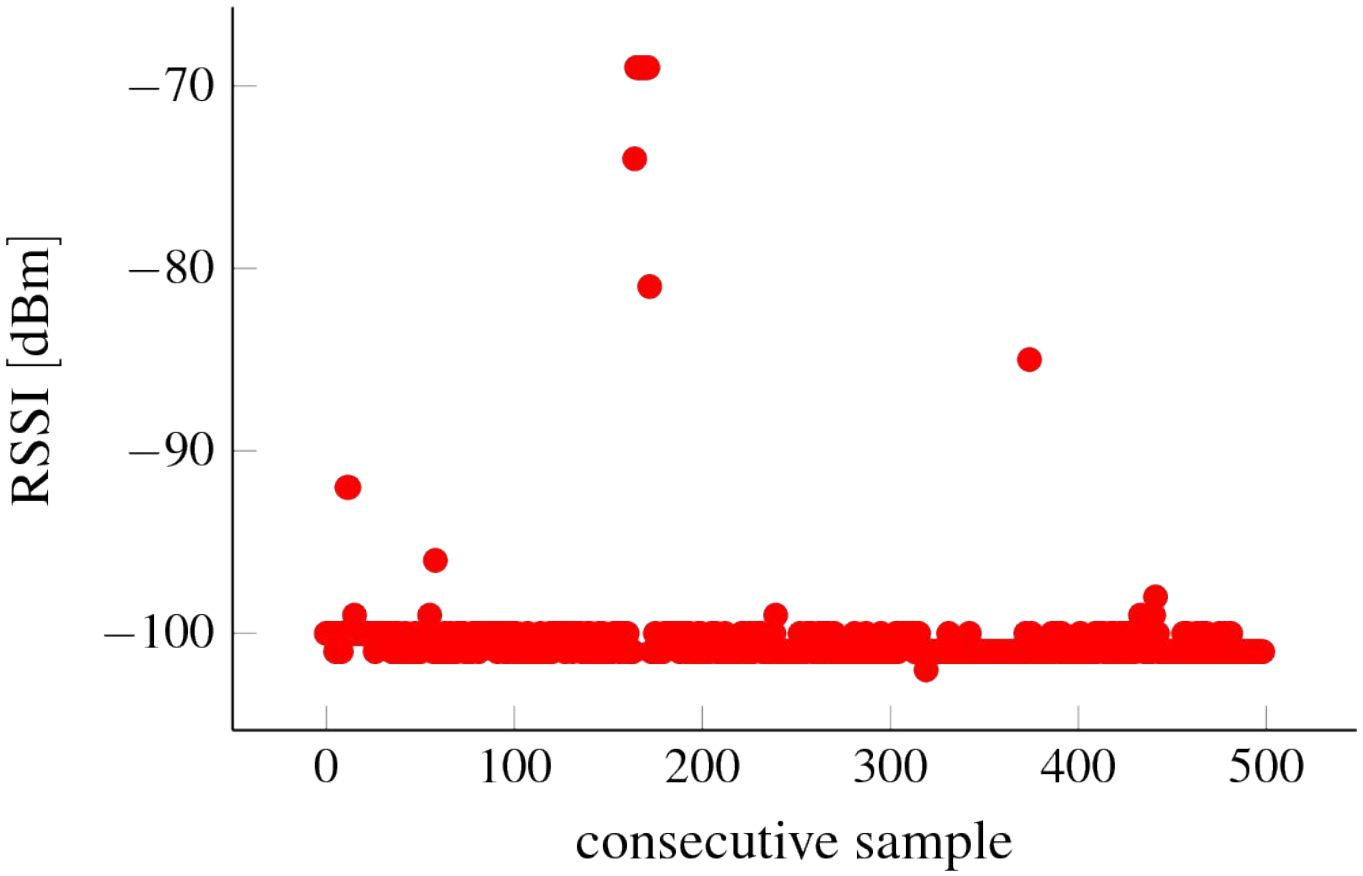
ZigBee radio detects **RSSI** (**Received Signal Strength Indicator**) for a given channel.

ZigBee channels 11-26



WiFi channels 1, 6, 11

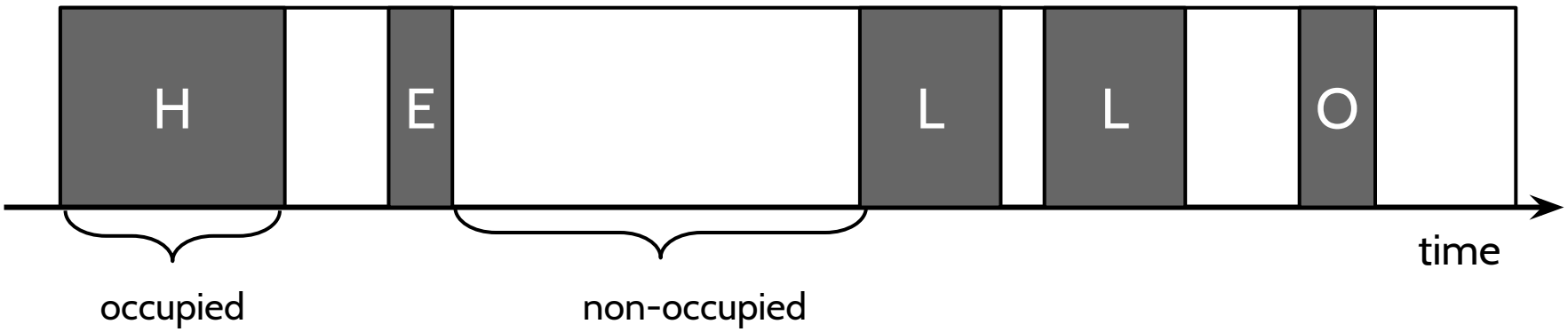
The Idea of CTC



ZigBee radio (CC2650 MCU) detects WiFi activity

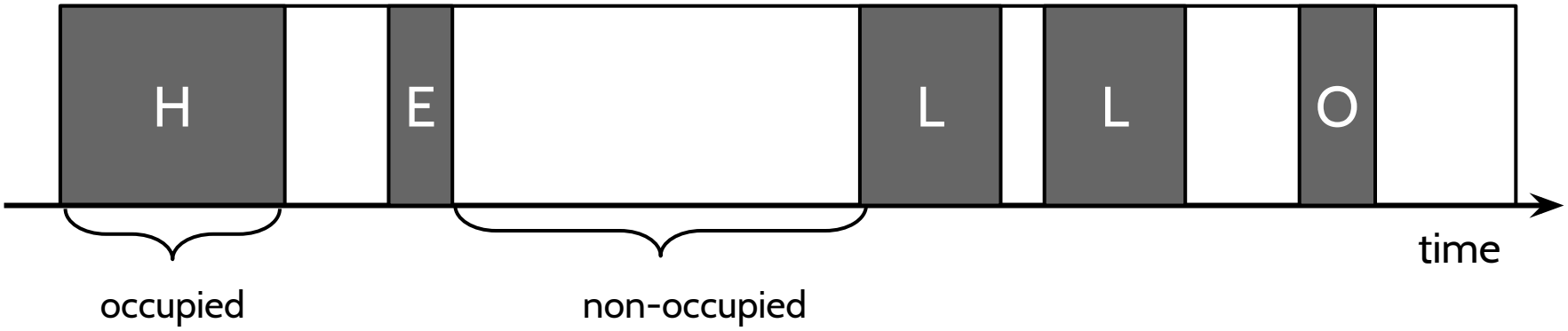
The Idea of CTC

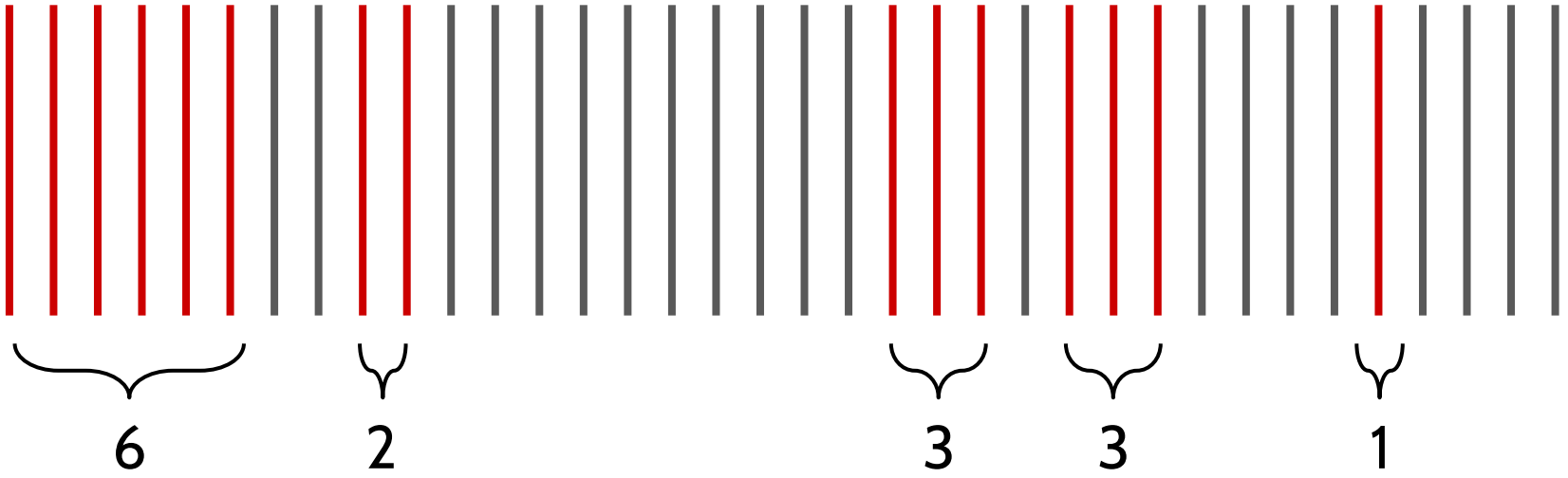
WiFi device occupies the channel **for** a specified interval.



The Idea of CTC

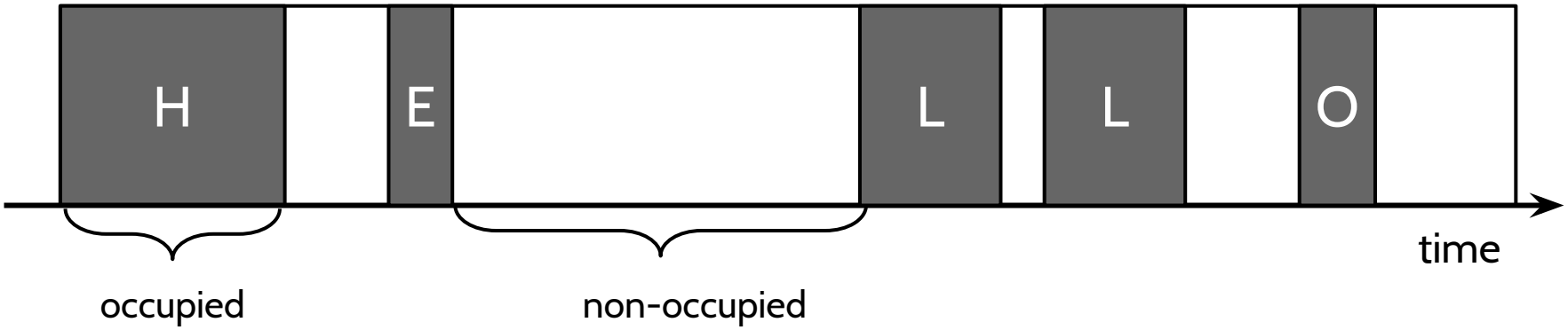
WiFi device occupies the channel **for** a specified interval.



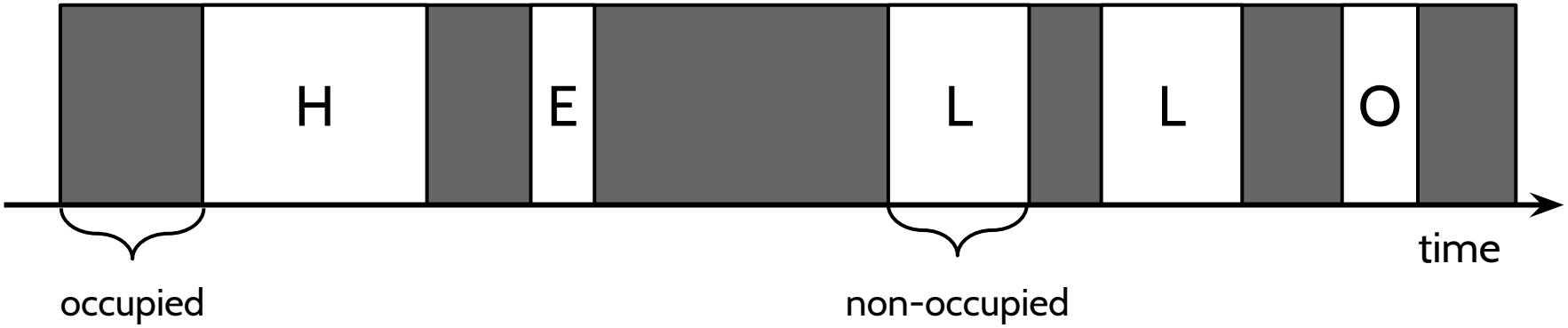


The Idea of CTC

WiFi device occupies the channel **for** a specified interval.



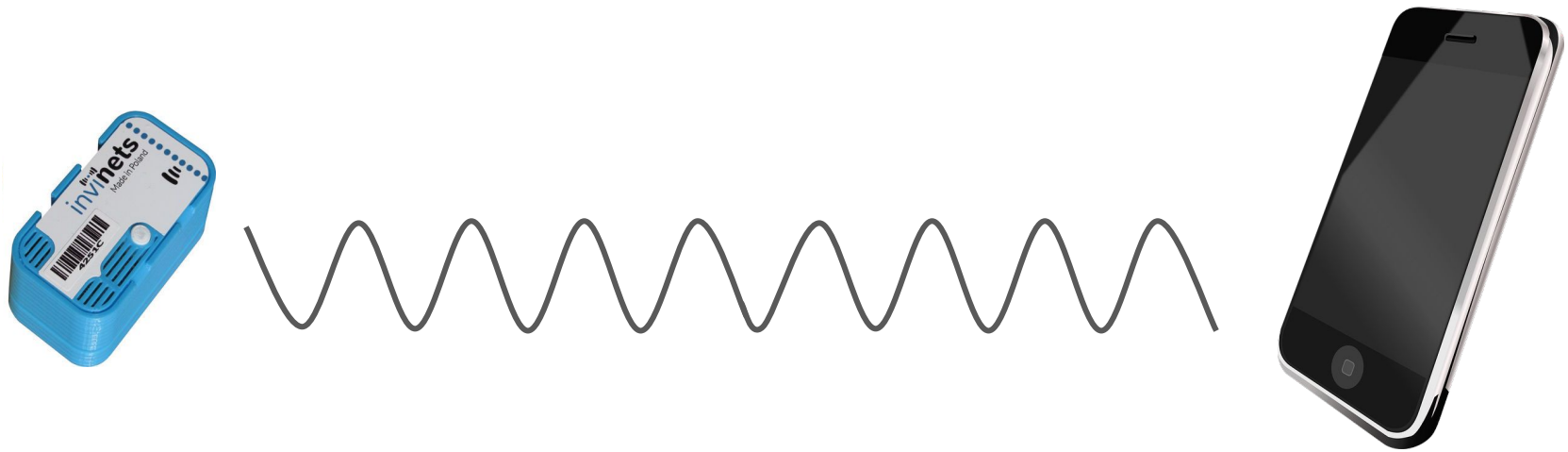
WiFi device occupies the channel **after** a specified interval.



Trade-off in Our Case

One needs to modify the smartphone's firmware to control when the WiFi packet is sent.

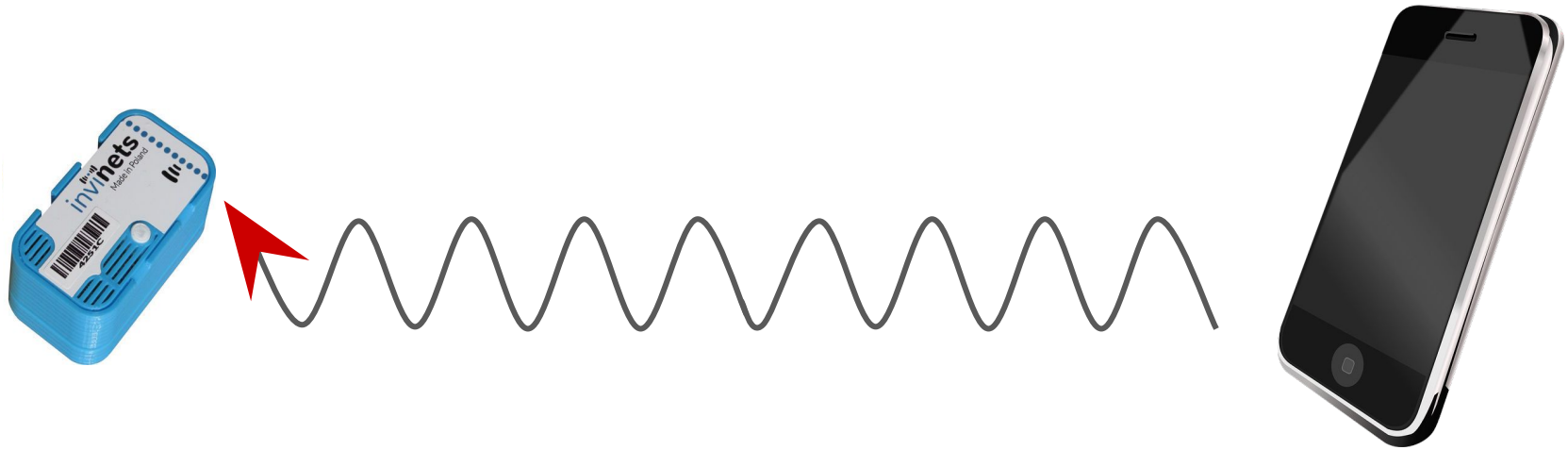
The customers should not be required to root their smartphones.



Trade-off in Our Case

One needs to modify the smartphone's firmware to control when the WiFi packet is sent.

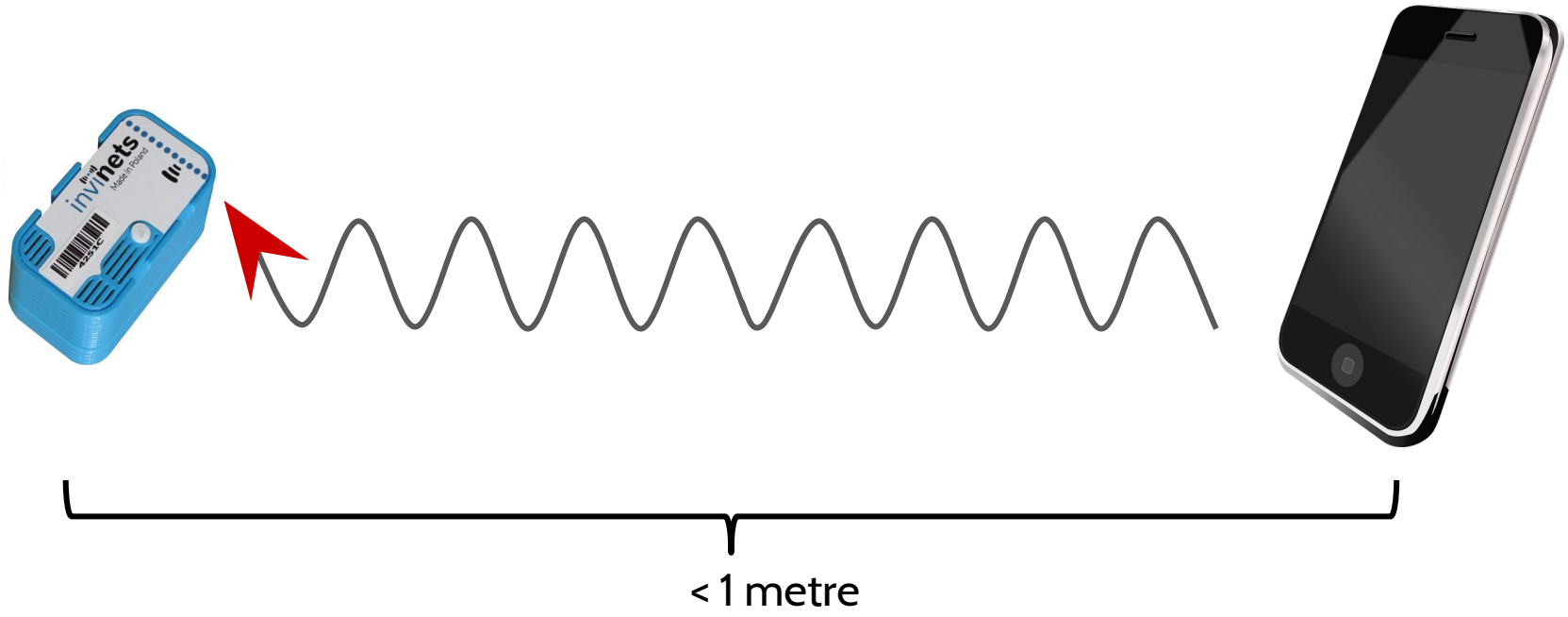
The customers should not be required to root their smartphones.



Trade-off in Our Case

One needs to modify the smartphone's firmware to control when the WiFi packet is sent.

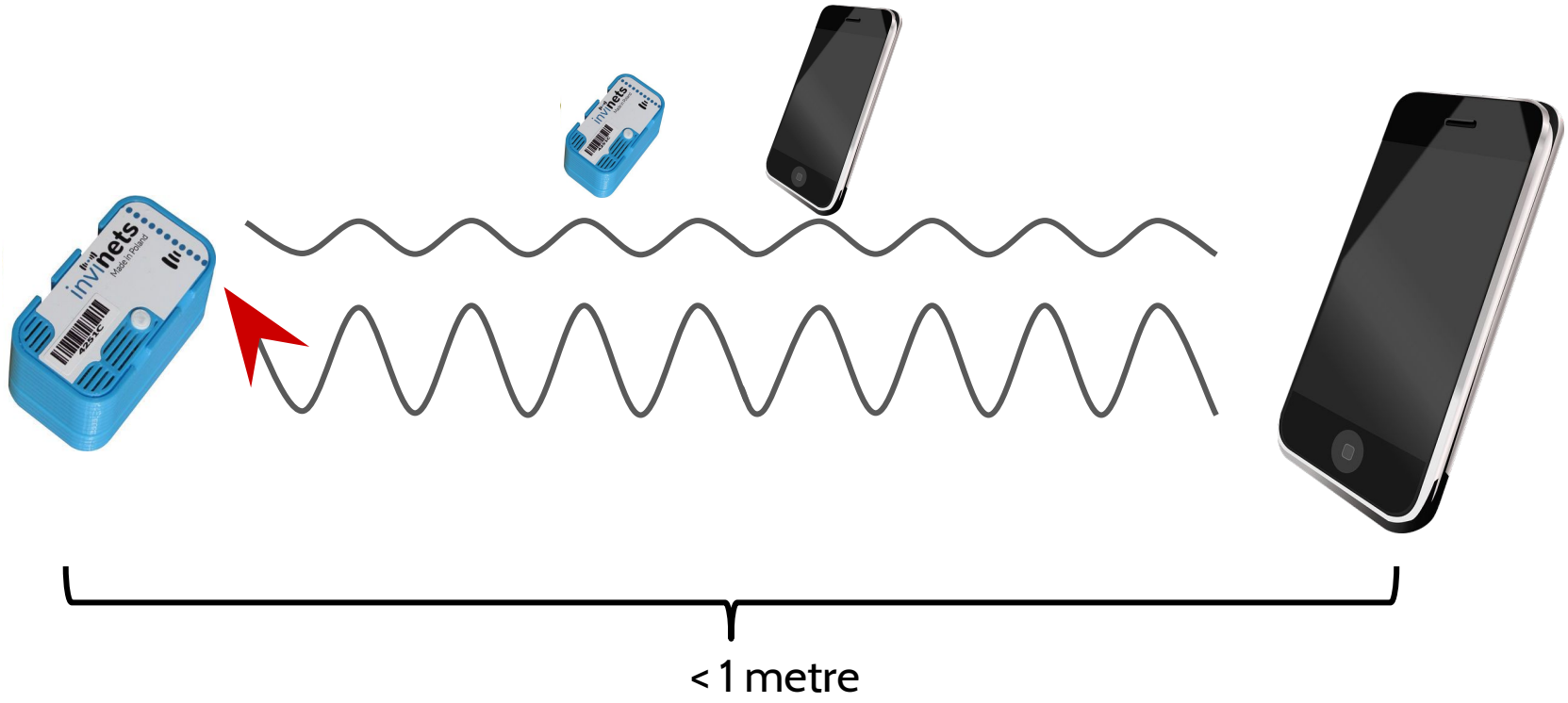
The customers should not be required to root their smartphones.



Trade-off in Our Case

One needs to modify the smartphone's firmware to control when the WiFi packet is sent.

The customers should not be required to root their smartphones.



We Get To Work

Prepare the ZigBee device

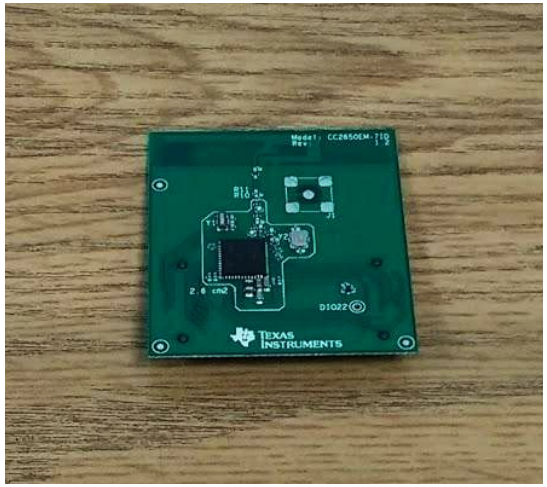
Prepare the WiFi device

Set the Alphabet

Evaluate

The ZigBee Side

CC2650 MCU connected to SmartRF06

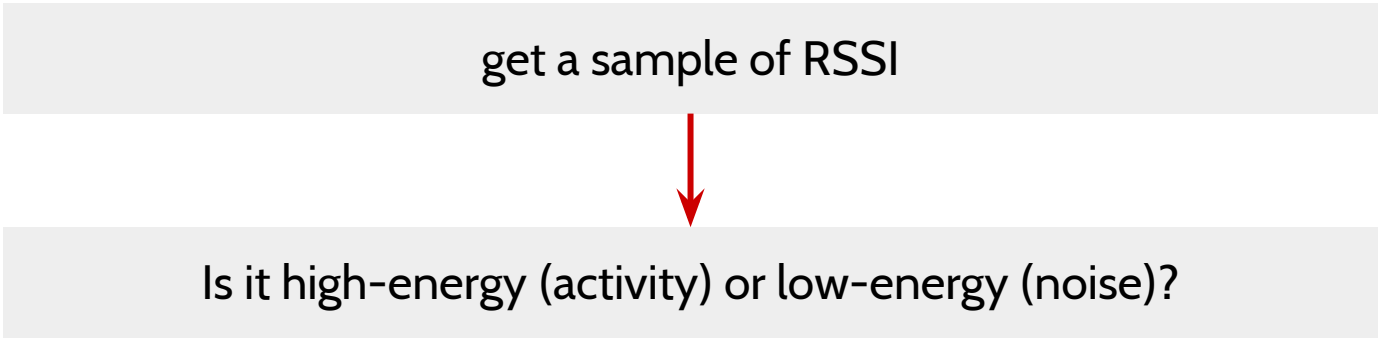


Parameters:

- ★ RSSI sampling interval of 250 µs
- ★ our own TinyOS-based OS
- ★ radio sensitivity of -100 dBm

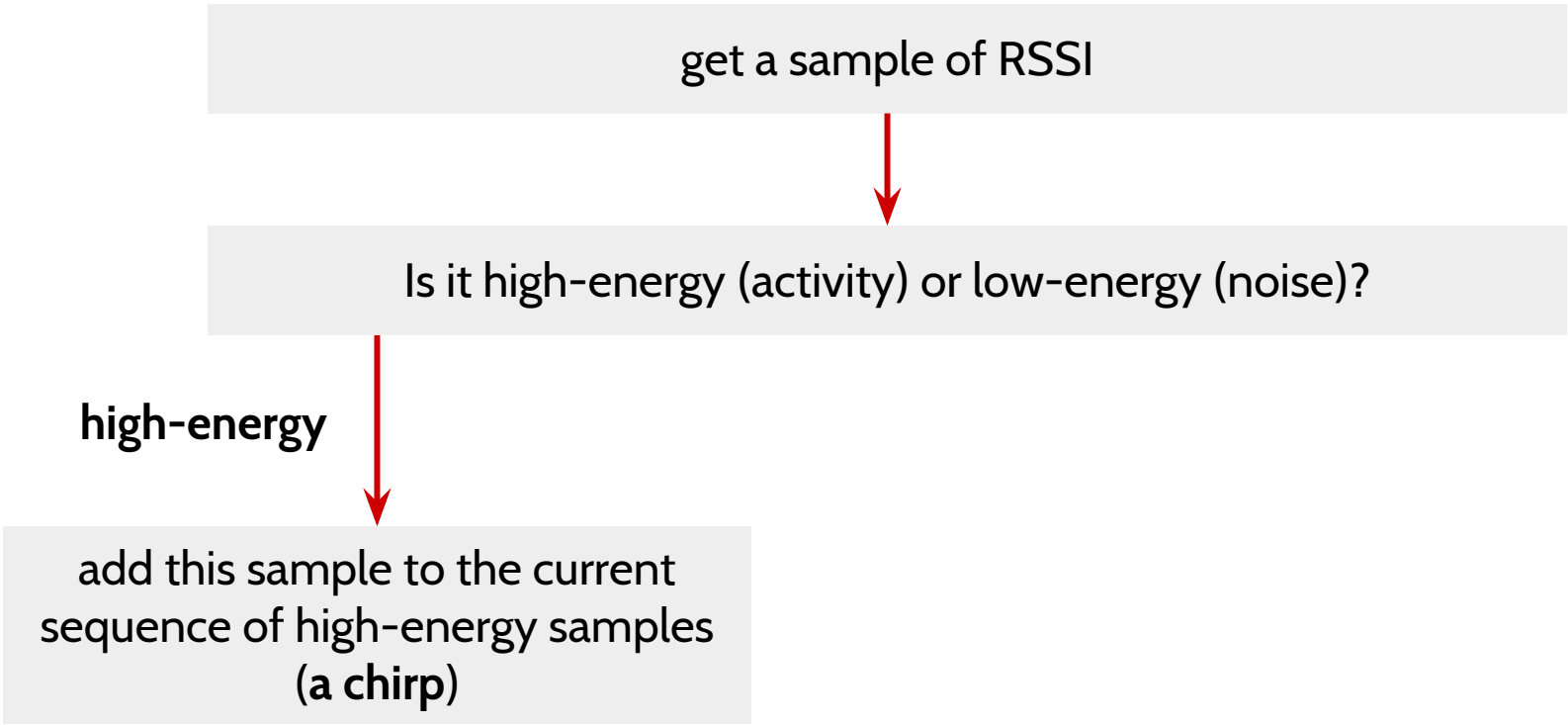
The Algorithm

(based on existing solutions: Esence and HoWiES)



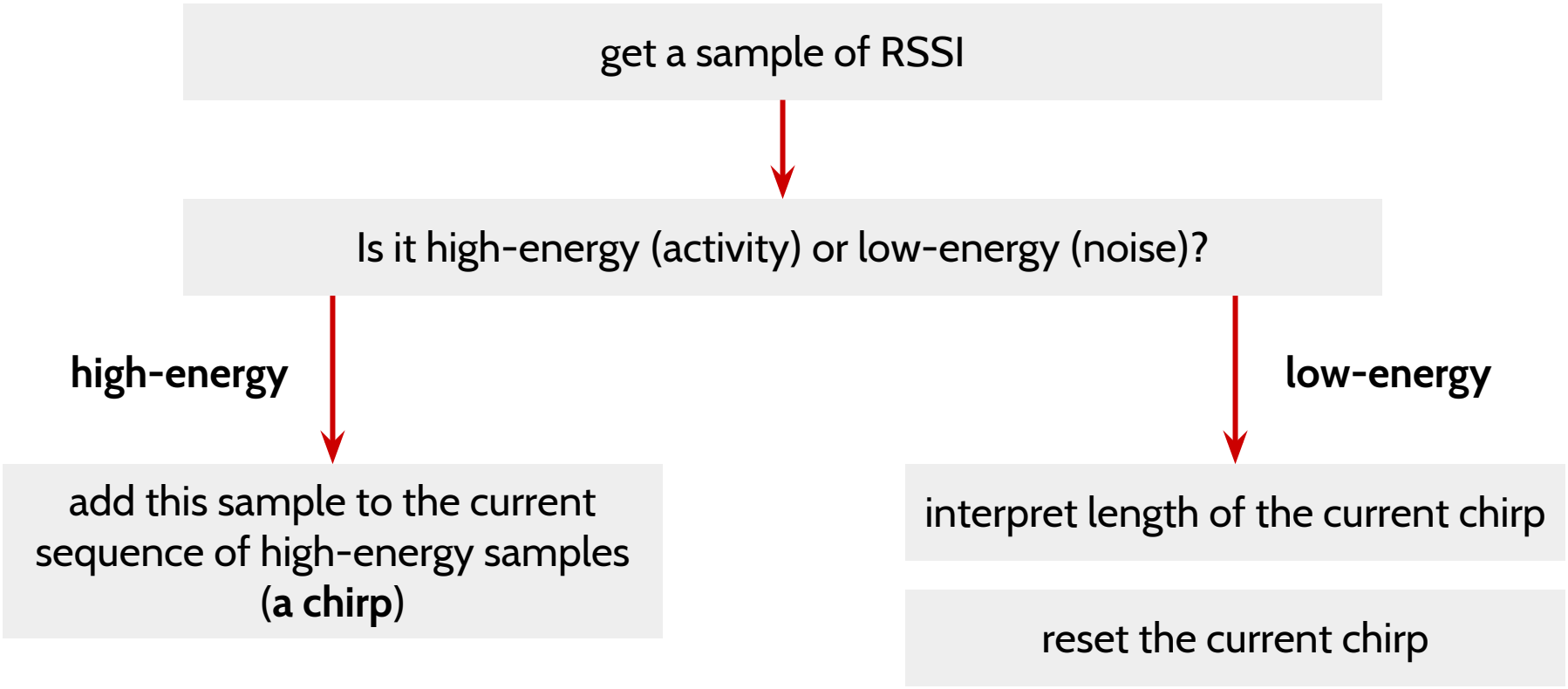
The Algorithm

(based on existing solutions: Esence and HoWiES)



The Algorithm

(based on existing solutions: Esence and HoWiES)

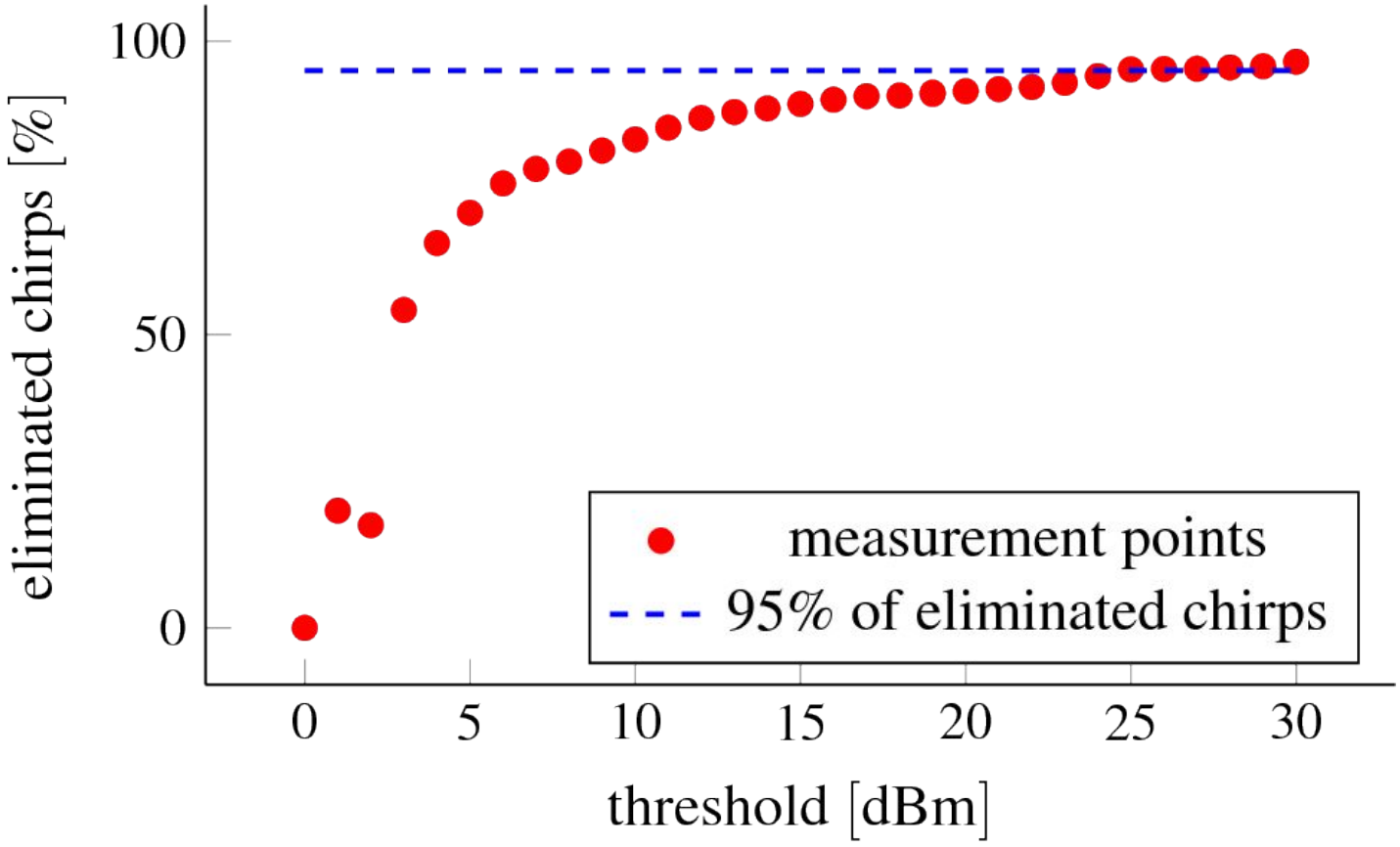


Energy of High-Energy Samples

$$E_{sample} > E_{noise} + E_{threshold}$$

Energy of High-Energy Samples

$$E_{sample} > E_{noise} + E_{threshold}$$



For a moderately noisy environment (a university room with a WiFi AP).

We Get To Work

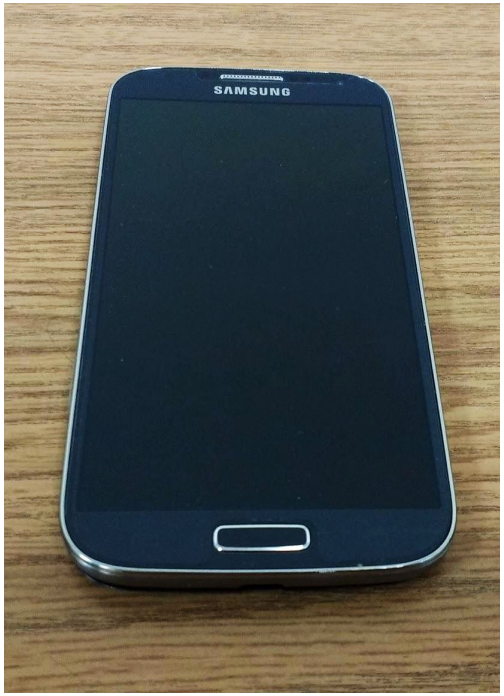
Prepare the ZigBee device

Prepare the WiFi device

Set the Alphabet

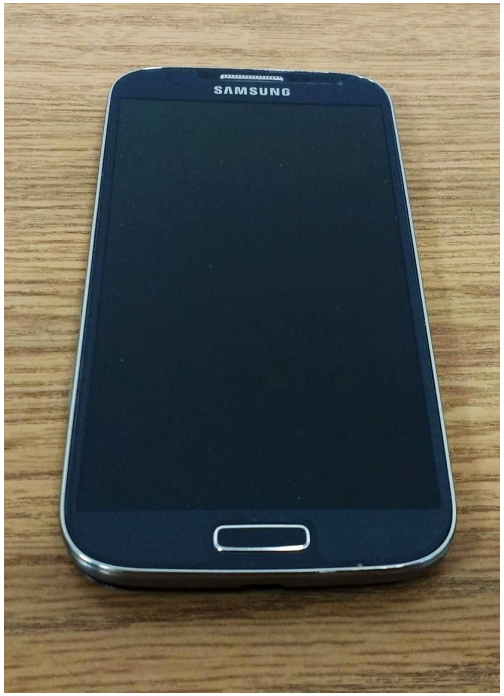
Evaluate

The WiFi Side



Parameters to control:

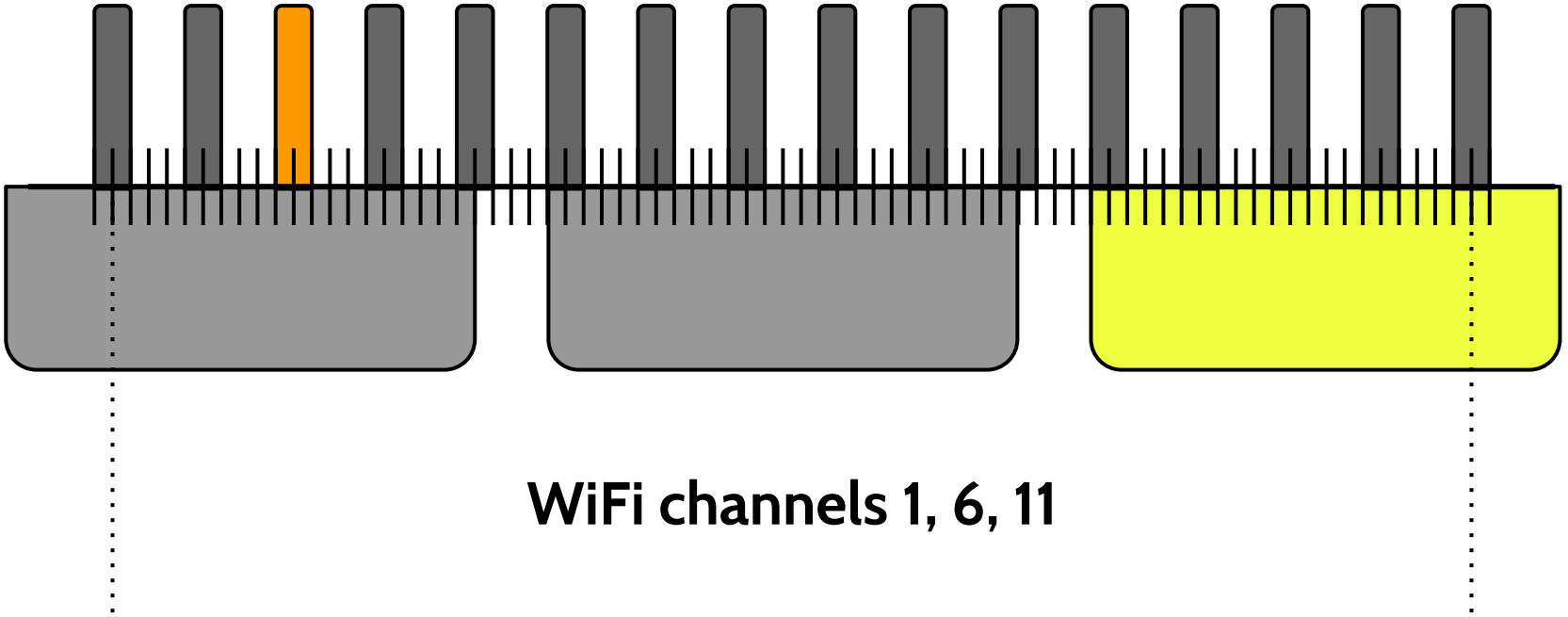
The WiFi Side



Parameters to control:

★ the WiFi channel

ZigBee channels 11-26

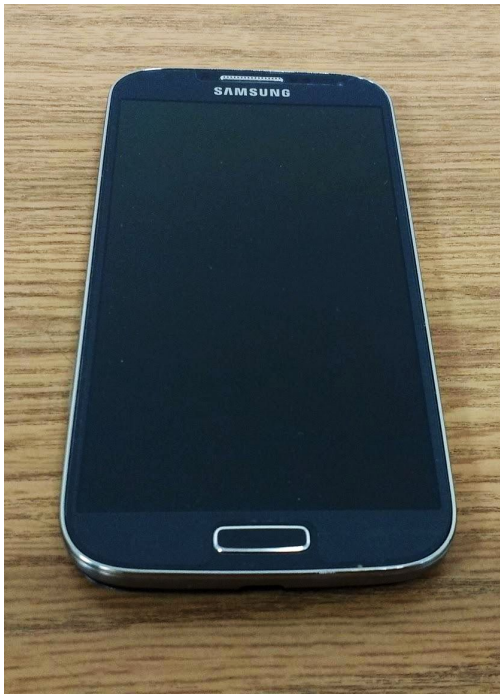


WiFi channels 1, 6, 11

2405 MHz

2480 MHz

The WiFi Side

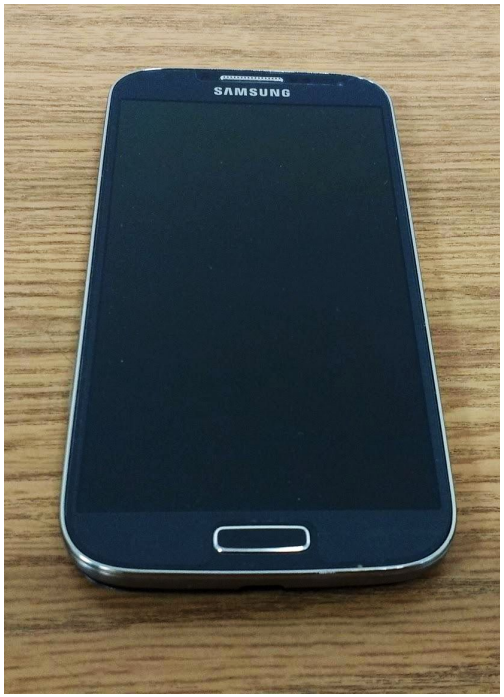


Parameters to control:

★ the WiFi channel

Maybe we can hop over various channels and find the overlapping one?

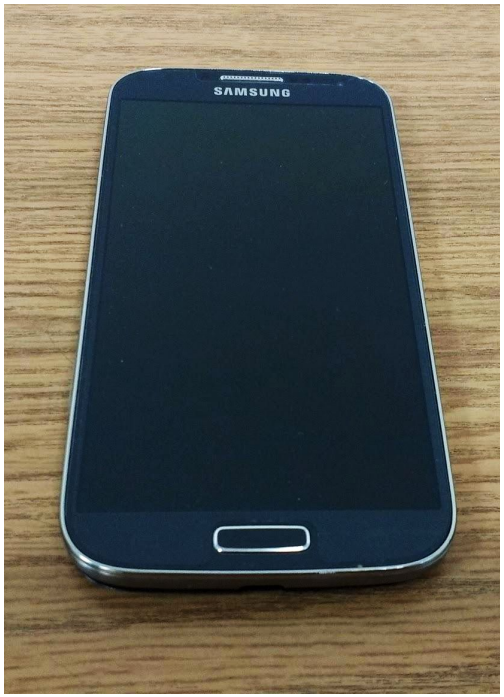
The WiFi Side



Parameters to control:

- ★ the WiFi channel
- ★ the bit rate

The WiFi Side

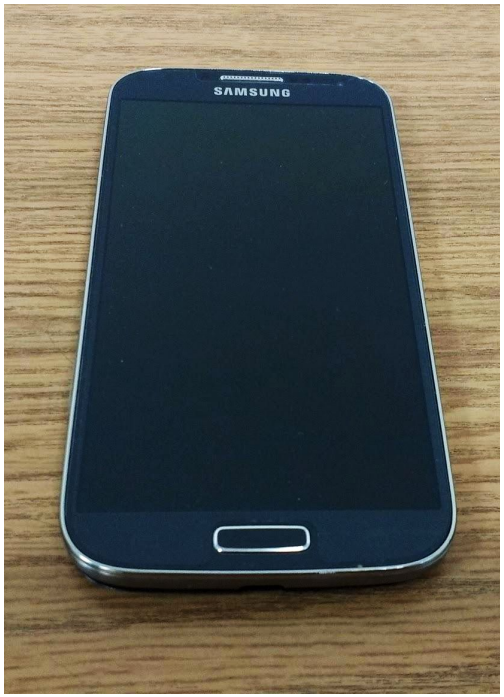


Parameters to control:

- ★ the WiFi channel
- ★ the bit rate

By default, AP sends broadcast messages at 1 Mbps.

The WiFi Side

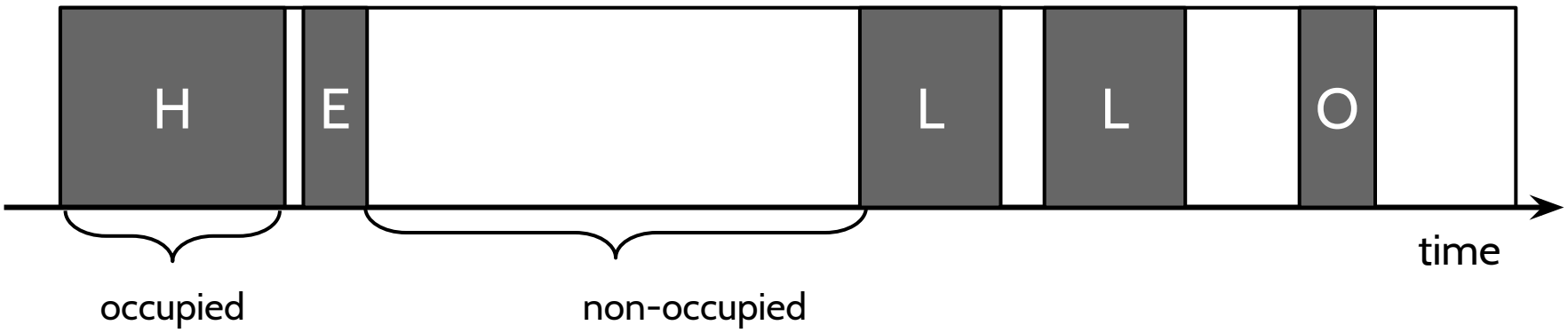


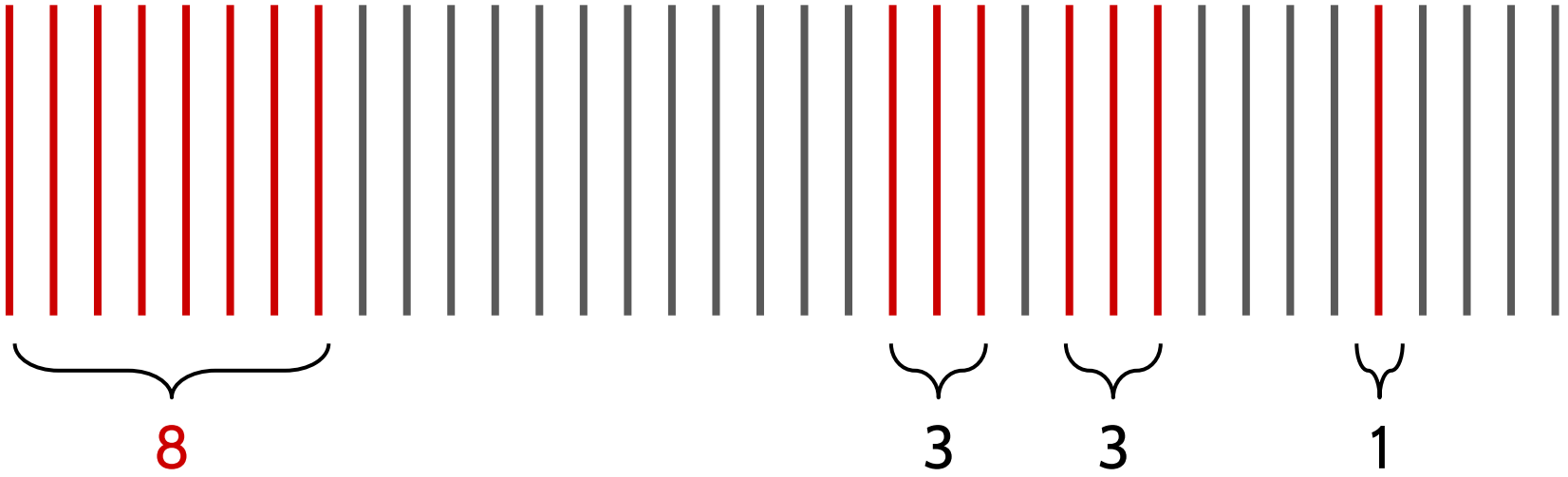
Parameters to control:

- ★ the WiFi channel
- ★ the bit rate
- ★ the interval between chirps

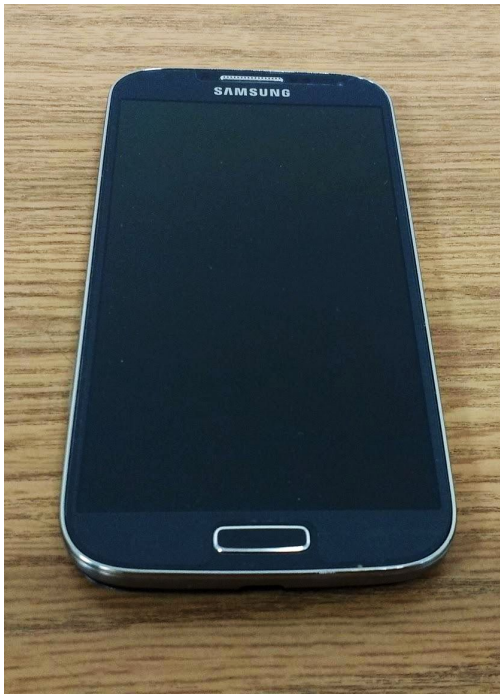
The Idea of CTC

WiFi device occupies the channel **for** a specified interval.





The WiFi Side

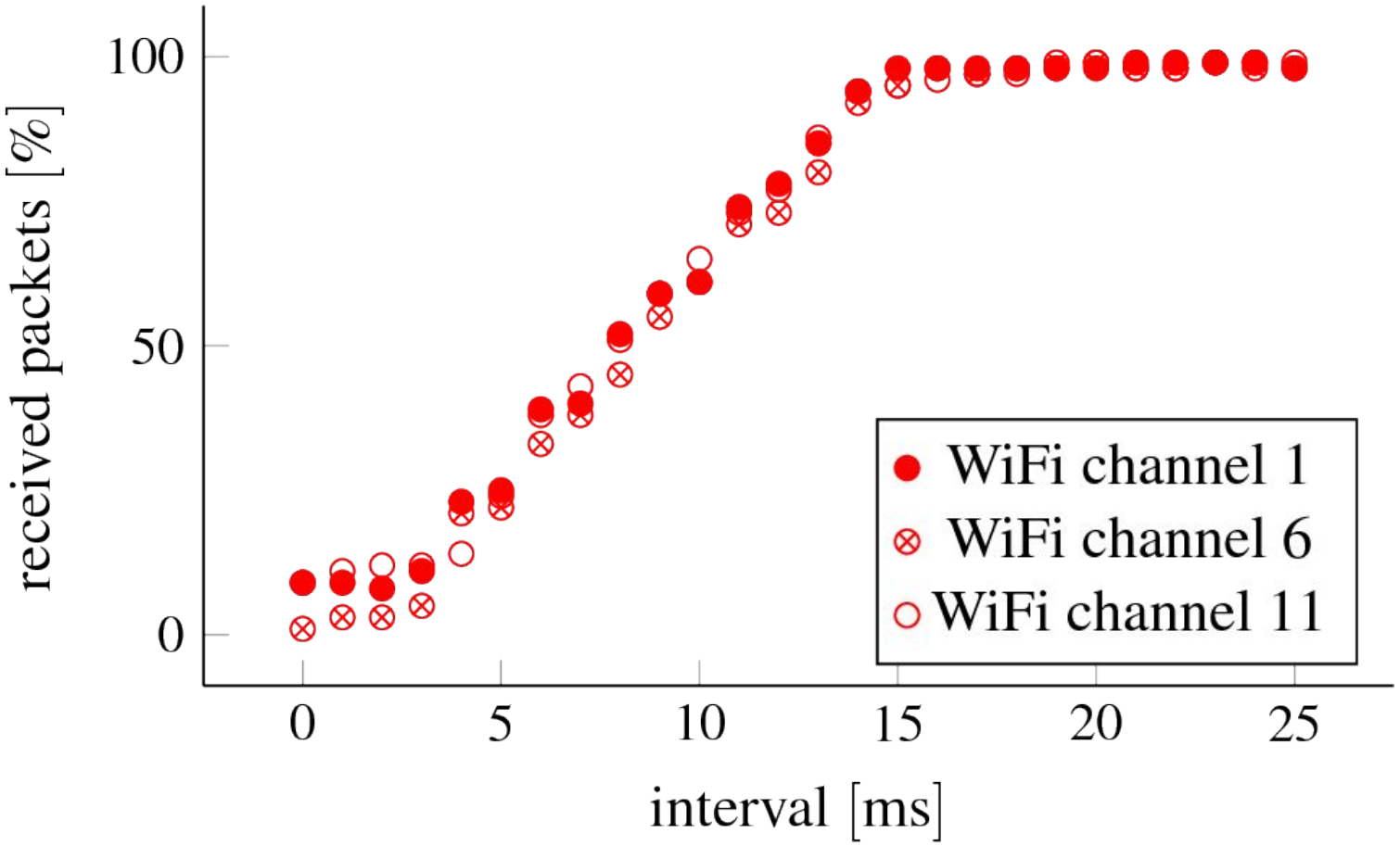


Parameters to control:

- ★ the WiFi channel
- ★ the bit rate
- ★ the interval between chirps

We should test it!

The Interval Between Packages



15 ms is pretty good. We chose 20 ms.

We Get To Work

Prepare the ZigBee device

Prepare the WiFi device

Set the Alphabet

Evaluate

The Alphabet Letter

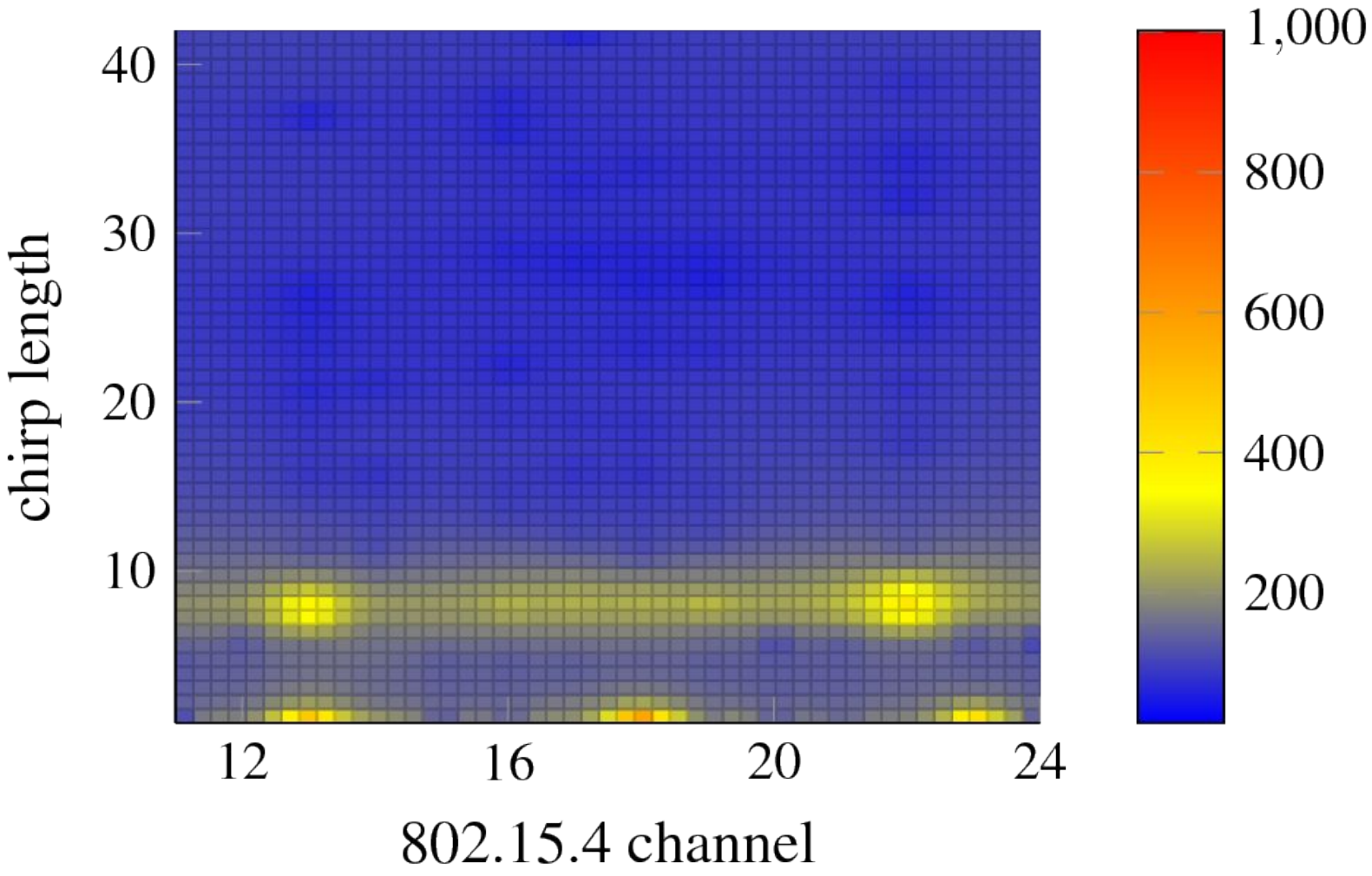
sends a packet of a certain size - a letter.

It takes some time.

The packet is detected as a sequence of *N* chirps by

The alphabet is a set of possible packet size.
They need to be distinguishable from each other.

Common Chirps



There are already many short chirps and 8-sample chirps in the air.

The Alphabet Limits

Packets should generate chirps of more than 8 samples.

Packets cannot exceed 1500 B limit.

The Alphabet Limits

Packets should generate chirps of more than 8 samples.

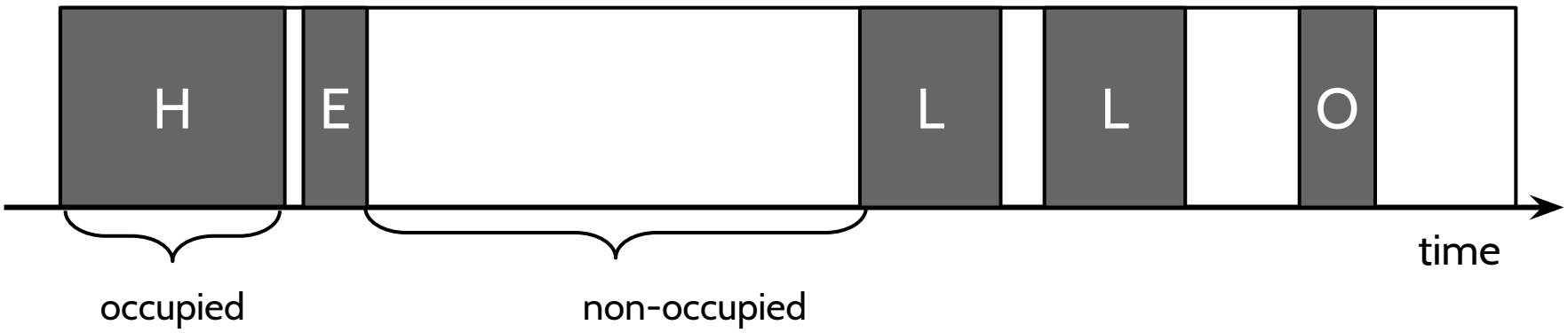
Packets cannot exceed 1500 B limit.

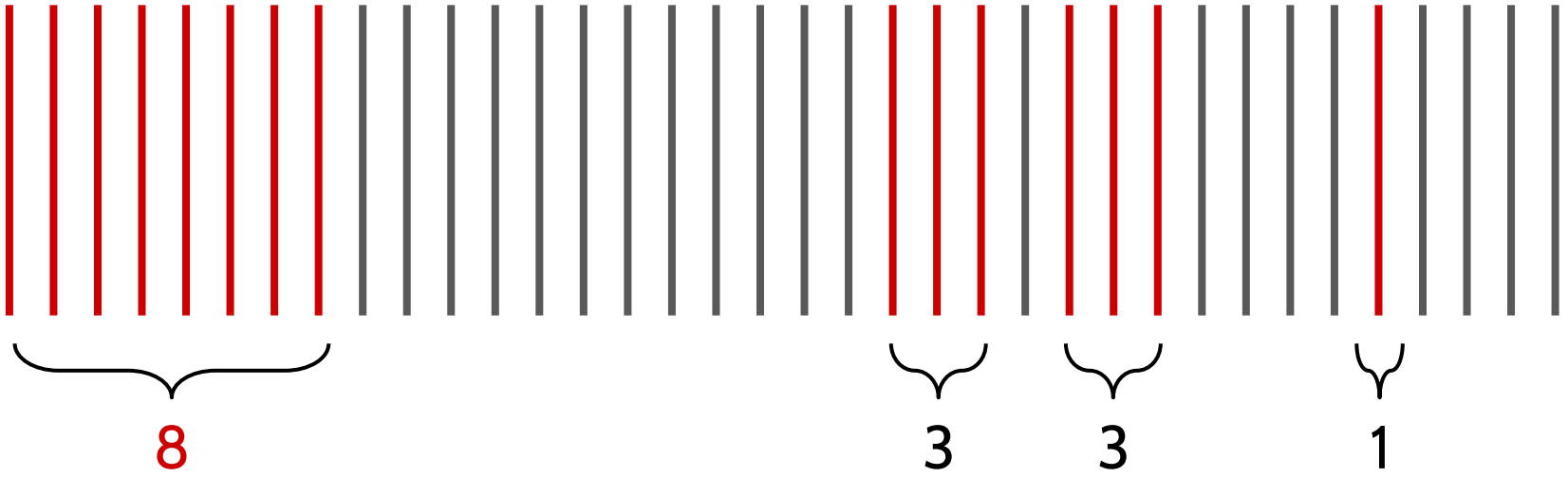
An additional problem:

A packet that should generate
N-sample chirp
is sometimes detected as
(N+7)-sample chirp.

The Idea of CTC

WiFi device occupies the channel **for** a specified interval.





The Alphabet Limits

Packets should generate chirps of more than 8 samples.

Packets cannot exceed 1500 B limit.

An additional problem:

A packet that should generate
N-sample chirp
is sometimes detected as
(N+7)-sample chirp.

Beacons-sticking forces us to eliminate packet lengths
that theoretically generate (N+7) chirps.

We Get To Work

Prepare the ZigBee device

Prepare the WiFi device

Set the Alphabet

Evaluate

First Evaluation

3 different smartphones

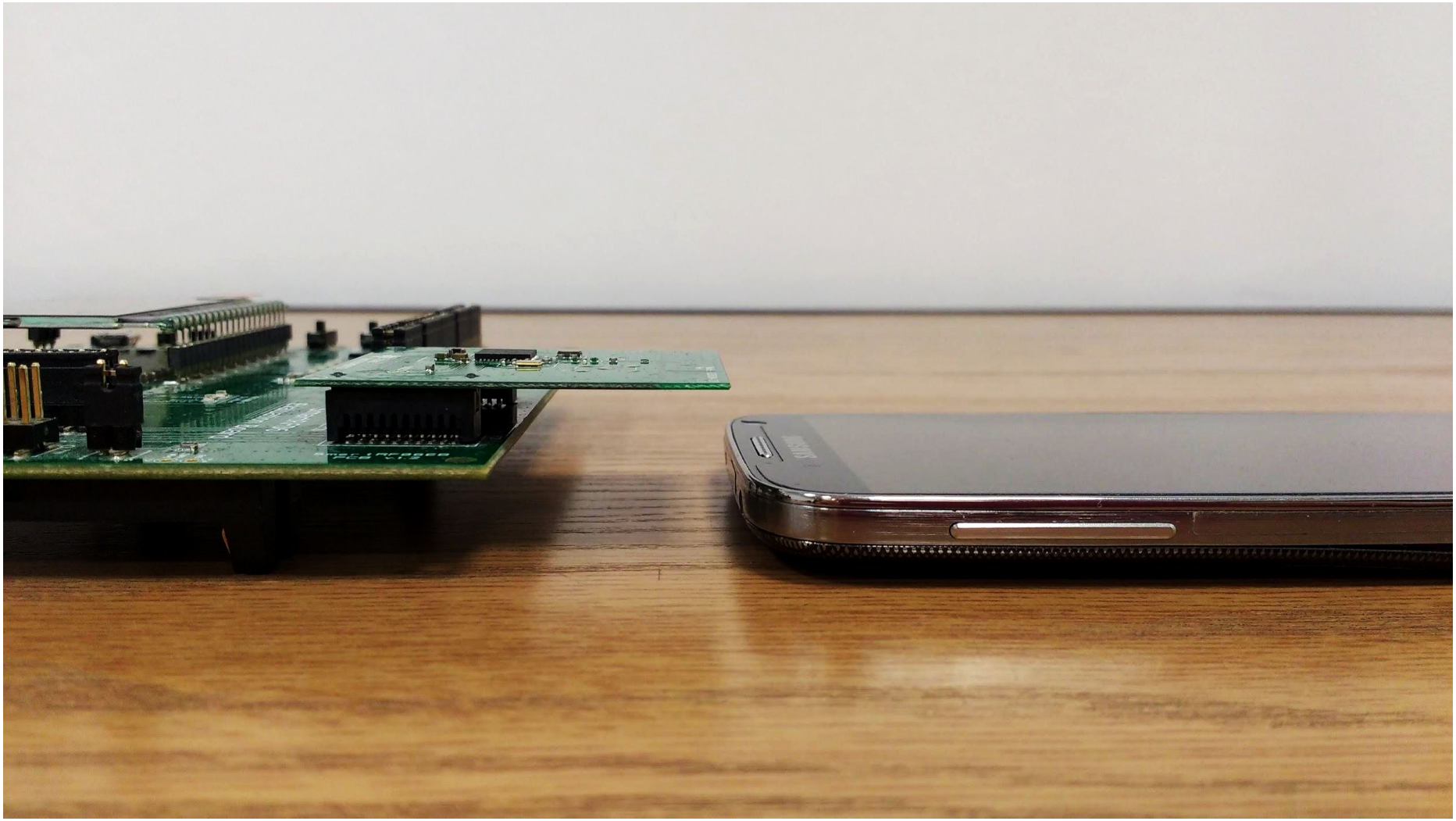
4 different environments (from quiet to moderately noisy)

overlapping channels

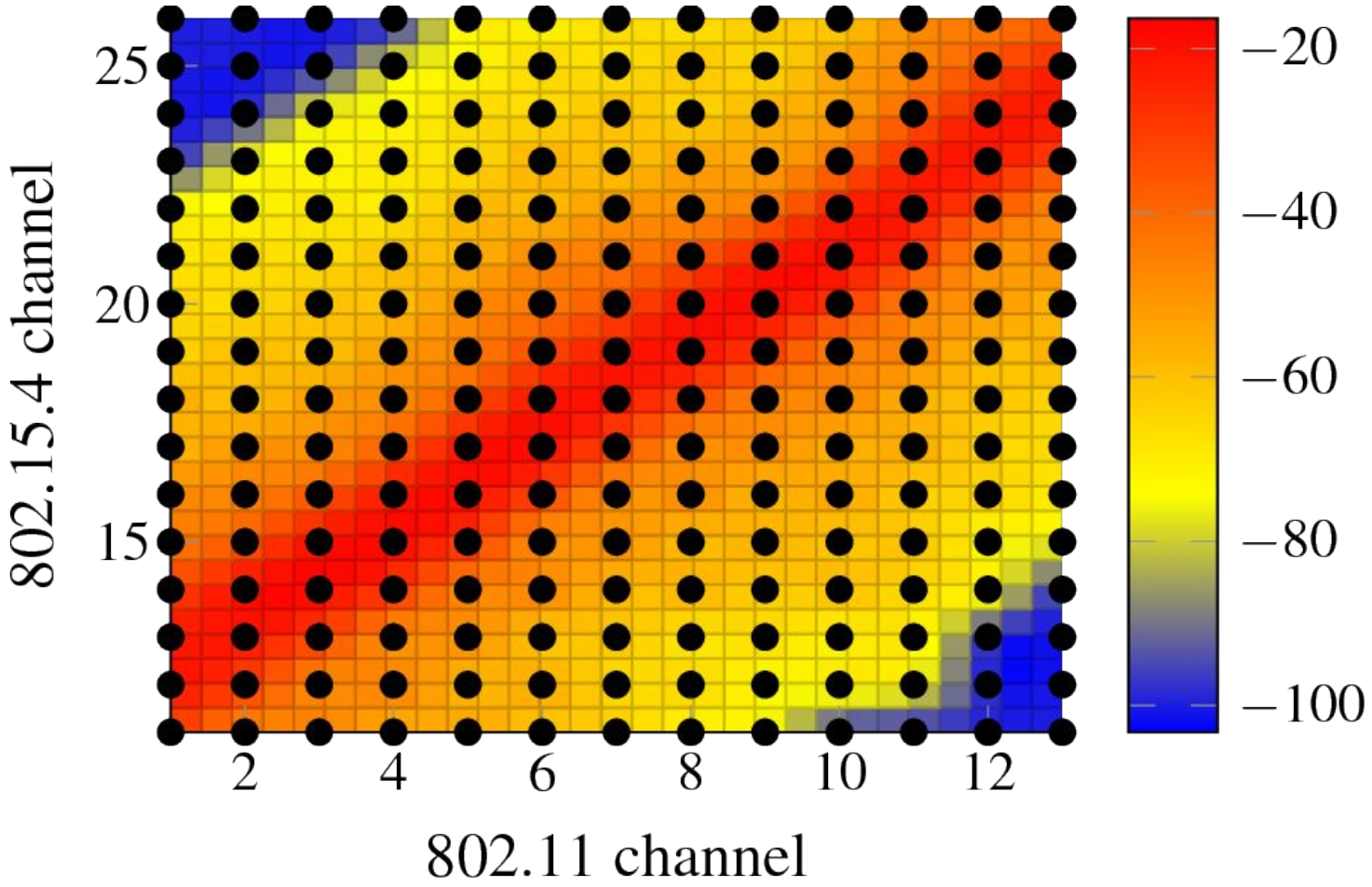
> 97 % accuracy

12Bps

Close Proximity

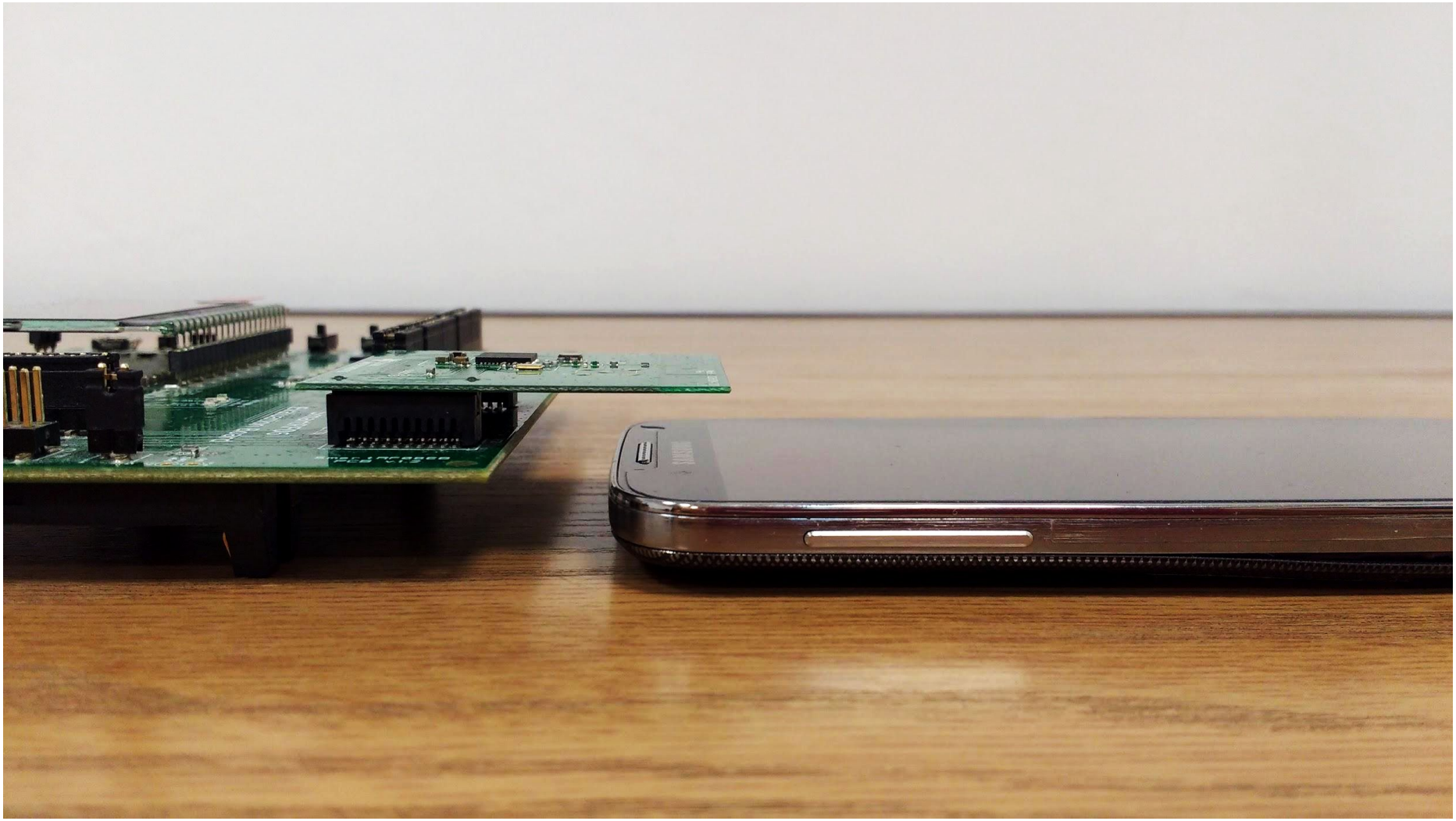


Non-Overlapping Channels

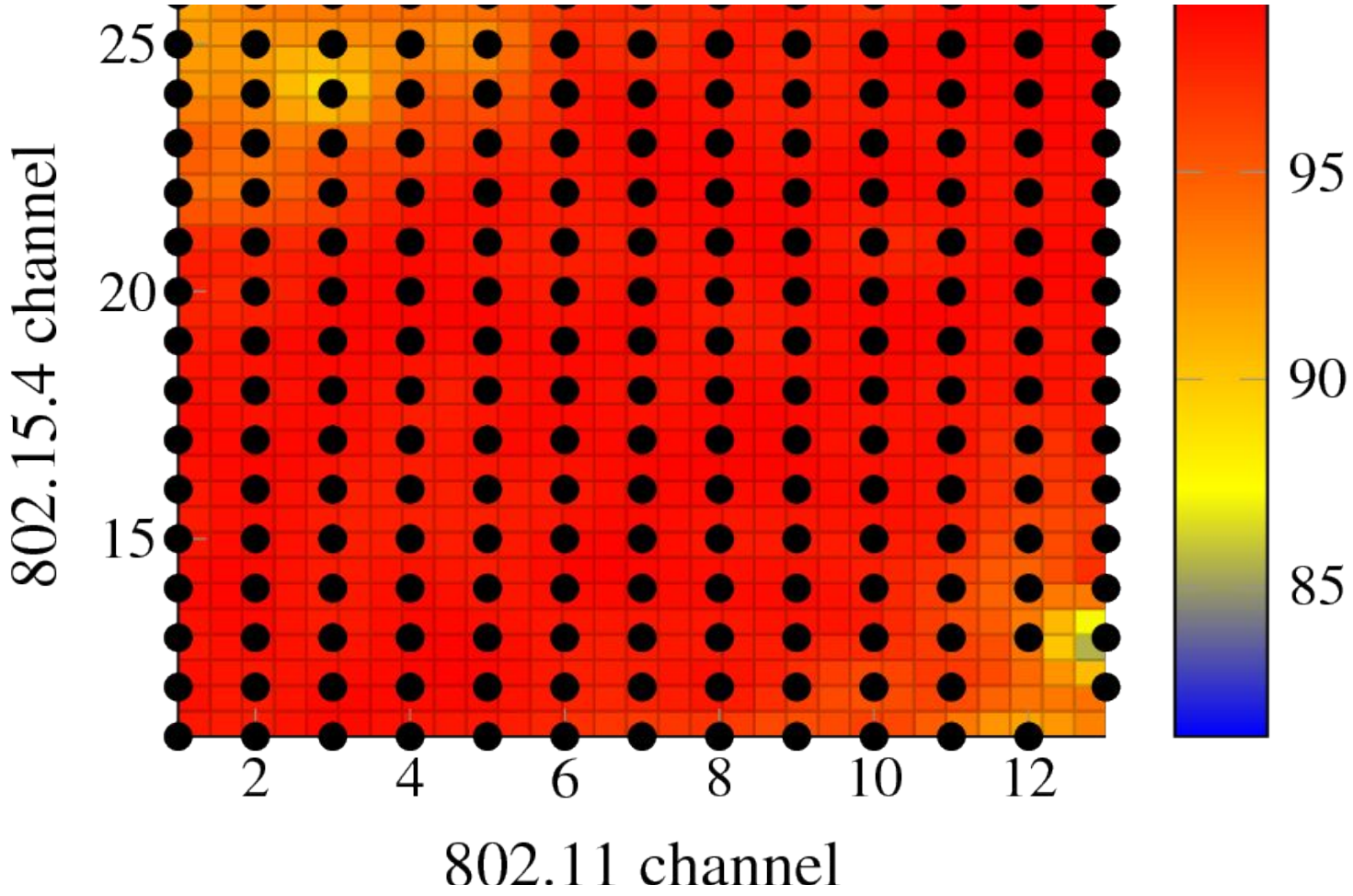


RSSI of decoded chirps for varying combinations of channels.
Quiet environment. Distance: close proximity.

Even Closer Proximity

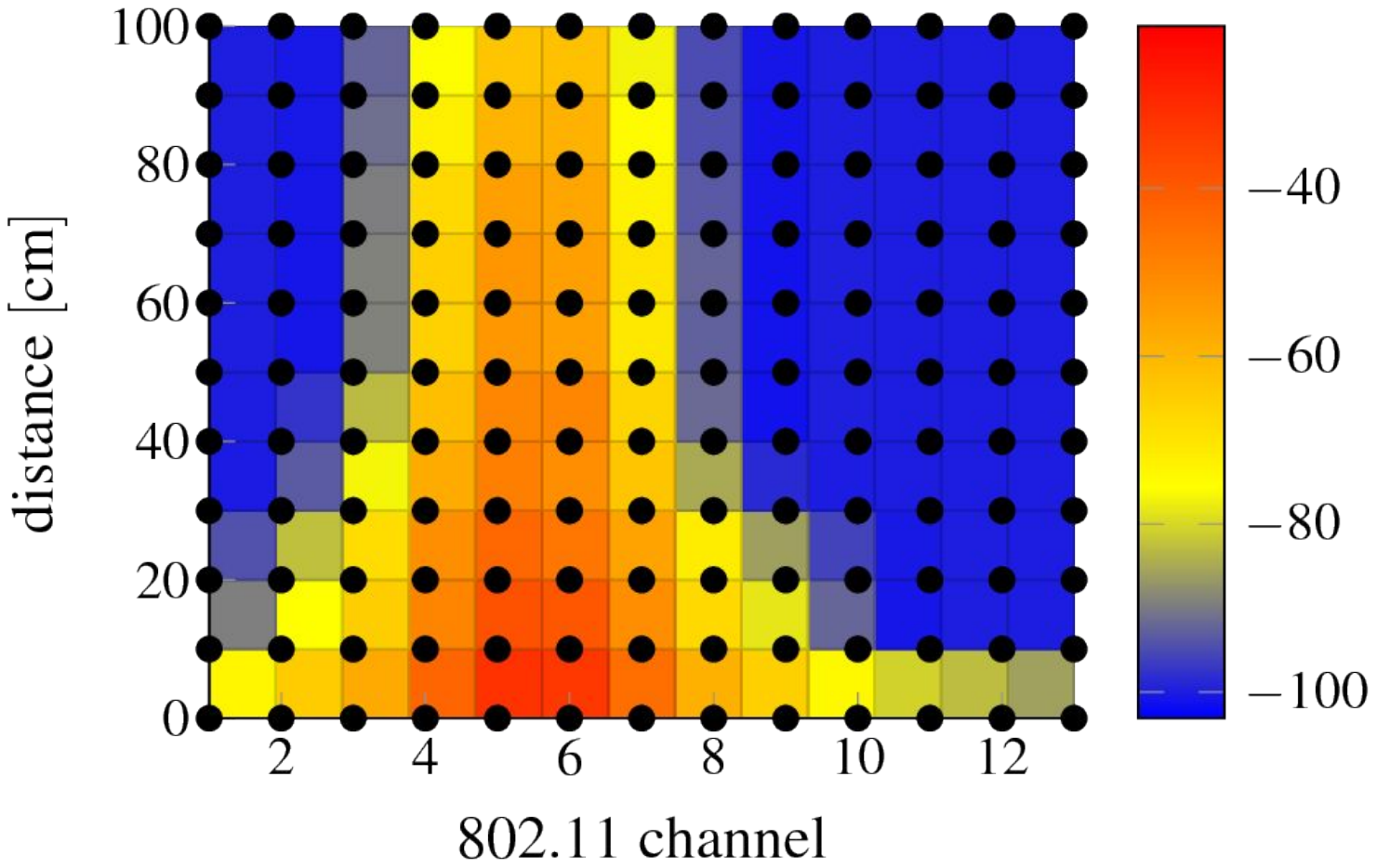


Non-Overlapping Channels



Percentage of correctly decoded packets for varying combinations of channels.
Quiet environment. Distance: even closer proximity.

Varying Distance



RSSI od decoded chirps for ZigBee listening on channel 17. Quiet environment.
For CTC above -76 dBm: 97% accuracy of communication.

Results

ZigBee should listen on one of its central channels (e.g. 17).

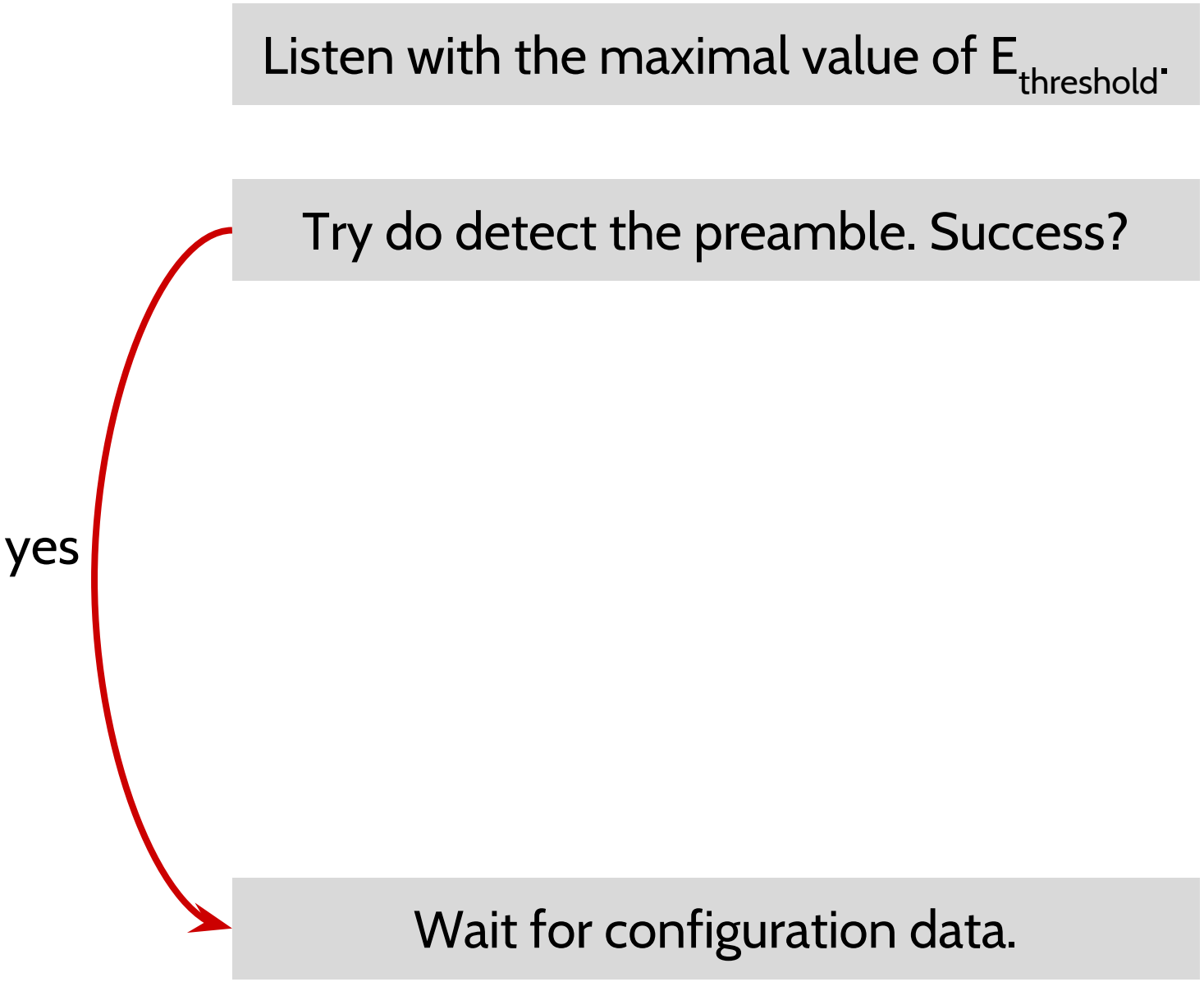
In a quiet environment CTC accuracy is above **95%** for all WiFi channels.

However it requires the devices to be very close to each other.

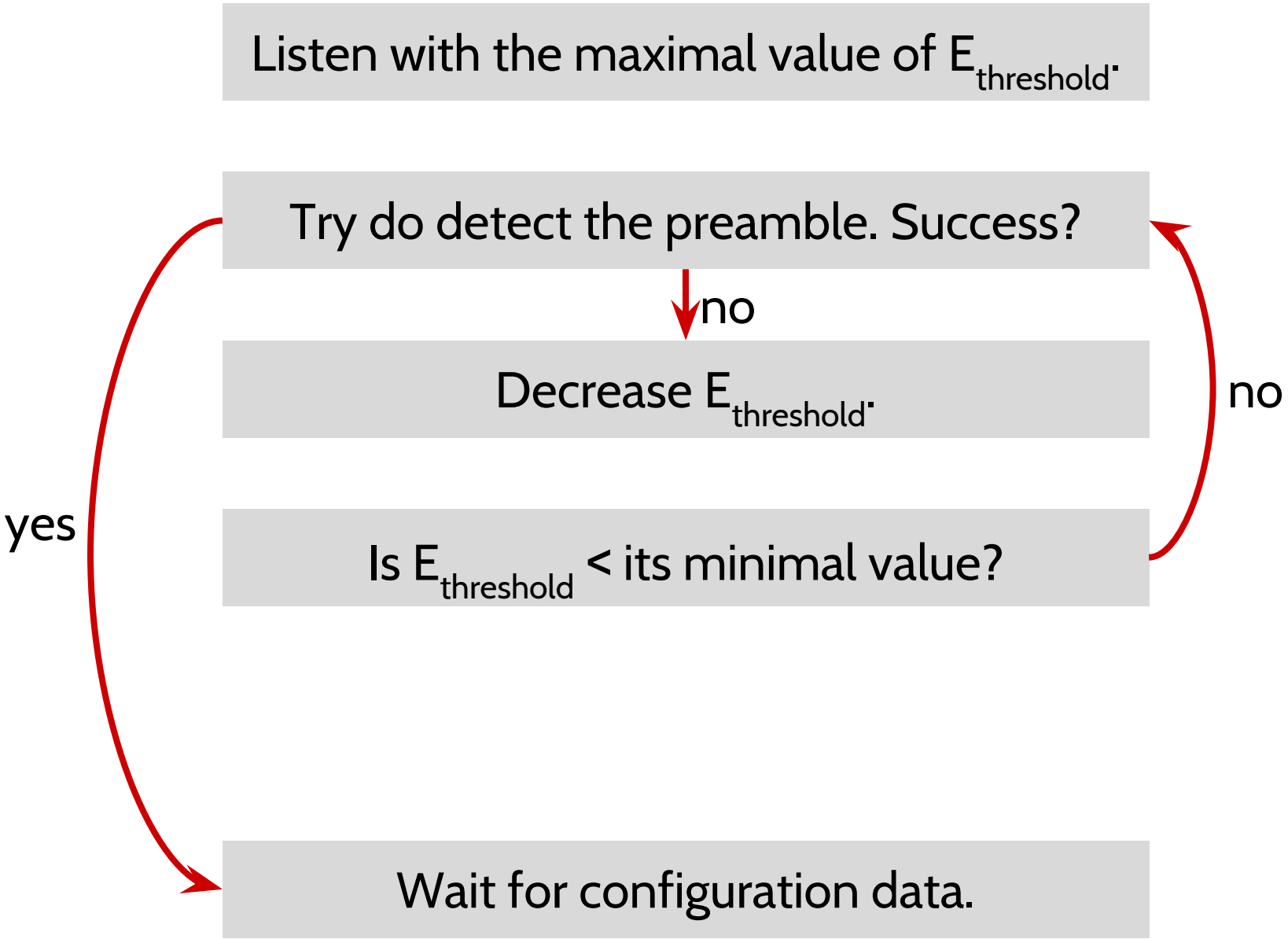
In case of noisy environments, hopping over channels might be necessary.

A lot depends on the adjusted value of $E_{threshold}$.

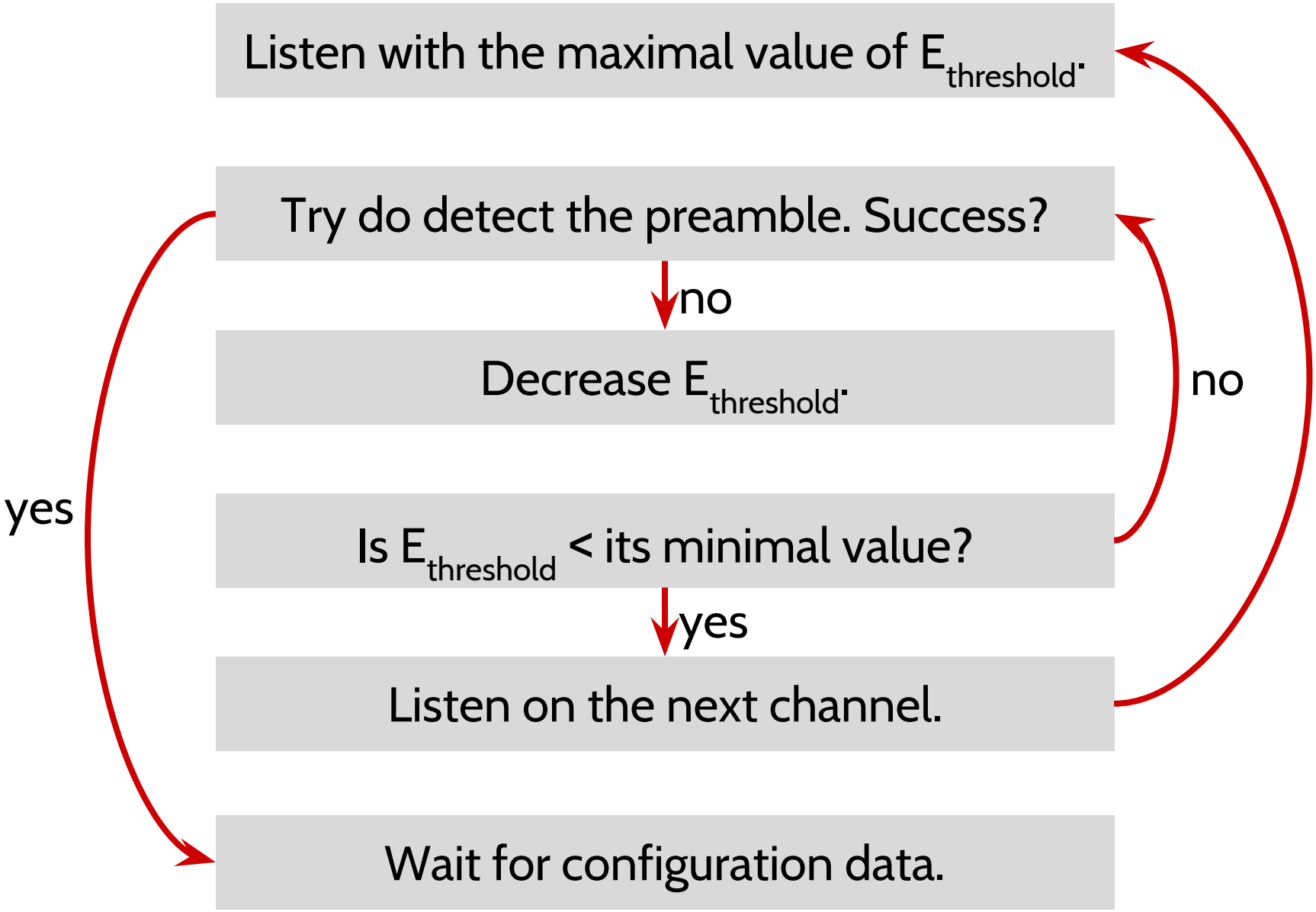
The Final Algorithm



The Final Algorithm



The Final Algorithm



It Was a Long Way

find appropriate RSSI threshold

choose inter-packets intervals

enforce the bit rate

deal with non-overlapping channels

inga.rub@mimuw.edu.pl

The presented research was supported by the National Center for Research and Development (NCBR) in Poland under grant no. LIDER/434/L-6/14/NCBR/2015.

The National Centre
for Research and Development

LIDER

The Noise Floor Estimate

Exponential moving average of detected low-energy samples:

$$E_{noise} := \min(\gamma, E_{sample} \cdot \alpha + E_{noise} \cdot (1-\alpha))$$

$$\alpha := 1/256$$

$$\gamma := -98 \text{ dBm}$$

We use every 80th low-energy sample.

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

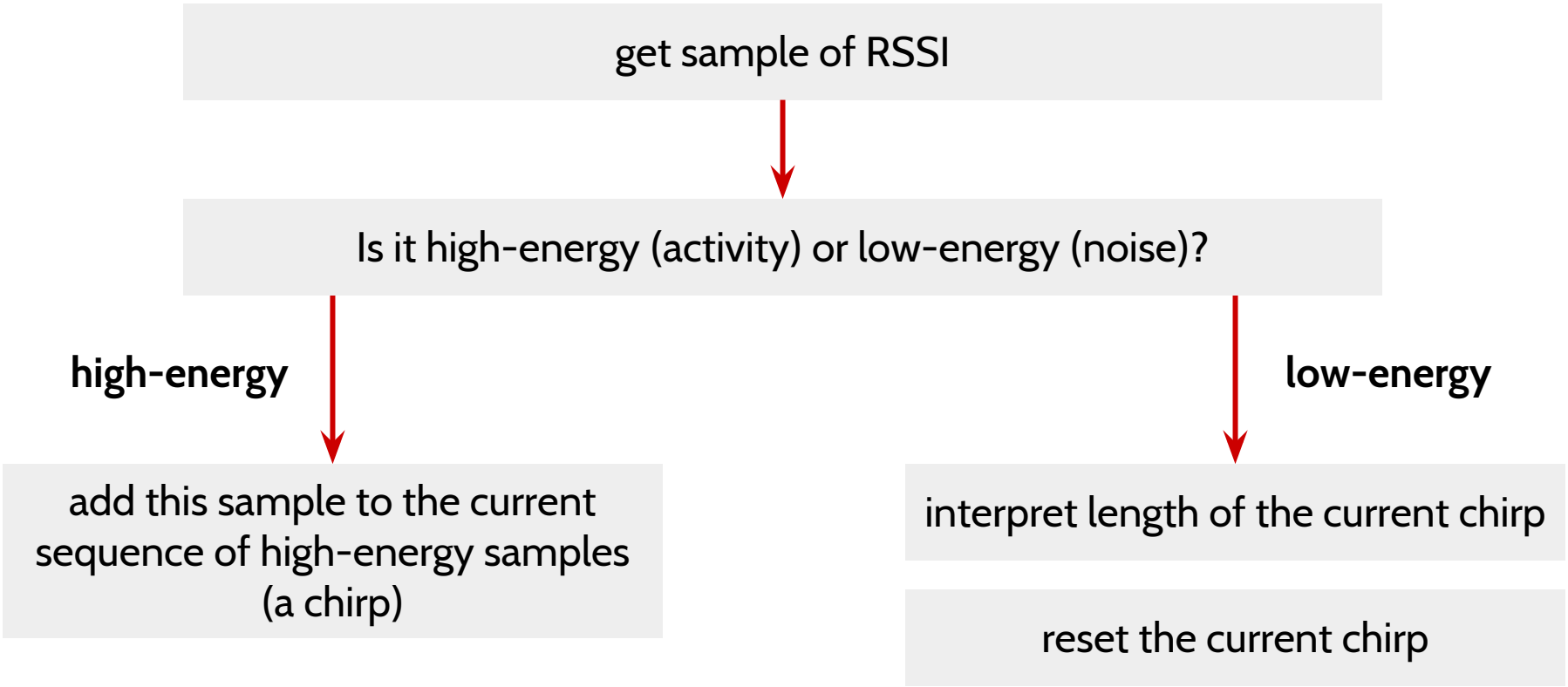
The Algorithm

(based on Esence [1] and HoWiES [2])

```
packetLength = 0
while (dataReceived) do
    sample ← getRSSI()
    if (isHighEnergy(sample)) then
        packetLength += 1
    else
        updateNoiseLevel(sample)
        interpretPacketLength(packetLength)
        packetLength = 0
    end if
end while
```

The Algorithm

(based on existing solutions: Esence and HoWiES)



The Algorithm

(based on existing solutions: Esence and HoWiES)