**High-Level Design**

**Document for**

**Wifi-Sniffer**



**Revision 0.1**

**8/11/2019**

Tandem Group Software Proprietary & Confidential

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Date** |  | **Contents** | **Author(s)** |
| 0.1 | August 11, 2019 | Initial version |  | Gal Zur and Matan Grynbaum-Nachmias |

# Table of Contents

Revision History **2**

Abbreviation 5

Terms 6

Overview **7**

**System and Architecture Requirements 11**

System Requirements 11

Architectural Requirements 11

General 11

Constraints 11

# List of Tables​

|  |  |
| --- | --- |
| Table ​0-1: Reference Documents | 5 |
| Table ​1-1: List of Abbreviations | 5 |
| Table ​1-2: List of Terms | 6 |
| Table 2: Control and Execution Communication Protocol | ​ 9 |

# List of Figures

|  |  |
| --- | --- |
| Basic Wi-Fi Sniffer Diagram | 7 |
| Control and Execution Diagram | 8 |
| File Transferring in the Wi-Fi Sniffer Diagram | 10 |

# Reference Documents

|  |  |
| --- | --- |
| **Item #** | **Name** |
| **1** | Wi-Fi sniffer – functional requirements |

**Table** ​**0-1: Reference Documents**

## 1.1 *Abbreviation*

|  |  |
| --- | --- |
| ***Abbreviation*** | ***Description*** |
| TBD | To Be Decided - where shown item still has not been decided |
| RPi | Raspberry Pi 3 |
| HPC | Host PC |
| Network Segment | The network sniffed by the sniffer, including the AP and the wifi devices. |
| HTCPC | Host TCP Client - The TCP client program in the HPC |
| RTCPS | Raspberry TCP Server - The TCP server program in the RPi |
| WMP | Wifi Monitoring Program - The program that monitors the wifi traffic in the Network Segment |
| Pipe | The pipe channel connecting the RTCPS to the WMP |
| Init file | The file containing all of the initializing settings. |
| HUDPC | Host UDP Client - The UDP client program in the HPC |
| HUDPS | Host UDP Server - The UDP server program in the HPC |
| RUDPC | Raspberry UDP Client - The UDP client program in the RPi |
| RUDPS | Raspberry UDP Server - The UDP server program in the RPi |
| WS | Work Station - the Wi-Fi devices that the Wi-Fi Sniffer listens to. |

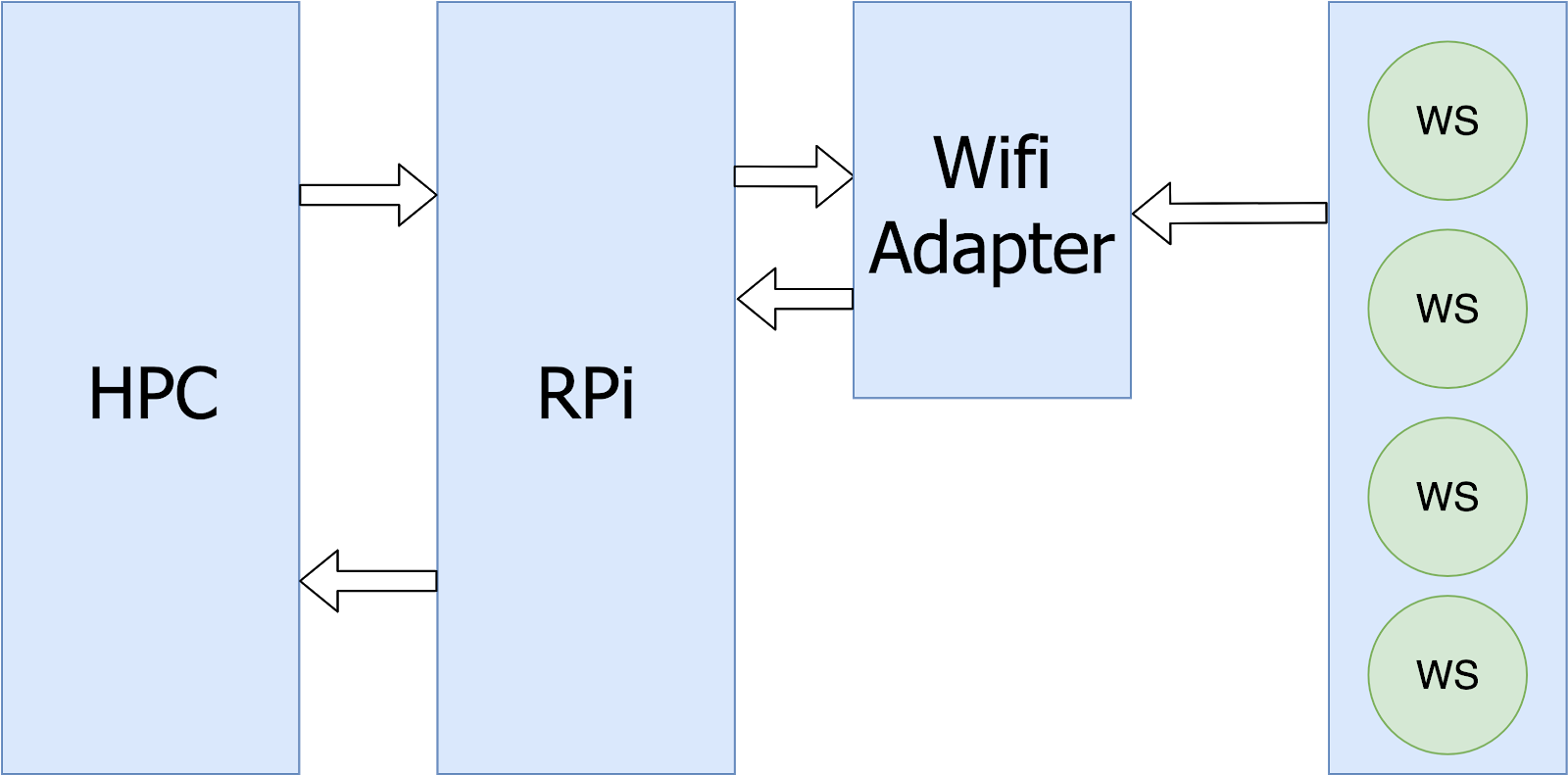
**Table** ​**0-2: List of Abbreviations**

## 1.2 *Terms*

|  |  |
| --- | --- |
| ***Term*** | ***Description*** |
| TCP | Transmission Control Protocol – reliable, ordered and error-checked data protocol – meant to transfer critical data |
| UDP | User Datagram Protocol - meant to transfer data where it is not crucial that all of the data will reach the target computer. |
| AP | Access Point – a device which opens a Wi-Fi network and allows devices to connect to it and allow for data transfer to\from the device |
| Wi-Fi device | Endpoint - any device which opens a Wi-Fi network and allows devices to connect to it and allows for data transfer to\from the device |
| Host-PC | the PC that the system will be connected to - a PC with display, keyboard etc. |
| Wireshark | Program - de-facto standard of analysis of network data - open-source |
| Raspberry-Pi | A System-On-Module - basically a low-performance fully-capable PC |
| Socket | Programmatic connection between two SW elements (running on the same or different systems) that allows data transfer |
| JSON | JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types |

**Table** ​**0-3: List of Terms**

## 2 Overview



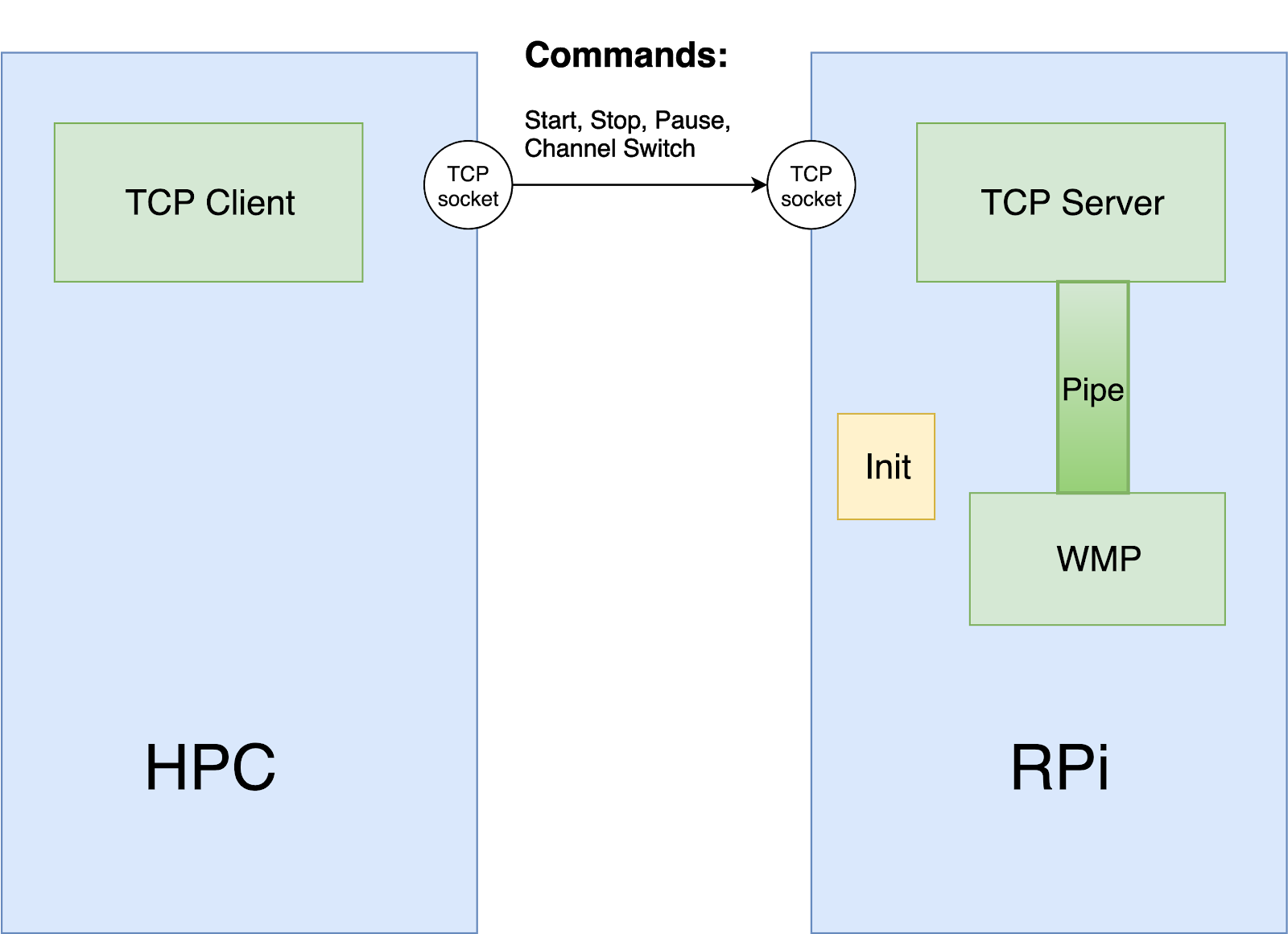
This system performs Wi-Fi Sniffing of Wi-Fi devices by the HPC using the WMP in the RPi. The RPi is connected via a physical connection to the Wi-Fi adapter. Each arrow in the diagram represents information sent from one device to another.

While the RPI comprises the WMP, the HPC runs wireshark via an implemented plugin, which could be developed in C.

Each one of the next couple of diagrams will explain one of the subsystems.

The overview can be a bit shorter, describing more about the high level flow of packets in the air received, processed and transmitted, and the configuring the Rpi.

I would expect to see the communication means between the components above the connecting lines, for example ethernet between Rpi and Wifi adapter.



The HPC runs Wireshark and controls the WMP by passing the commands using TCP. The HTCPC sends the command, that was entered by the user (from the list of commands mentioned above) to the RTCPS. The RTCPS uses the Pipe to execute the command requested by the user.

Well described. Long acronyms :D

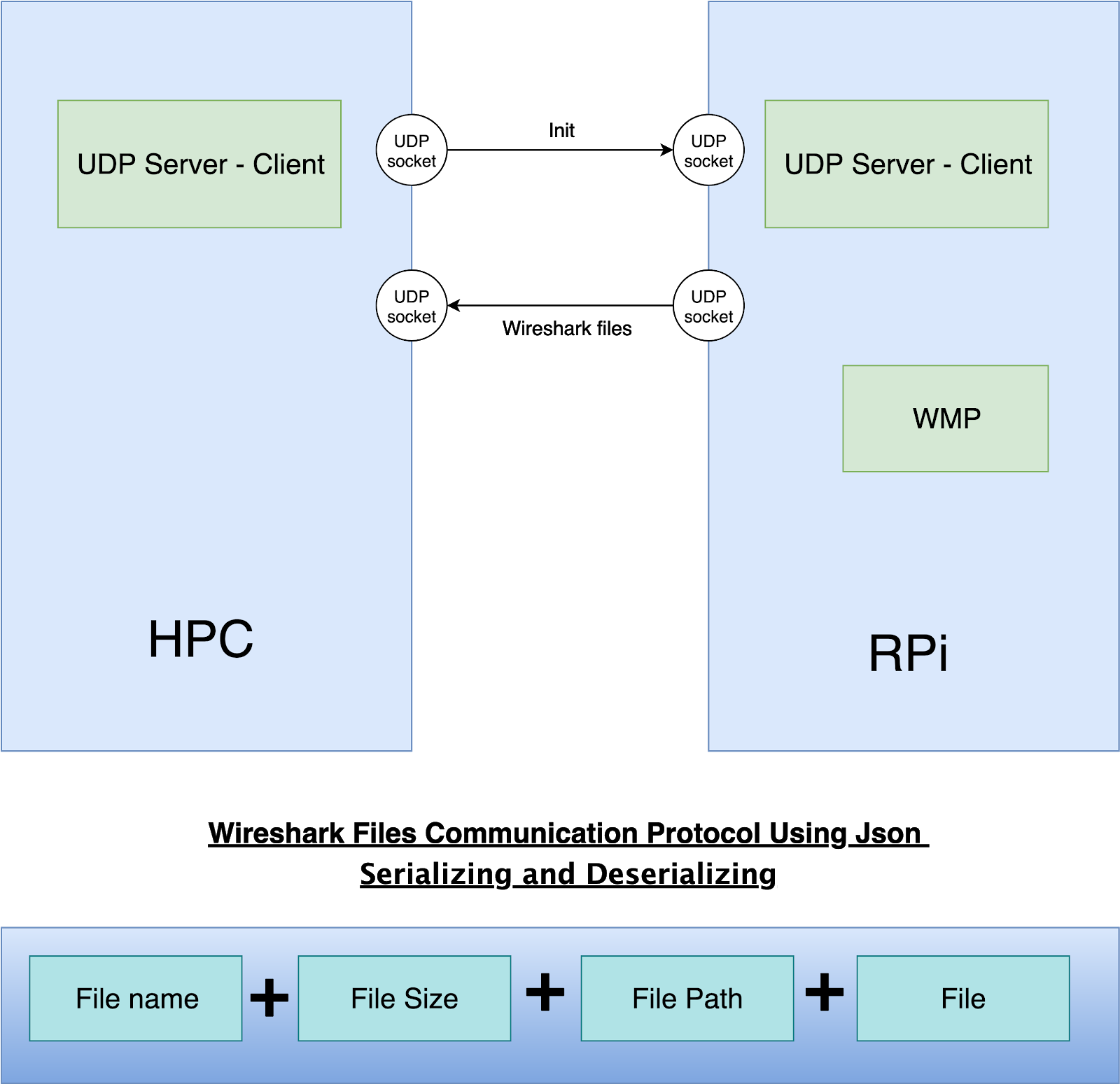
## Communication Protocol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command | Sender | Receiver | Description (Goal) | The Role of the Pipe |
| Start | HPC | RPi | Reading the Init file then activating of the WMP as a new process and creation of the Pipe. |  |
| Stop | HPC | RPi | Stopping the WMP and ending the process of the  WMP. | Sends the Stop command through the Pipe then ending the process. |
| Pause | HPC | RPi | Stopping the WMP loop. | Sends the Pause command through the Pipe - by doing so, stopping the monitoring loop. |
| Switch  Channels | HPC | RPi | TBD | TBD |

This would be much more meaningful as sequence diagram/s.

There should also be a proper structure for the message format. The TCP is the low level network protocol, but how is the message formatted in the application level? Is it a JSON? A bare string? A number? An object?

**File Transfer**



File transferring in the Wi-Fi Sniffer is done using servers and clients. At first, the Init file is transferred to the RPi using the HUDPC and the RUDPS, then when the WMP sniffs data, using the Wi-Fi adapter, it sends the Wireshark files over to the HPC using the RUDPC and the HUDPS. The Wireshark files will be sent in a JSON format in the way mentioned above so the Wireshark plugin will be able to read it properly and in an easy fashion.

## 3 System and Architecture Requirements

### 3.1 System Requirements

* Raspberry Pi 3 b+ / Raspberry Pi 4
* Wi-Fi Adapter (specific type unknown)

### 3.2 Architectural Requirements

***3.2.1 General***

#### 3.2.2 Constraints

##### 3.2.2.1 Coding Language

The servers, clients and WMP will be programmed using Python and the Wireshark plugin will be programmed using C.

##### 3.2.2.2 Operating​ ​System

The HPC’s operating system still needs to be discussed by reviewing the implications of each solution. As for the RPi the operating system used will be Raspbian.

I would like to see more on the flows within the programs and their class components. To understand the viability of this project, you must go into lower implementation details, how the plugin processes the JSON files and what Wireshark requires for a plugin to implement. How the firmware processes packets, and how it behaves in different configurations.

Standard UML class diagrams and sequence diagrams could be useful.