Computer Networks (*Redes de Computadores - RCOMP*) - 2022/2023

# The Shared Board Protocol (SBP)

Application Protocol

# Introduction

The Shared Board Protocol (SBP) facilitates secure and efficient data exchanges between the Shared Board App and the Shared Board Server network applications. This protocol is based on the reliable Transmission Control Protocol (TCP) and serves as a standardized communication framework for collaborative board management.

1. **SBP Description**

**2.1. TCP Connection Establishment**

Prior to any data exchange, the client application (Shared Board App) and the server application (Shared Board Server) establish a TCP connection.

The client initiates the TCP connection by sending a connection request, and the server accepts incoming connections.

Once established, the TCP connection remains active while the client application is running.

**2.2. Client-Server Model**

SBP follows a client-server model where the client application initiates requests, and the server application processes and responds to those requests.

However, once the TCP connection is established, both the client and server applications can initiate data exchanges by sending requests to each other.

Every request sent by either the client or the server requires a corresponding response from the receiving party.

**2.3. SBP Message Format**

All data exchanges in SBP adhere to a standardized message format.

The message format consists of the following fields:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Offset (bytes)** | **Length (bytes)** | **Description** |
| **VERSION** | 0 | 1 | SBP message format version. This field is a single byte and should be interpreted as an unsigned integer (0 to 255). The present message format version number is one. |
| **CODE** | 1 | 1 | This field identifies the type of request or response, it should be interpreted as an unsigned integer (0 to 255). |
| **D\_LENGTH\_1** | 2 | 1 | These two fields are used to specify the length in bytes of the DATA field. Both these fields are to be interpreted as unsigned integer numbers (0 to 255).  The length of the DATA field is to be calculated as:  D\_LENGTH\_1 + 256 x D\_LENGTH\_2  The length of the DATA field may be zero, meaning it does not exist. |
| **D\_LENGTH\_2** | 3 | 1 |
| **DATA** | 4 | - | Contains data to meet the specific needs of the participating applications, the content depends on the message code. |

*Table 1- SBP message format*

The VERSION field indicates the SBP message format version being used (presently version one).

The CODE field specifies the type of request or response being sent (see Table 2 for some fundamental message codes).

D\_LENGTH\_1 and D\_LENGTH\_2 together determine the length of the DATA field.

The DATA field holds application-specific data required for processing the message.

If the DATA field is not needed for a particular message, its length can be set to zero.

## SBP message codes

Table 2 contains a list of some fundamental message codes that are implemented on application using SBP.

|  |  |
| --- | --- |
| Code | Description |
| 1 | End session |
| 2 | Success |
| 3 | Failure |
| 4 | Authentication Request |
| 10 | Create a Board Request |
| 20 | Share Board Request and List of Boards Owned by User |
| 21 | Activate Users Request |
| 22 | Share a Board with User Request |
| 30 | Arquive Board Request and List of Boards Owned |
| 31 | Arquive Board with Board Already Chosen |
| 40 | Add Post-It Request and List of Boards User is Present |
| 41 | Add Post-It with Board Already Chosen |
| 50 | Modify Post-It Request and List of Boards User is Present |
| 51 | Modify Post-It with Data for Post-It |
| 60 | Undo Last Change in Post-It Request and List of Board the User is Present |
| 61 | Undo Last Change in Post-It with Board and Data Already chosen |

*Table 2- SBP message codes*

These message codes enable various operations and interactions within the Shared Board App and the Shared Board Server.

# User authentication

Upon establishing the TCP connection, the Shared Board App client is required to authenticate the local user by sending an AUTH request.

Until successful authentication, the server must ignore any requests from the client with a code value above four and respond with an ERR message.

User authentication is accomplished by including a username and password pair in the AUTH request's DATA field.

The username and password are represented as two null-terminated ASCII strings.

The server responds to the AUTH request with an ACK if the authentication is successful or an ERR if it fails.

In case of authentication failure, the client can attempt additional AUTH requests with different credentials.