Homework 4

In the lecture we introduced superdense coding. The goal of this exercise is to better understand the protocol by implementing it as link layer protocol in QuNetSim.

QuNetSim implementation of superdense coding

Necessary reading: Chapters 1.2 to 4.1
Your task is to write two protocols, one for the sender and one for the receiver, such that sender and receiver generate entanglement whenever there is no data to transmit and, whenever there is data to transmit and they have entanglement stored, they use the stored entanglement to for dense coding.

Note: QuNetSim has methods built in for superdense coding, EPR creation and message transmission, but the point is to learn how to write protocols for yourself. You can use the built in versions to verify your code, but explicit implementation of the protocols is expected here. You are only allowed to transmit qubits.

In this assignment is a secret message encoded in a long binary string. The goal is to transmit this message using the link-layer protocol. In the code is a function called `get_next_message`. Calling this function repeatedly will result in one of three things: 1) A binary string of length `DATA_FRAME` will be returned; 2) The integer -1 will be returned; or 3) The boolean `False` will be returned. 1) Occurs with probability p, representing a data traffic scenario where in some cases, there are no messages to transmit. Conversely, 2) happens with probability $1 - p$. 3) Occurs when the data connection is terminated and no more data will arrive (and the protocol can stop).

The protocol works as follows. The sender makes a request for data using `get_next_message`. The sender then prepares a *header qubit* which encodes the type of data that will follow. If 2) occurs, then the sender will encode the header qubit with the value 1 and transmit it. Following, the sender will generate `EPR_FRAME` number of EPR pairs, storing half of them locally. If 1) Occurs, then 0 is encoded in the header qubit. The header qubit is transmitted followed by data. Here there are two choices to make. If there is stored entanglement, then the sender uses superdense coding to transmit as much data as possible. When the entanglement runs out (or there was none to begin with), the sender uses standard qubit encoding to send the rest of the data. When 3) occurs, the sender simply terminates.

In the attached file, we have provides starter code in which there are TODO notes and hints which will guide you where to write code to implement the protocol. Successful implementations will decode the message using these methods. Once the entire binary string is collected, the function `decode_secret_message` will trigger and a readable message will indicate that the homework was completed correctly.

Once the message can be successfully decoded, modify `p` to values $1, 0.99, 0.98, 0.97, 0.5, 0$ and plot the number of times you were able to transmit a word with entanglement versus the number of times it was transmitted without.