

COMP-9311

Tian Huang

Nov 17th, 2025

URP Protocol Report

1. Organization

I choose Python as my programming language for this assignment. There are 2 Python files:

Sender.py

 URP sender, including PLC

Receiver.py

 URP receiver

Both sender and receiver includes protocol state machine and logging statics tracking

2. General Design

State Machine:

Sender:

 CLOSED → SYN_SENT → ESTABLISHED → FIN_SENT →
 CLOSED

Receiver:

 CLOSED → LISTEN → ESTABLISHED → TIME_WAIT →
 CLOSED

Key Feature:

Single Timer

Sliding Window, using a dictionary to buffer unconfirmed segments

No order buffering

3. Error Handeling:

A 16 bit long sum check, similar to TCP:

Sum all 16 bit words in segment

Fold carry bits

Take one's complement

4. Data Structure

Sender:

Sliding Window:

window: {seq_num: (segment, payload_len, send_time)}

unacked_bytes

base

next_seq

Fast Retransmission:

Dup_ack_count

Last_ack

Timer:

Oldest_unacked_seq

Timer_running

Receiver:

Buffer

Received_bytes

Expected_seq

5. How it works

Sender:

Establish Connection

Generate random ISN

Send SYN

Start timer

Wait for ACK

If valid ACK, then ESTABLISHED, if timeout, retransmit

Data Transfer:

Windows Management, ACK validation, fast retransmission and single timer in TCP style

Connection Termination:

FIN → start timer → wait for ACK → retransmit if timeout/exit if valid ACK

PLC:

All outgoing segments go pass PLC, all incoming ACK go pass reverse PLC. URP remains unaware of PLC

Receiver:

Connection:

Receiving SYN → validate → send ACK → ESTABLISHED

Data:

Discard corrupted segment

Write in order

Buffer out of order segments

Send ACK immediately

Termination"

Get FOM → send ACK → wait 2 seconds → CLOSED