

Receiver Module

This is the main module for the Receiver Module and Class

```
class receiver_rdt.Receiver(soc)
```

Bases: **object**

Receiver, a class with defined behavior to receive data from a sender

Attributes:

packets: Array of received decoded data

soc: socket that receiver uses to bind and receive data over ip: ip address to receive data from port:

port number to receive data from base_seq: the lowest sequence number to index by max_seq: the

highest sequence number known to the receiver

```
add_packet(seq_num, data, expand_pkts)
```

Given seq_num, data add data to Receiver packets, if expand_pkts is True, seq_num is bigger than self.max_seq. In this event, add entries until seq_num is reached and input data

Parameters:

- **seq_num** (*int*) – sequence number of data packet
- **data** (*String*) – decoded data
- **expand_pkts** (*Boolean*) – true if seq_num >= self.max_seq

```
base_seq = -1
```

```
clear_packets()
```

Clears Receiver object packets to emptiness

```
get_packets()
```

Retrieves Receiver object packets

Returns: self.packets

max_seq = -1

packets = []

rebase_packets(seq_num, data)

Given seq_num, data add data to Receiver packets, this function is called if seq_num is smaller than self.base_seq where self.packets is modified to make seq_num the new self.base_seq and populate decoded data in self.packets

Parameters:

- **seq_num** (*int*) – sequence number of data packet
- **data** (*String*) – decoded data

run_receiver()

Waits for data from self.soc, verifies data and populates data in self.packets using class methods.
Exits 15 seconds of no activity after sender/client sends a sequence number of -1 is sent

receiver_rdt.convert_sender_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

receiver_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

`receiver_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

`receiver_rdt.make_receiver_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

`receiver_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