# Roboc0p - A Wi-Fi Monitor and Analyzer for RoboCup

Ricardo Dias, Ivo Angelico, Cesar Gomes

**Abstract**

One major well-known problem in the RoboCup environment is related with the WiFi medium capacity with respect to the number of devices operating in the 802.11x standards. Despite the effort in finding a standard solution for an effective way to monitor and instantly pinpoint network problems and offenders have not succeeded yet.

The objective of this project was to develop an open-source solution of a network monitor tool tailored for the RoboCup competition environments.

We believe that with this tool we will be able to quickly identify most problems related to the network and speed-up the setup time between matches in any league that is using an official WiFi Access-Point.

This project was born from the necessity of quickly identifying the root of the problem and mitigate it in the Middle-Size League (Fig. 1), but this problem most likely exists also in other leagues that are using an official WiFi Access-Point. Every aspect of the requirements for monitoring WiFi in the MSL should perfectly fit any other RoboCup league that uses WiFi.

Fig 1 / Middle-Size League finals in RoboCup 2013 – Eindhoven, Netherlands.

# Introduction

The MSL is a robotic football competition in which the games are played by two teams. The robots communicate with each other and with other specific agents (such as referees or coaches) through WiFi (IEEE 802.11a). The typical setup in competition is depicted in Fig. 2.

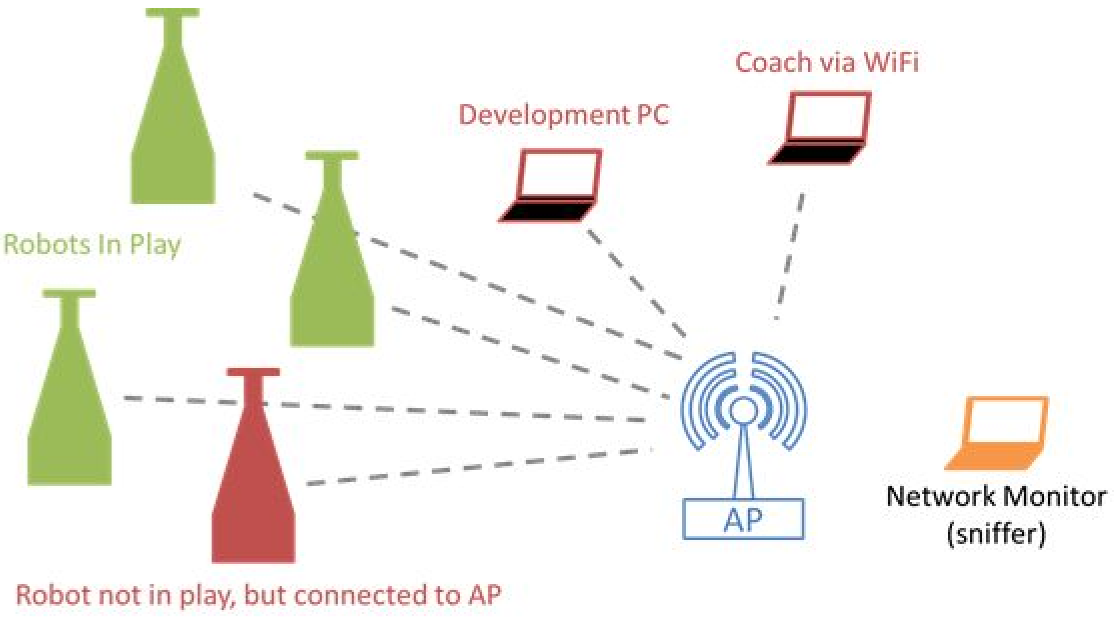

Coach via WiFi
Development PC
Robots In Play
AP
Network Monitor (sniffer)
Robot not in play, but connected to AP

Fig 2 / Final Setup – Network monitor on a match

During the game is expected that only the robots in play, coaches and the referee are connected to the network. When not in game, all teams may connect robots and development computers to do tests.

## Communication issues and rule violations

To ensure smooth game play in the competition:

- Only the robots that are playing, coaches and the referee should be connected to the network. Sometimes teams forget to disconnect development pcs and other robots which may affect the game and is not allowed by the rules.
- Teams have a limited bandwidth allowed defined by the rules to avoid network congestion.

## Solution

Develop an application that monitors the network usage, relates the connected devices with the teams at each time and provides data that allows detection of rule violation.

# Objective

In order to comply with the tight schedule and to maximize the number of matches, it is important that the whole match rolling process is streamlined. Teams should never be impaired if they can't be accountable for any problem (such as problems with the WiFi) that impacts their match. Decreasing the time period that is used to assess and solve those kind of problems, decreases the risk of delaying matches, so we believe this is the perfect opportunity to develop a network monitor solution specially tailored for the RoboCup environment.

For this application, the following requirements have been identified:

- Open-source software
- Ability to monitor a IEEE 802.11a Wireless network
- Ability to monitor the game being aware of players and teams
  - List and categorize stations with their associated team
  - Estimate the throughput usage of the players and teams
  - Provide sufficient information for referees to detect violations to the rules
- A GUI to display relevant information and provide user interaction
- Interface with the MSL Referee Box application to get information about the teams in play

# Work Plan

**Stage 1 - Preparation**

- State-of-the-art study regarding network monitoring solutions (both HW and SW)

**Stage 2 – Development**

- Implementation of a BETA version

**Stage 3 – Test**

- Test BETA version in local tournament (Portuguese Robotics Open)

**Stage 4 – Development**

- Improvements and further development for v1.0

**Stage 5 – Final Test in RoboCup 2018**

- Test v1.0 in RoboCup 2018, Montreal, Canada

# Implementation

After analysing the state of the art, we concluded that there are mostly two ways of monitoring a WiFi network: intrusive and non-intrusive.

For a better precision of the results and also to directly control the access to the field network, RoboC0p could potentially run inside a dedicated router with custom firmware that would be also the official field access-point. However, this entails that a specific router would become a

requirement for every league that would like to use RoboC0p, which is a restriction that we must avoid.
The only viable solution that allows the application to be independent from the network infrastructure and that doesn't interfere with the network is **sniffing**. To achieve this, a computer with a wireless card that supports monitor mode to sniff the packets is required.

The Roboc0p application was implemented under Linux and its UI was developed in QT4, thus C++ was chosen as the programming language to use.

## Network Monitoring

The most used software suite to monitor wireless networks is the open source suite of tools **Aircrack-ng**. In order to monitor a network, the wireless card must support monitor mode. The application **airmon-ng** of the **Aircrack-ng** suite is used to set the wireless card to monitor mode, which allows packets to be captured even if the wireless card is not associated with the access point. It is also used to know which interfaces are available so that the user may choose which one to use.

To scan for available wireless networks and gather information about them, **Iwlist** was used.

**nmcli** is a command-line tool for controlling NetworkManager and reporting network status. The version of **nmcli** used was v1.2.6, run with the arguments SSID,BSSID,CHAN,ACTIVE. It's important to note that **nmcli** is an active project currently under development and different versions of the application may take different arguments which in turn may make it incompatible with Roboc0p.

**Airodump-ng** is a packet sniffer. It allows to capture the packets filtering by the BSSID and it has information about the traffic such as the stations MAC Addresses, the number of packets sent by the stations, when were the stations first and last seen and with what power.
It was necessary to modify the application and further develop it to be able to get the size of the packets sent. With that information we are able to calculate the throughput.

Within the RoboC0p software, a class called IW is solely responsible for any interactions with the wireless interface(s). The methods of this class allow the user to get information about the interface and the networks, to control the wireless board monitor mode and to control the capture state.

## Teams Dataset

Before the competition, teams are required to provide a list of the MAC addresses of the robots and the development devices that may connect to the network during the competition in the following syntax:

- A line with “Team Name:” followed by the team name
- One mac address and an ID separated by a comma
- Lines started by # will be ignored

**Example**

```
Team Name: CAMBADA
#This are the team robots
84:3a:4b:64:76:bc,R2
84:3a:4b:65:09:f8,R3
84:3a:4b:65:0b:92,R4
84:3a:4b:65:09:8e,R5
a4:4e:31:c6:74:20,R6
```

To correlate the mac addresses information to the teams were created the objects player and team. Information from the database are uploaded to these objects.

The team object has informations about the associated players, about the game (such as if the team is on the game) and statistics about the used bandwidth.

The player object has informations about the stations that were provided by the teams and about every other station that connects to the network. If the station was on the list of devices provided by the teams the object will have the following informations: identification, mac address, the associated team and. If a station is connected to the network (regardless of being on team’s information or not) the player object will have the following informations: first and last time that packets of this stations were seen, power and statistics such the number packets and the used bandwidth.

## Middleware

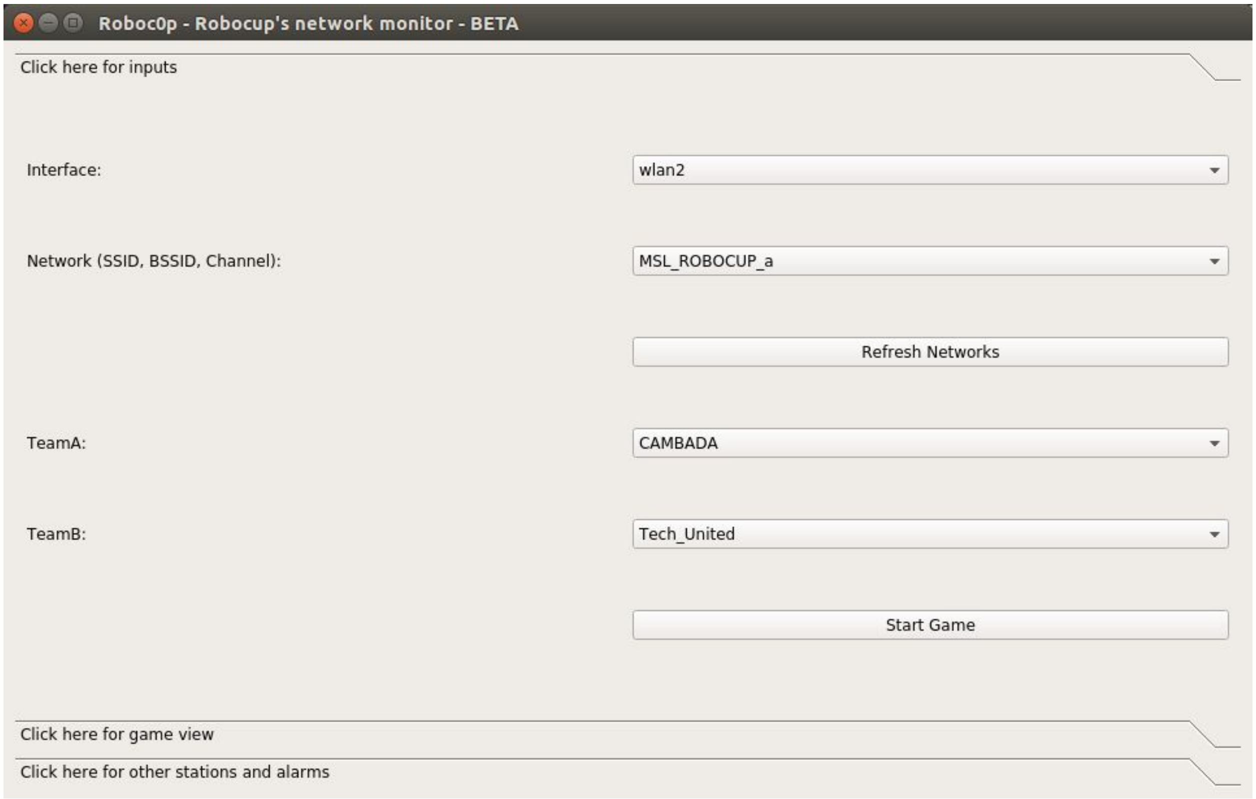
Some auxiliary functions were created and are available for future developments. The functions allow to get and display information from the database, operate over statistics and parse information from airodump.

# User Interface

The UI is divided into 3 sections:

- Configuration
- Match
- Other Stations

## UI - The Configuration Section



Roboc0p - Robocup's network monitor - BETA
Click here for inputs
Refresh Networks
Start Game
Click here for game view
Click here for other stations and alarms

Interface: wlan2

Network (SSID, BSSID, Channel): MSL_ROBOCUP_a

TeamA: CAMBADA

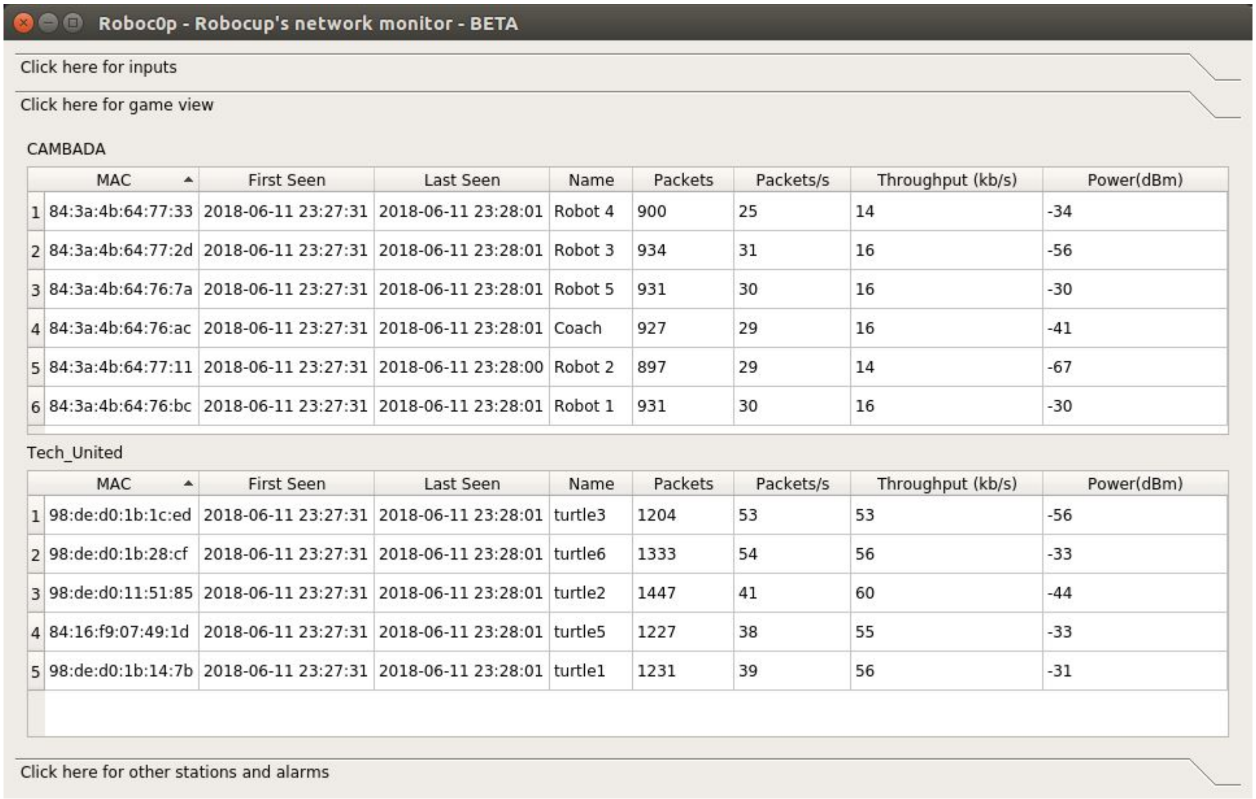
TeamB: Tech_United

Fig X / UI - Configuration Section.

This section lists the detected network interfaces. After selecting the interface to use, the application scans the available networks. A scan can also be forced by pressing the **“Refresh Networks”** button.

Selecting the teams in play is the last step before clicking the **“Start Game”** button. While the match is happening, all the previous options are disabled to prevent changing them unintentionally.

## UI - The Match Section


Roboc0p - Robocup's network monitor - BETA
Click here for inputs
Click here for game view
Click here for other stations and alarms

CAMBADA

| | MAC | First Seen | Last Seen | Name | Packets | Packets/s | Throughput (kb/s) | Power(dBm) |
|---|---|---|---|---|---|---|---|---|
| 1 | 84:3a:4b:64:77:33 | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | Robot 4 | 900 | 25 | 14 | -34 |
| 2 | 84:3a:4b:64:77:2d | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | Robot 3 | 934 | 31 | 16 | -56 |
| 3 | 84:3a:4b:64:76:7a | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | Robot 5 | 931 | 30 | 16 | -30 |
| 4 | 84:3a:4b:64:76:ac | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | Coach | 927 | 29 | 16 | -41 |
| 5 | 84:3a:4b:64:77:11 | 2018-06-11 23:27:31 | 2018-06-11 23:28:00 | Robot 2 | 897 | 29 | 14 | -67 |
| 6 | 84:3a:4b:64:76:bc | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | Robot 1 | 931 | 30 | 16 | -30 |

Tech_United

| | MAC | First Seen | Last Seen | Name | Packets | Packets/s | Throughput (kb/s) | Power(dBm) |
|---|---|---|---|---|---|---|---|---|
| 1 | 98:de:d0:1b:1c:ed | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | turtle3 | 1204 | 53 | 53 | -56 |
| 2 | 98:de:d0:1b:28:cf | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | turtle6 | 1333 | 54 | 56 | -33 |
| 3 | 98:de:d0:11:51:85 | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | turtle2 | 1447 | 41 | 60 | -44 |
| 4 | 84:16:f9:07:49:1d | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | turtle5 | 1227 | 38 | 55 | -33 |
| 5 | 98:de:d0:1b:14:7b | 2018-06-11 23:27:31 | 2018-06-11 23:28:01 | turtle1 | 1231 | 39 | 56 | -31 |

Fig X / UI - Match Section.

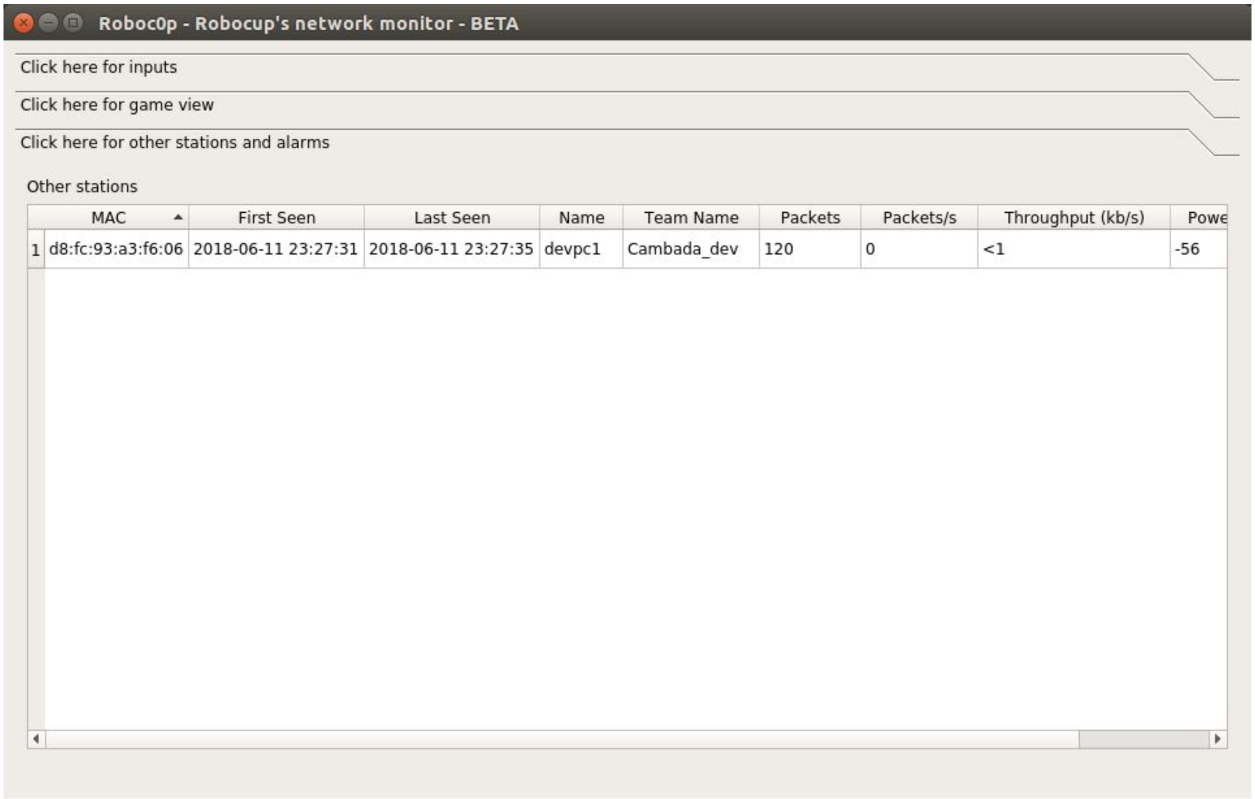
This section displays information about the playing teams connected devices.

- MAC Address
- Friendly Name (defined in the team configuration file)
- When the station was seen first
- Timestamp of the most recent packet received from that station
- Total number of packets received
- Average packets per second
- Average bandwidth kb/s
- Power (dBm)

A warning is also displayed when a team is using more bandwidth than the rules limit.

**Note**: the throughput should be seen as indicative - as wireless communication are prone to collisions and other errors, may require retries that may not be done by the users but by the drivers of the wireless hardware devices. It’s not possible to say how much bandwidth the user deliberately using.

## UI - The Other Stations Section


Roboc0p - Robocup's network monitor - BETA
Click here for inputs
Click here for game view
Click here for other stations and alarms

Other stations

| | MAC ▲ | First Seen | Last Seen | Name | Team Name | Packets | Packets/s | Throughput (kb/s) | Powe |
|---|---|---|---|---|---|---|---|---|---|
| 1 | d8:fc:93:a3:f6:06 | 2018-06-11 23:27:31 | 2018-06-11 23:27:35 | devpc1 | Cambada_dev | 120 | 0 | <1 | -56 |

Fig X / UI - Other Stations Section.

The last section displays information about devices that are connected and shouldn't - these devices are not listed under the playing teams configuration file. If available, information about the team and the device friendly name will be displayed.

# Logging

Syslog is a Linux system utility that allows system logging. For Roboc0p, were created three levels of log using syslog that allows both debugging and record the data of interest that was displayed by the application. LOG_INFO are logs of the read of teams and players from the database and when it are on game. LOG_NOTICE are logs about the game status. LOG_ALERT are logs about informations of interest for the game such as players that are on game and shouldn't or teams using more bandwidth than the one define on the rules.

# Results

The aforementioned solution was implemented and tested according to the work plan. A BETA version was provided for the Portuguese Open 2018 MSL competition. This was the first time the development team had contact with the application real environment.
As expected, during this pilot test, a few situations occurred for the first time, and some of which would be very difficult to force in lab conditions. Some fixes were provided in-loco and feedback from teams was gathered - from missing useful information to the actual User Interface.

After a new period of development, a first version was made available for the MSL community to use during RoboCup 2018 in Montreal, Canada.
Being a non-intrusive monitoring method of, the Technical Committee approved the use of this software during the whole tournament.

The most relevant results are listed below:

- The application ran during the whole tournament with no computer reboots - which proved the robustness of the solution.
- The “Other Stations Section” proved to be very useful to identify non-playing teams connected to the access-point. Teams found it was much more efficient to pin-point and mitigate network problems coming from AP connection abuses.
- A team was caught cheating and thus was disqualified in a decisive match.
- Very positive feedback received from the teams on the usefulness and need for such a monitoring solution.

# Conclusion and Future Work

This report covers the RoboC0p application - a solution developed to monitor the WiFi network within the RoboCup environment. The main objective was to quickly identify and mitigate the root of network problems, which is a problem that affect most RoboCup leagues that share an Access-Point between teams.
The results show the effectiveness of the solution with very positive feedback from the MSL teams. Furthermore, a significant effort was put in keeping the application as league-agnostic as possible.
As future work, a better integration with the MSL Referee Box application is suggested - teams in play can be automatically selected. A dynamically configurable definition of league rules is still required, for example for the bandwidth limit.