

Command & Control

Sockets & Header Information

Your task is to design a protocol that forwards messages from a Command & Control (C&C) application to a broker which in turn distributes these messages to a number of Workers. The aim of the assignment is to get to know sockets, datagram packets and threads and to design your own 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.

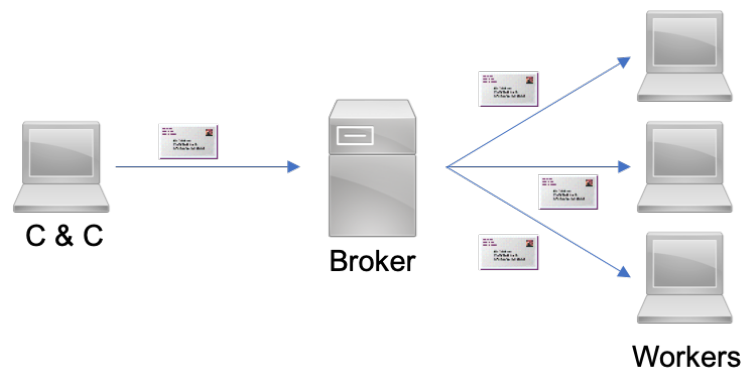


Figure 1: Example topology for a command & control scenario with a broker, a controller and a number of workers. In a first step, workers would inform the broker about their availability for work. Following this, the broker would distribute any messages from a controller to a number of workers.

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.

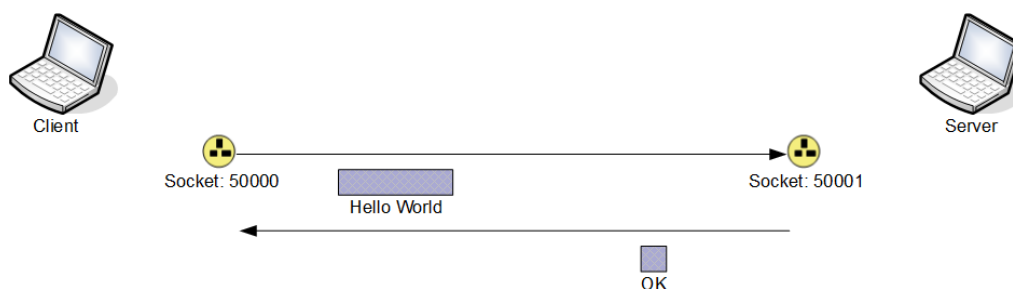


Figure 2: The simplistic implementation of the sample program provides an exchange of two packets between two nodes. The server would start by listening to a given port - 50001 in this example - and the client would transmit a packet addressed to this port. The server responds to the incoming packet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

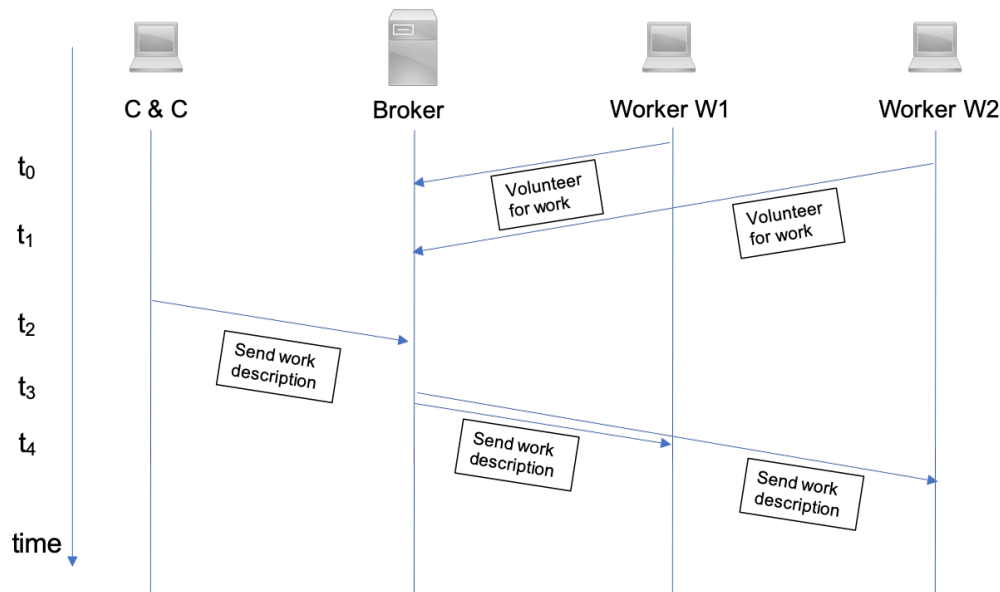


Figure 3: Scenario that your implementation should address

Your implementation should have the following features:

1. Workers that accept a name as input, send a message that they are volunteering for work to a broker and print work assignments that have been forwarded by a broker.
2. C&C server that accepts a work description as input and transmits a message with the work description to a broker.
3. A broker that receives messages from workers, maintains lists of available workers and forwards incoming work descriptions to the workers.
4. Workers may choose to reply to the broker with results from work they carried out or withdraw their availability.
5. Work descriptions from a C&C application may request the same work description to be sent to one, a number of workers or all workers.
6. The broker and the workers may implement acknowledgements and the C&C may wait for acknowledgements from a broker before proceeding to accept input of another work description.

As an extension to the simplistic version of one C&C application communicating with one or two workers, your extended implementation should allow for multiple C&C applications and a go-back-N or selective-repeat solution for all endpoints that allows the retransmission of messages if messages were lost due to faults.

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.