IRCP

Developing an interference resistant chatting protocol

Tarik Moussa
Jacob Bachmann

Agenda

- Aim
- Issues
- Problems & Approaches
- Demo
- Weaknesses

Aim

What is the aim of the protocol?

Two things:

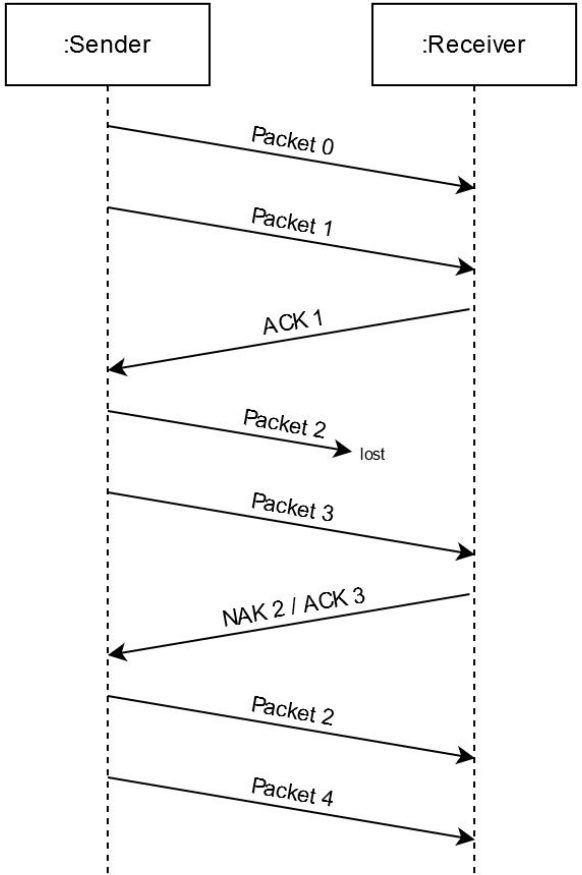
- Reliable chatting between two parties.
- Recognizing and dodging interferences.

Issues

The chatting issue

- A lossless information exchange between two nodes.
- Reliable information transmission can not be assumed.
- Integrity of transmitted packets can not be assumed.

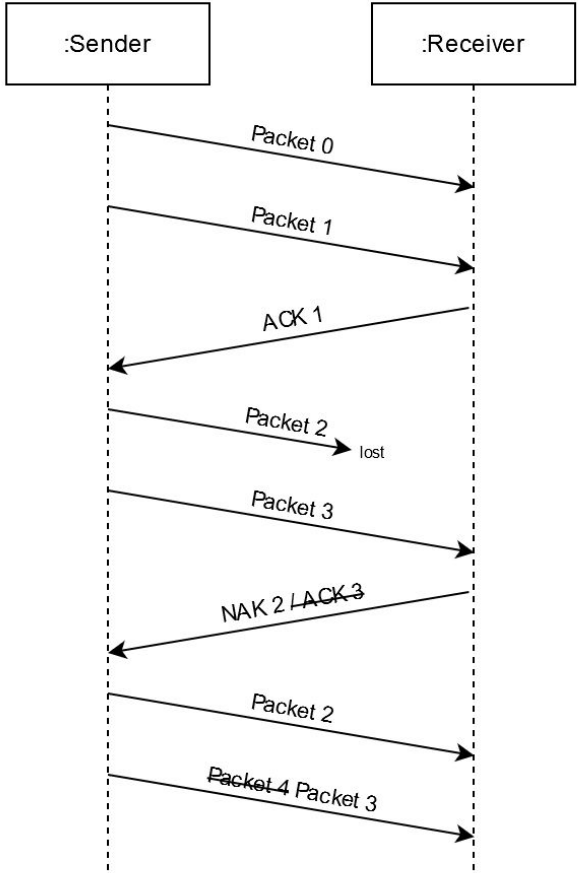
Idea: Sliding window / Go-Back-N
Idea: CRC32 checksum



The chatting issue

- A lossless information exchange between two nodes.
- Reliable information transmission can not be assumed.
- Integrity of transmitted packets can not be assumed.

Idea: Sliding window / Go-Back-N
Idea: CRC32 checksum



The interference issue

- **Interference:** is a change of a signal on the way between its source and the receiver. In other words: the addition of unwanted signals to a useful signal
- Problem: Interferer prohibits information exchange within frequency.
- Question: How to recognize presence of interferer?
- Idea: Make usage of **RSSI**.
- (**RSSI**: **R**eceived **S**ignal **S**trength **I**ndication)

Problems & Approaches

Problem 1: What RSSI value indicates an inference?

Even though we now measure the RSSI for every packet, we do not know which RSSI values to treat as an interference and which RSSI values should be considered legit.

Approach:

- Characterizing RSSI → legit RSSI has a deviation of ~9 dB
- Documentation of the CC1200 → legit RSSI has a deviation of 10-16 dB

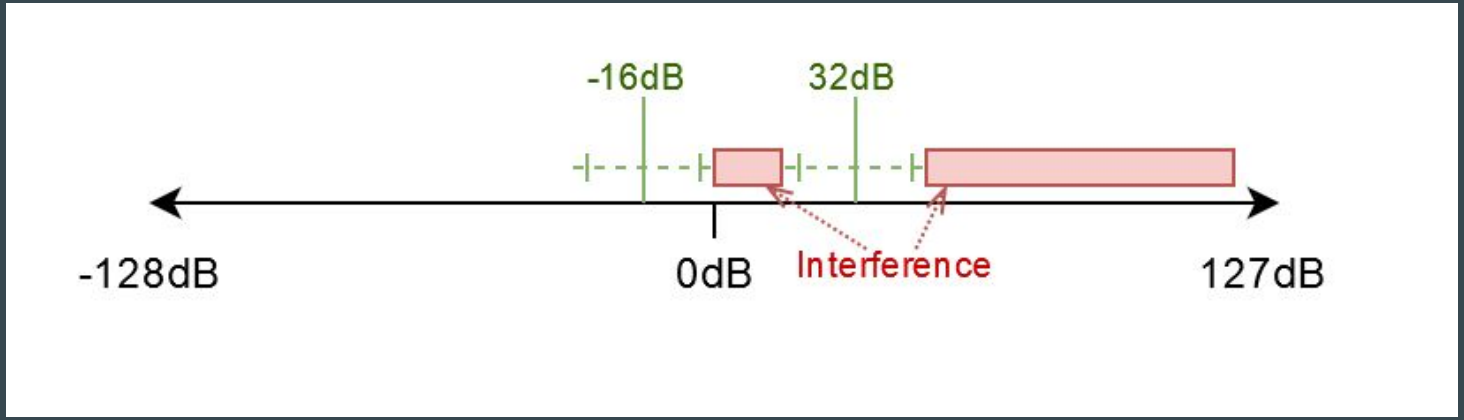
Problem 2: How do the nodes know which RSSI values are valid?

We now know how what deviation to expect, but around what RSSI value?
The nodes have to track two RSSI values of interest: RSSI-IDLE and RSSI-HIGH.

Approach:

- With handshake, nodes start tracking the RSSI values.
  → RSSI-IDLE if no packet was received
  → RSSI-HIGH if a packet was received
- With calculating the average, over time the RSSIs should even out.

Problem 1 & Problem 2: Result of approaches

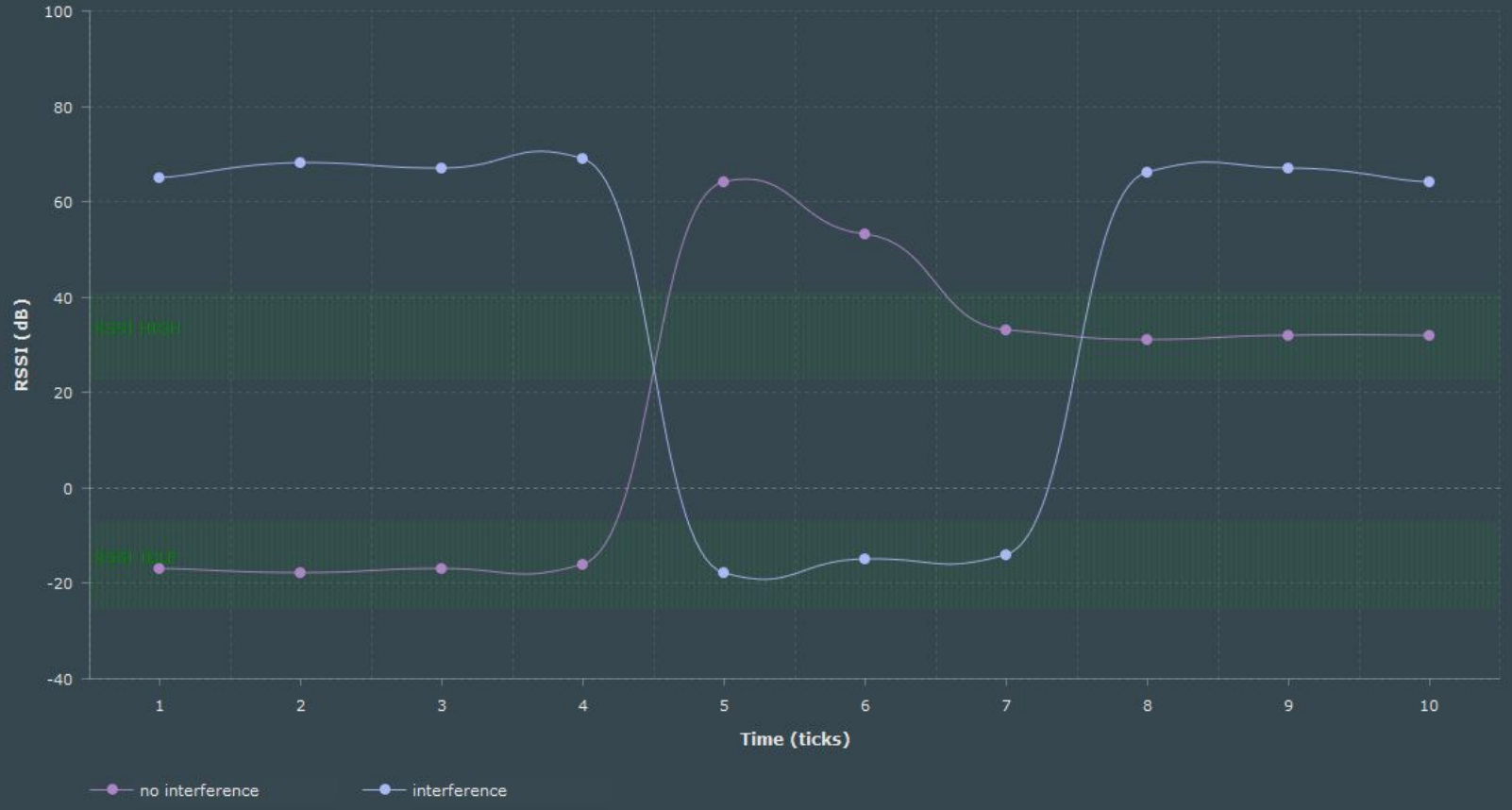


Problem 3: Sudden spikes of RSSI without interference?

Since we now have working interference recognition the next problem arises.
Sudden spikes of the RSSI without an interfering node led to wrong assumptions.

Approach:

- Tracking history of high RSSI
  → Many spikes in short time indicate the possibility of an interference


100
80
60
40
20
0
-20
-40
RSSI (dB)
1
2
3
4
5
6
7
8
9
10
Time (ticks)
no interference
interference

Problem 4: Switch to same frequency without communicating it

The last problem in our way to a successful interference avoidance was the incapability of the nodes to communicate while an interference.

Approach:

- Determine next frequency with a seed
- Seed is exchanged during handshake

Demo

Weaknesses

3 weaknesses we can think of

1. Interference prior to handshake unhandled
2. Interference recognized by only one node
   → one switches freq., one stays
3. Protocol very vulnerable to people knowing it with bad intentions
   → missing encryption
   → missing node identification (address/identifier/...)

...

Thanks for your attention!

Any Questions?