Sender Module

This is the main module for the Sender Module and Class

class `sender_rdt.`**`Sender`**`(`*`soc, ip, port`*`)`

Bases: **`object`**

Sender, a class with defined behavior to send data to a receiver

Attributes:

packets: Array of 3 object arrays containing: [formed byte packet, boolean ack, Timeout retransmission thread]

soc: socket that sender uses to send data over ip: ip address to send data to port: port number to send data to base_seq: the lowest sequence number to index by

`arrange_pkts``(`*`data`*`)`

Given chunks of data, populate each entry of Sender packets with packet, False (for acknowledgement), thread.Timer for timeout and retransmit

Parameters: **data** (*Array of Strings*) – array of chunks of data

`find_recv_base_window``(`*`window_size`*`)`

Given window size and Sender packets, find the closest unacknowledged packet and calculate the window

Parameters: **window_size** (*int*) – size of window

`packets` = *None*

`run_sender``()`

This function assumes Sender packets to be populated, through arrange_packets. Sends packets in a Selective Repeat fashion

`send_pkt``(`*`seq_num`*`)`

Retransmits packet after timeout by thread.Timer and resets timeout

Parameters: **seq_num** (*int*) – sequence number to retransmit

`sender_rdt.`**`convert_receiver_payload`**`(`*`data`*`)`

Decodes packet payload to retrieve sequence number and message of packet

Parameters:	**data** (*Bytes*) – sequence of Bytes to decode
Returns:	send_seq, sequence number of packet
Return type:	Bytes
Returns:	msg, data from packet
Return type:	String

`sender_rdt.`**`make_checksum`**`(`*`data`*`)`

Forms checksum from data using crc32 function from zlib library

Parameters:	**data** (*Bytes*) – sequence of Bytes to calculate checksum
Returns:	checksum of data
Return type:	Bytes

`sender_rdt.`**`make_packet`**`(`*`seq_num, msg`*`)`

Forms packet by combining calculated checksum and formed payload

Parameters:	• **seq_num** (*int*) – int to convert to bytes • **msg** (*String*) – characters to encode
Returns:	payload, sequence of bytes containing seq_num and msg
Return type:	Bytes

`sender_rdt.`**`make_sender_payload`**`(`*`seq_num, msg`*`)`

Forms packet payload by encoding sequence number and message of packet

Parameters:
- **seq_num** (*int*) – int to convert to bytes
- **msg** (*String*) – characters to encode

Returns: payload, sequence of bytes containing seq_num and msg

Return type: Bytes

`sender_rdt.`**`verify_integrity`**`(`*`sent_chksum, data`*`)`

Verifies checksum from received packet

Parameters:
- **sent_chksum** (*Bytes*) – received checksum with length of 8 bytes
- **data** (*Bytes*) – sequence of bytes to calculate checksum with

Returns: if sent_chksum is the exact same as calculated checksum

Return type: Boolean