

Cyolo Core Engineer Home Assignment

Hello and congrats on receiving Cyolo's home assignment for core engineer!

We are building a framework for transporting messages of several protocols that were designed on top of TCP - over UDP.

We already designed what we call an envelope that enriches packets with info needed to resolve the whole frame. Following is the envelope format:

- **Bytes 0-3:** Frame ID - All packets of the same frame share the same frame ID.
- **Bytes 4-5:** Total packets - The total number of the packets that this frame consists of.
- **Bytes 6-7:** Packer number - The serial number of the packet within the frame. The first packet of the frame will be numbered 0, the second - 1, and so on...
- **Bytes 8-9:** Payload length - The length of this packet payload.
- **Bytes 10-n:** Payload

Your part is to create a function - `func(net.PacketConn) io.Reader` - that joins packets into protocol frames with the following constraints:

- The order of the frames that are read from the `net.Conn` must remain the same as the order of the first packet of each frame. Meaning that if the first packet of frame A came before the first packet of frame B, then frame A must be sent before frame B even if the last packet of frame B came before the last packet of frame A.
- The returned reader's `Read` implementation must be blocking and only return `EOF` if the provided `net.PacketConn` returned an `EOF`

We expect the implementation to be simple and readable as possible for better maintainability.

Good Luck!