



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# CS2031 Telecommunication II

## Assignment #1: Publish-Subscribe

Davy Nolan, 17330208  
December 2018

### Contents

1	Introduction.....	2
	1.1 Assignment Requirements.....	2
2	Design and Implementation.....	3
	2.1 Explanation of Program.....	5
	2.2 Snapshots.....	6
3	Advantages and Disadvantages.....	7
4	Reflection.....	7

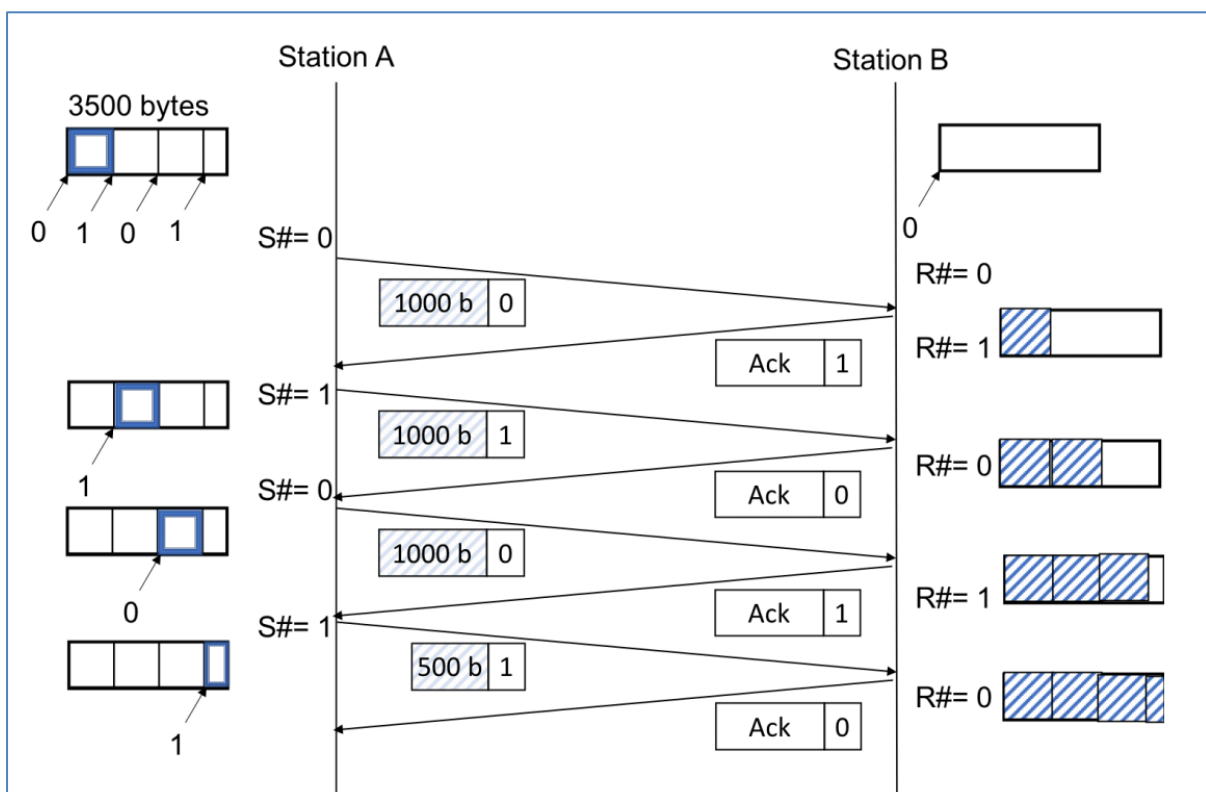
# Introduction

## *Assignment Requirements:*

- ⇒ To design a protocol that forwards messages from a **publisher** to a **broker** which in turn distributes these messages to a number of **subscribers**.
- ⇒ Subscribers should **accept a topic as input**, **send a subscription message** to a broker and **print published messages** that have been forwarded by a broker. They may also choose to **unsubscribe from topics**.
- ⇒ Publishers should **accept a topic and a message as input** and **transmit a message** with the topic and message to a broker. Messages from publishers should have a **sequence number**.
- ⇒ A broker should **receive subscriptions**, should **maintain lists of subscribers** to various topics and should **forward incoming publish-messages** to the subscribers with matching topics.
- ⇒ Subscribers should **print messages in the order of sequence numbers**.
- ⇒ The broker and the subscribers **may implement acknowledgements** and the publisher may wait for an acknowledgement from a broker before proceeding to accept input of another topic and message.

# Design and Implementation

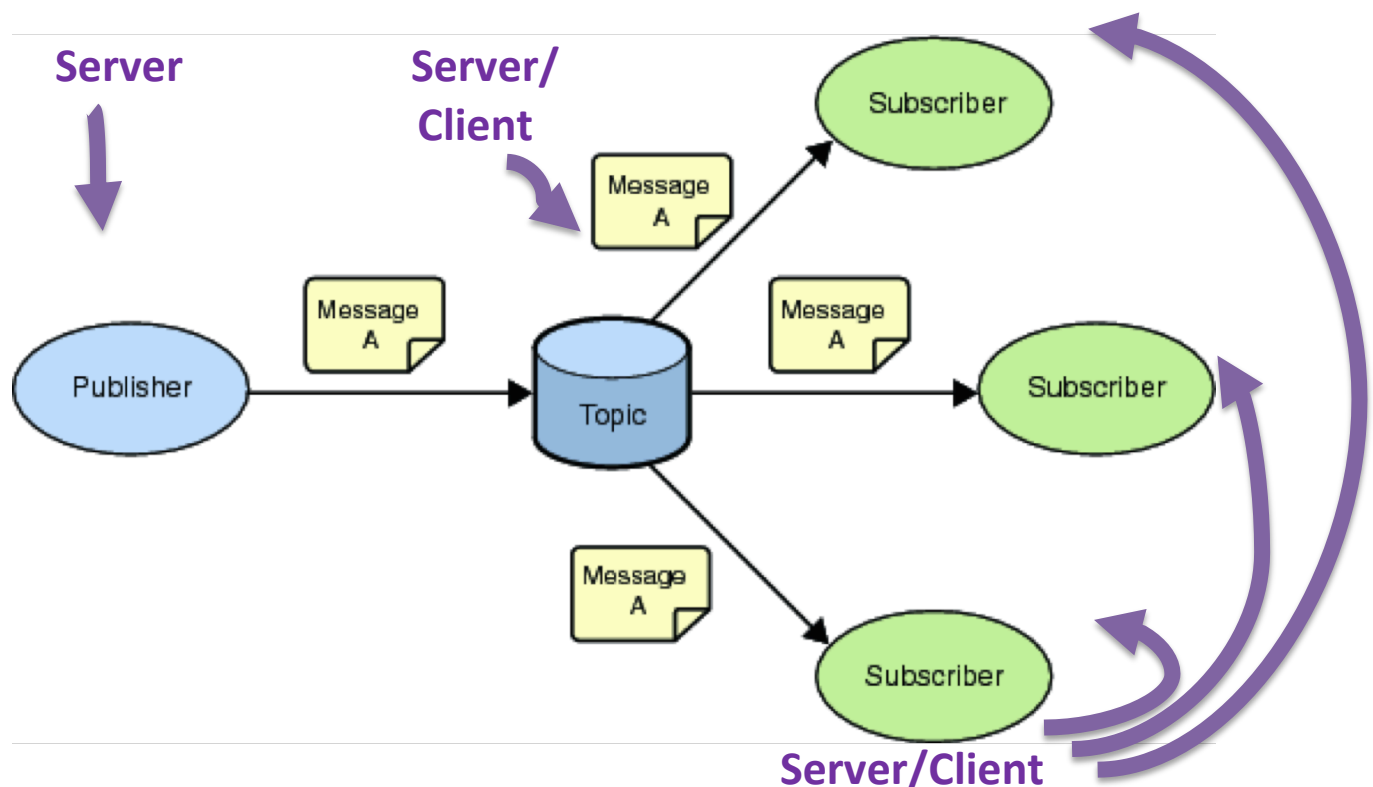
The design consists of 3 stations; Publisher, Broker and Subscriber. After the publisher sends data to the broker, the publisher must wait until the broker sends an acknowledgement to send another. This is also the case with the broker and subscriber. A Stop&Wait ARQ (shown below) could be implemented for the assignment requirements.



In this assignment, an acknowledgement is equivalent to a string containing "OK", so once a socket receives a package, it will send back "OK" to let the sender know that their message has been received (Similar to the use of the term "Delivered" and "Seen" in modern day messaging applications).

The provided classes: Server, Client, Node, PackageContent, StringContent and the tcdlib.jar file.

The subscriber can be implemented as a Client, the broker can be implemented as both Client and Server. This is because the broker is a client to the publisher but a server to the subscriber. Lastly, the publisher can be implemented as a Server.



## *Explanation of Program*

### **Publisher Class:**

- ⇒ The **constructor** accepts Terminal terminal and int port. Terminal is a class which is a part of tcdIO from the tcdlib.jar file. Port is the port number of the publisher socket.
- ⇒ The **void onReceipt** function assumes that all incoming packages contain a String and prints the String. It also checks to see if the incoming package contains the word “end” and if it does, it ends the program. Upon receiving the package and printing the String, it responds with “OK” acknowledgement.
- ⇒ The **void start** function prints message to the terminal; “Waiting for contact”.

### **Subscriber Class:**

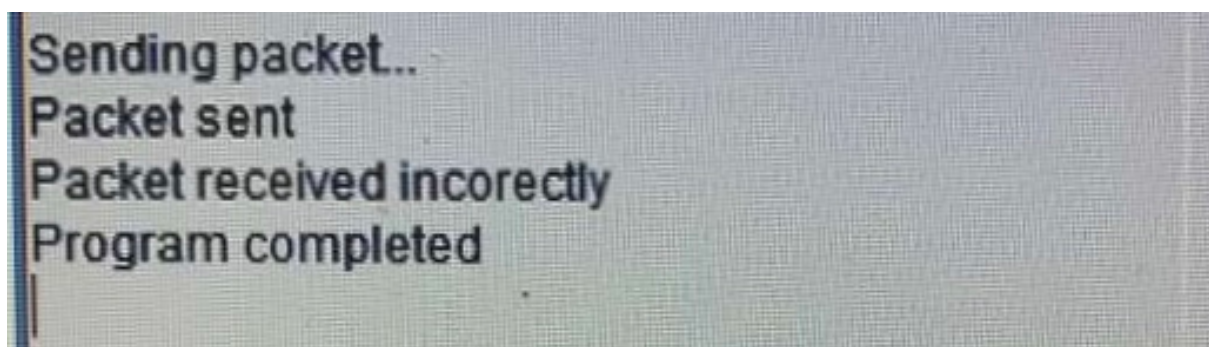
- ⇒ The constructor accepts Terminal terminal, String dstHost, int dstPort ,int srcPort and String topic. The constructor attempts to create a socket at the srcPort and create an InetAddress for the destinations. The topic is what genre the subscriber wants to subscribe to.
- ⇒ The void onReceipt functions assumes all incoming packages contain a String and prints the String.
- ⇒ The void start function sends a packet to the Broker containing a String.

## Broker Class:

- ⇒ The constructor accepts Terminal terminal, String dstHost, int dstPort, int srcPort and String [] topics. The constructor attempts to create a socket at the srcPort and create an InetAddress for the destinations. The topics are the possible topics the subscriber can subscribe to.
- ⇒ The void onReceipt functions assumes all incoming packages contain a String and prints the String. Upon receiving the package and printing the String, it responds with "OK" acknowledgement.
- ⇒ The void start function sends a packet to the Subscriber containing a String.

## *Snapshot of Sent Packets*

### Subscriber Terminal



```
Sending packet...  
Packet sent  
Packet received incorrectly  
Program completed
```

# Advantages & Disadvantages

## *Advantages*

- ⇒ Program utilised tcdIO effectively with the terminal windows.
- ⇒ Program includes required classes; Publisher, Subscriber and Broker.
- ⇒ Program effectively uses the provided Node class.
- ⇒ Program handles errors in each function with Try and Catch cases.

## *Disadvantages*

- ⇒ Program does not implement the ability to unsubscribe from a topic.

## Reflection

Overall I found the assignment challenging. I believe it could have gone better if I had more knowledge of the topic at hand. However, the assignment allowed me to apply both my knowledge of telecommunications and programming in Java effectively and also extended my knowledge in both fields.