

Publish-Subscribe

Sockets & Header Information

Your task is to design a protocol that forwards messages from a publisher to a broker which in turn distributes these messages to a number of subscribers. The aim of the assignment is to get to know sockets, datagram packets and threads and to design a protocol i.e. to decide on packet layout and packet handling, for the communication between a number of nodes and to learn to describe your solution as part of a report.

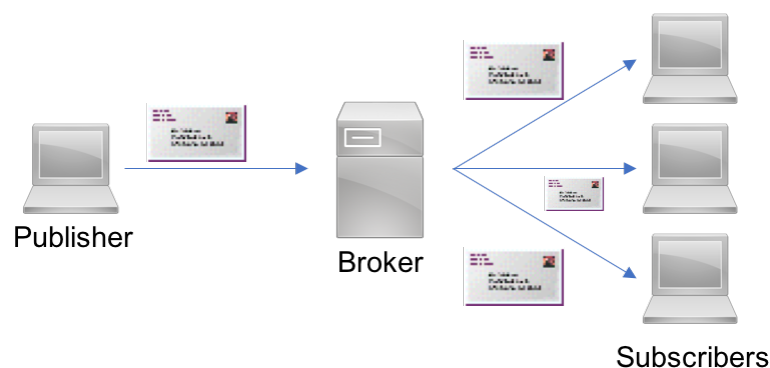


Figure 1: Example topology for a publish-subscribe scenario with a broker, a publisher and a number of subscribers. In a first step, subscribers would inform the broker about their interests in specific topics. Following this, the broker would distribute any messages from a publisher matching the topics of the subscribers.

The implementation of the sample program provides functionality that transmits a packet from a client to a server, containing a String entered by the user. The server responds to incoming packets with a packet containing the String "OK". The Client opens a port on port number 50000, creates a packet from the input provided by the user and sends this packet to another port on the local machine with the port number 50001.

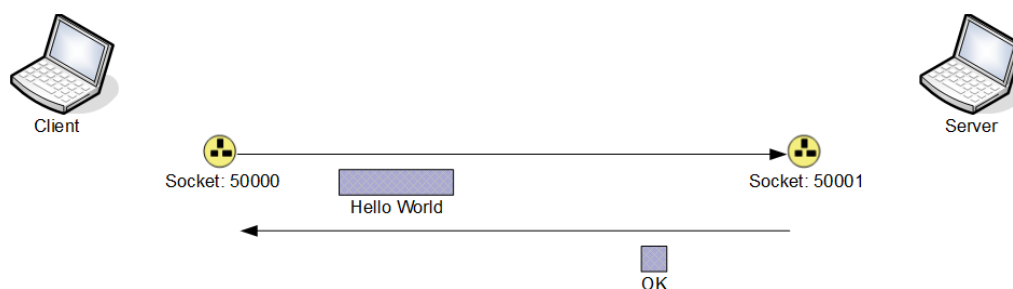


Figure 2: The simplistic implementation of the sample program provides an exchange of two packets between two nodes. The server would start by listening to a given port - 50001 in this example - and the client would transmit a packet addressed to this port. The server responds to the incoming packet with packet addressed to the port the client's message originated from.

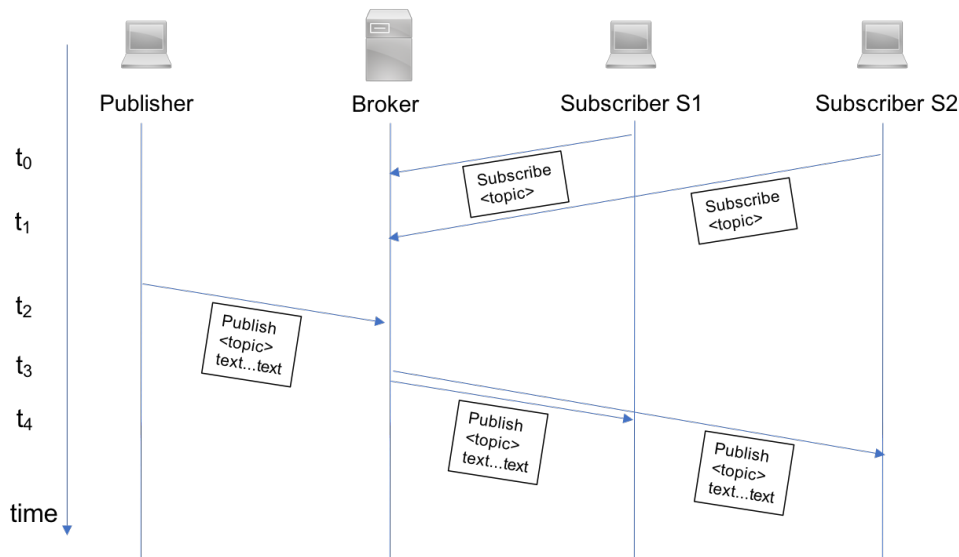


Figure 3: Scenario that your implementation should address

Your implementation should have the following features:

1. Subscribers that accept a topic as input, send a subscription message to a broker and print published messages that have been forwarded by a broker.
2. Publishers that accept a topic and a message as input and transmit a message with the topic and message to a broker.
3. A broker that receives subscriptions, maintains lists of subscribers to various topics and forwards incoming publish-messages to the subscribers with matching topics.
4. Subscribers may choose to unsubscribe from topics.
5. Messages from publishers should have a sequence number and subscribers should print messages in the order of sequence numbers.
6. 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.

As an extension to the simplistic version of one publisher communicating with one or two subscribers, your extended implementation should allow for multiple publishers, multiple brokers and a go-back-N solution for publishers and brokers that allows the retransmission of messages from the publisher if the broker has missed a messages.

The implementation should be accompanied by a report that explains the design and implementation of the protocol, the choices that you have made and the advantages and disadvantages that these decisions introduced. The description of the design should be accompanied by snapshots of some of the packets that were transmitted by your implementation. The explanations of the packets should highlight the management information in these packets and illustrate how this information is used by your implementation. The report should conclude with a reflection on the assignment as a whole, what went well for you and what you could have done better, and an estimation of the time that you spent on the assignment.

Submission Details

The files that contain the implementation and the report should be submitted through Blackboard. Every file should contain the name of the author and the student number. The source files of the implementation should be submitted as an archived file e.g. “.zip” or “.tar.gz”. The report should be submitted as either word- or pdf-document.

The name of the archive file and the report should include the name and the student number of the author; for example, “123456-John-Doe-FlowC-implement.zip” and “123456-John-Doe-FlowC-Report.pdf” where the name of the student is John Doe and the student # is 123456. The deadline for the submission is given in Blackboard.

Marking Scheme

The marks for the implementations will be split 50% for the implementation and 50% for the documentation through the report.