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

Integrative Project of the 4th Semester

Application Protocol

1. The Shared Board Protocol (SBP)

The purpose of this application protocol is facilitating data exchanges between the ***Shared Board App*** and the ***Shared Board Server*** network applications.

1.1. SBP description

- Is TCP (Transmission Control Protocol) based, therefore, prior to any actual data exchange, a TCP connection must be established.
- Uses the client-server model. The **client application** (***Shared Board App***) is the one that takes the initiative of requesting a TCP connection establishment with the counterpart **server application** (***Shared Board Server***), which should accept incoming connection requests.
- Once the TCP connection is established, the client-server is no longer mandatory, both the client application and the server application are allowed to take the initiative of sending data, **a request**. The counterpart application must be always available to receive a request, process it, and then send **a response** to the received request.
- Every request (sent by the client or the server) has a mandatory response (correspondingly sent by the server or the client), both requests and responses share a same general message format described ahead.
- Once established, the TCP connection between the client and the server is kept alive and is used for all required data exchanges while the client application is running.

1.2. SBP message format

Every data exchange through the TCP connection (requests and responses) must comply with the bytes sequence description in Table 1, this is the message format version one. This message format is not expected to change during the SBS development.

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

1.3. SBP message codes

Table 2 contains a list of some fundamental message codes that must be implemented by every application using SBP, these five message codes are accepted without preceding authentication.

CODE	Meaning
0	**COMMTEST** – Communications test **request** with no other effect on the counterpart application than the response with a code two message (ACK). This request has no data.
1	**DISCONN** – End of session **request**. The counterpart application is supposed to respond with a code two message, afterwards both applications are expected to close the session (TCP connection). This request has no data.
2	**ACK** – Generic acknowledgment and success **response** message. Used in response to successful requests. This response contains no data.
3	**ERR** – Error **response** message. Used in response to unsuccessful requests that caused an error. This response message may carry a human readable phrase explaining the error. If used, the phrase explaining is carried in the DATA field as string of ASICII codes, it's not required to be null terminated.
4	**AUTH** – User authentication **request**. Described ahead.

Table 2- SBP message codes

The project development teams will along the sprint C of the Integrative Project of the 4th Semester, establish additional unique message codes as needed to implement new features. Notice that, as long as the message format is the same, adding new message codes doesn't change the message format version.

2. User authentication

Once the TCP connection between the ***Shared Board App*** application (client) and the ***Shared Board Server*** application (server) is established, the ***Shared Board App*** is forced to authenticate the local user by sending an AUTH request. Prior to successful AUTH, the server must ignore any request from the client with a code value above four and send back an ERR message as response.

The user authentication is achieved by a username and password pair, both will be provided to the client application (***Shared Board App***) by the local user running it.

The username and the password values are incorporated in the AUTH request at the DATA field as two null terminated strings of ASICII codes, first the username, followed by the password.

The response to an AUTH request may be an ACK, meaning the authentication was successful, or an ERR, meaning it has failed. In the latter case, additional AUTH requests could be tried by the client.

By the end of sprint B of the Integrative Project of the 4th Semester, these five fundamental message codes are expected to be implemented and fully operational in the ***Shared Board App*** application (client) and the ***Shared Board Server*** application (server).