

# COMP3331 Assignment Report

Andrew Phuoc Nguyen (z5162792)

October 20, 2018

## 1 Report Questions

### 1.1

*A brief discussion of how you have implemented the STP protocol. Provide a list of features that you have successfully implemented. In case you have not been able to get certain features of STP working, you should also mention that in your report.*

The STP protocol is implemented in the Sender and Receiver class in the logic of its methods and as the actual packet in the STP class. Supporting class Logger is contained in receiver and sender and its sole purpose is to record the logs and collect data for the final summary. The PLD class affects the transmission of packets to simulate real life transmissions.

The features that work are:

- Basic ACK and SEQ number usage
- Send file
- Receive file
- PLD module
- Resend previous ACKs upon unexpected ACKs (not fully working)
- Error correction (partial)
- Timeout
- Fast Retransmit
- Setup
- Set-up
- Teardow
- Checksum
- Buffer out of order packets (not fully working)
- Log (final summary not fully reliable)

Unimplemented features:

- Calculating RTT (sample and dev)

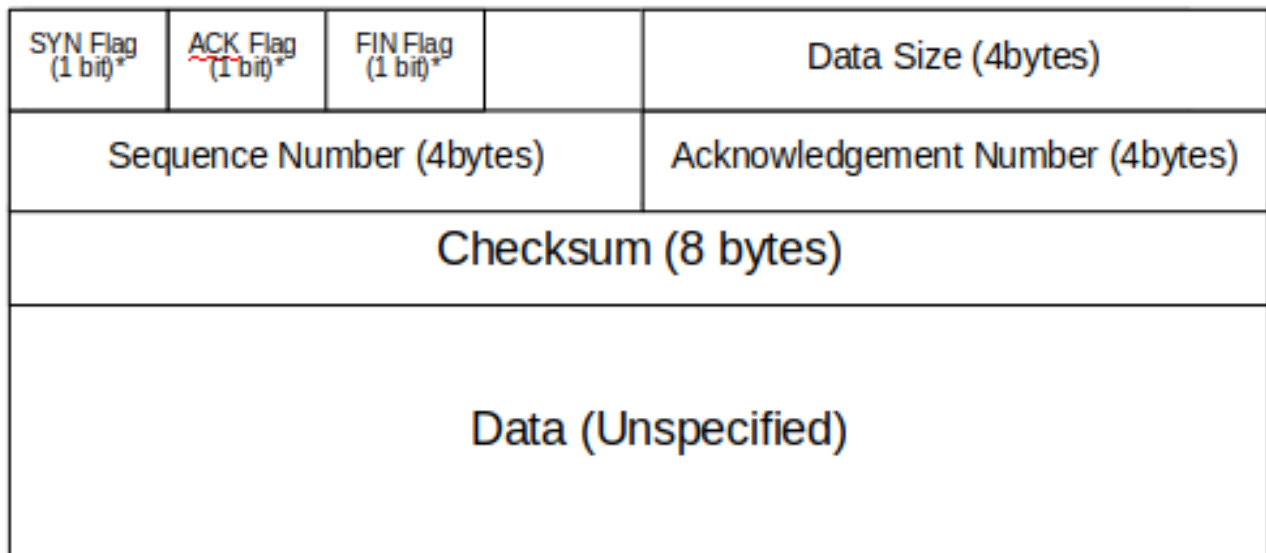


Figure 1: STP Diagram.

## 1.2

*A detailed diagram of your STP header and a quick explanation of all fields (similar to the diagrams that we have used in the lectures to understand TCP/UDP headers)*

\* Note that boolean is not defined in Java and thus its size is dependent on the JVM used. All other values are based on Java primitives.

SYN Flag - Notifies receiver that the sender wishes to initiate a file transfer and to form a connection.

ACK Flag - Notifies the receiver that the packet contains no data and signifies an acknowledgement of receiving data.

FIN Flag - Notifies receiver that the sender has completed their operation and that they have completed their portion of the teardown process and they the other may proceed with their teardown.

Data Size - The number of bytes of data sent in the packet.

Sequence Number - First byte of a segment.

Acknowledgement Number - Notified receiver that the data has been received up to the ACK number.

Checksum - A numerical hash of the rest of the packet using CRC32 used to identify any changes in the packet during transmission.

Data - Data being transmitted.

## 1.3

*Discuss any design trade-offs considered and made. Describe possible improvements and extensions to your program and indicate how you could realise them.*

Trade-offs made is in the use of Java as the language to implemenet the protocol as the object-oriented language is relatively slower and inefficient compared to lower level languages such a C. However it made creating STP packets much easier due to classes which allowed for the grouping and scoping of related information. Additionally there is a lot of calculations done on the receiver and sender which could have been offset on the packet to make the server-side

calculations shorter such as the relative segment of the packet being sent, mws, mss, and total file size. However this reduces the number of repetitive data being sent.

Improvements, apart from implementing the incomplete features, include congestion control which would have increased the maximum window size as the packets succeeded in being sent. This increase could have made sending the file a much faster process as fewer segments are needed with larger MWS. This can be achieved on the sender side by collating successful ACKs (matches expected ACK) and unsuccessful ACKs ( those not matching expected ACK) and either doubling, increasing or returning to a MWS of 1, according to the congestion control protocol used.

## 1.4

*Indicate any segments of code that you have borrowed from the Web or other books.*

No code from web sources were lifted directly.

## 1.5

*Questions.*