**Customer Requirements Specification**

**(Lastenheft)**

Author: Lucas Kaczynski

(TINF20C, SWE I Praxisprojekt 2021/2021)

Project: Websockets im LwIP HTTP Server

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 4 (Laura Reeken, Benjamin Esenwein, Yannis Plaschko, Maximilian Meier,

Lucas Kaczynski, Isabel Schwalm)

Rotebühlplatz 41

70178 Stuttgart

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 20.09.2021 | Benjamin Esenwein | Created |
| 0.2 | 27.09.2021 | Team 4 | Formulated the general components of the project using the CRS template |
| 0.3 | 05.10.2021 | Lucas Kaczynski | Cleaned up the goal description. Added a Product Environment description. |
| 0.4 | 15.10.2021 | Team 4 | Content review. Improved spelling and grammar. Added English translation. |
| 0.5 | 22.10.2021 | Team 4 | Document clean-up |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### Offene Punkte

In diesem Abschnitt sollen alle Probleme und offenen Fragen gesammelt werden. Bei einem fertigen Lastenheft sollte er leer sein, aber bei Zwischenversionen kommt diesem Abschnitt besondere Bedeutung zu!

CONTENTS

*1.* Goal 3

2. Product Environment 4

3. Product Usage 7

*3.1.* Business Processes 7

*4.* Other Product Characteristics 8

4.1. System Environment 8

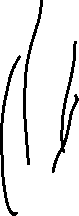
5. References 9

# Goal

LwIP is focused on consuming as few resources as possible while providing a fully functional TCP/IP stack. Because of its high efficiency, LwIP is used for example by Intel, Xilinx and Analog Devices in embedded systems [1]. For example, LwIP can be used on an ESP8266 [2].

The goal of this project is to address the architectural deficiencies of patch "#9525 (httpd: add websocket support)" [3] in coordination with the project community. This experimental base is to be improved and brought through the approval process in the open-source project.

Furthermore, a demo server is to be designed and implemented in a virtual environment under Windows.

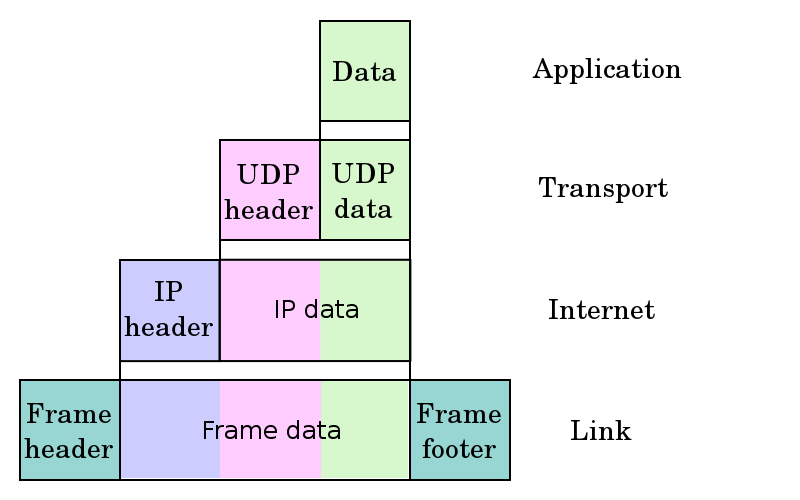


For demonstration and testing purposes of the features, a GUI-based test client shall be designed and implemented.

# Product Environment

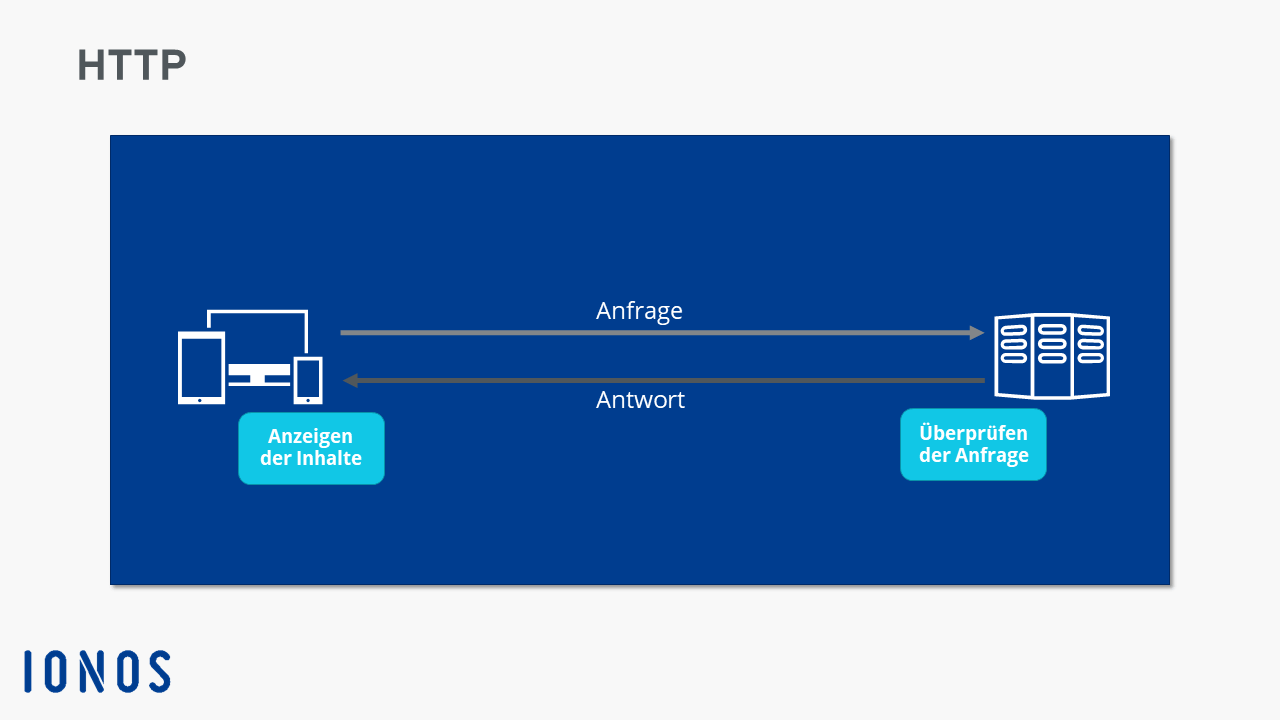
TCP/IP is a collection of protocols that enables communication between different Internet-enabled devices. TCP/IP defines how information is packetized, addressed, transmitted, routed and received.

TCP/IP is divided into four layers. The lowest layer is the link layer. It contains all hosts that a device can access within its network without going through a router. Data packages can be exchanged via the link layer within this local or virtual network. One layer above is the internet layer. With this it is possible to transfer data packages from one network to another network. This process is called routing. Located above the internet layer is the transport layer. This is where the host-to-host connections take place. These end-to-end message transfer services are independent of their underlying network. The top most layer is the application layer. This is where process-to-process data exchange for applications takes place. Applications can exchange information through already established connections in the lower layers. The Hypertext Transfer Protocol (HTTP) and the File Transfer Protocol (FTP), for example, run via the application layer [4].



Picture 1. https://en.wikipedia.org/wiki/File:UDP\_encapsulation.svg

TCP/IP establishes a connection between two communication endpoints. These communication endpoints are also called sockets. Data can be sent and received by both parties. However, this procedure causes delays because the client must first send a request to the server before the web page can be sent to the client.



Building on this, websockets extend this connection and make it bidirectional. Data can be sent and received between the sockets simultaneously. A client gets the web page immediately when it opens a connection to the server.

The connection between the server and the client remains open. Thus, the server can send information to the client without the client needing to request it. This process can be seen in figure 1 [5].

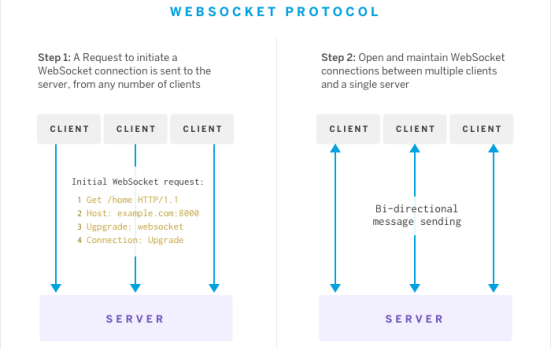


Figure 1. Websocket Protocol.

TCP/IP used to run over http version 1.0. In the meantime, http version 2.0 has been released, which enables faster connections. A comparison of the two versions can be found in Table 1.

|  |  |  |
| --- | --- | --- |
| Feature | http 1.0 | http 2.0 |
| Datastream | Multiple TCP connections are opened for different page elements (JS, CSS, image files). | A TCP connection is opened over which several page elements can be transferred in parallel. |
|  |  |  |
| compression | Data is transmitted uncompressed. | Data is compressed into binary code and then transmitted. |
| prioritization | Data packages are not prioritized. | Data packages are prioritized. |

Table 1. Comparisson between http version 1.0 and 2.0 [6], [7].

# Product Usage

The user is given the possibility to use websockets within lwIP. For this purpose, the documentation of lwIP is supplemented with instructions on the use of websockets and code examples. The user can set up a demo server running a rudimentary HTTP server whose communication is realized using a websocket. The HTTP server can be accessed through a simple GUI and allows the user to check the features of the HTTP server.

## Business Processes

This is an open-source project. Accordingly, profit is not the primary interest of this project. The project is developed with the aim to represent a time and cost saving for companies and developers.

# Other Product Characteristics

The reduced resource consumption while providing a full-fledged TCP/IP protocol characterizes the lwIP TCP/IP stack. This makes lwIP suitable for use in embedded systems with minimal RAM and ROM availability. The attractiveness of this product clearly lies in its open-source availability. This promotes the dissemination and further development of the product.

IwIP is a free software written in the C programming language.

## System Environment

See 4. Product Data

* Server for Demonstration usage: Virtual Windows Environment (Orientation at ESP8266-Hardware and -Environment)
* Client: Virtual Windows Environment

# References

[1] ‘lwIP’, *Wikipedia*. Sep. 27, 2021. Accessed: Oct. 15, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=LwIP&oldid=1046741291

[2] lujji, ‘HTTP server with WebSockets on ESP8266’, *lujji*. http://lujji.github.io/blog/esp-httpd/index.html (accessed Oct. 15, 2021).

[3] ‘lwIP - A Lightweight TCP/IP stack - Patches: patch #9525, httpd: add Websocket support [Savannah]’. https://savannah.nongnu.org/patch/?9525 (accessed Oct. 15, 2021).

[4] ‘Internet protocol suite’, *Wikipedia*. Oct. 13, 2021. Accessed: Oct. 15, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Internet\_protocol\_suite&oldid=1049680584

[5] ‘Was ist WebSocket?’, *IONOS Digitalguide*. https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-websocket/ (accessed Oct. 15, 2021).

[6] ‘HTTP/2 und SEO: Welche Vorteile hat das Protokoll gegenüber HTTP 1?’, *seonative*, Dec. 11, 2019. https://www.seonative.de/http2-und-seo-welche-vorteile-hat-das-neue-protokoll/ (accessed Oct. 15, 2021).

[7] CDN77, ‘HTTP/2 technology demo’. http://www.http2demo.io/ (accessed Oct. 15, 2021).