

Use the provided software template to implement a client that talks to a server.

The files provided are:

test-client.c	run this to confirm that your compiler and OS work
test-server.c	run this on your machine to see how test-client works
your-client.c	use this template to write your client
Makefile	(commands to compile/link the above)

The client programs take two optional arguments:

-p portnum	overrides the default server port number to connect to
-s servername	overrides the default server DNS name or IP address

The defaults are:

test-client	55000	localhost (i.e., your own machine)
your-client	55001	dns.postel.org

The test-server has one optional argument:

-p portnum	port number the server listens on
------------	-----------------------------------

The default for test-server is:

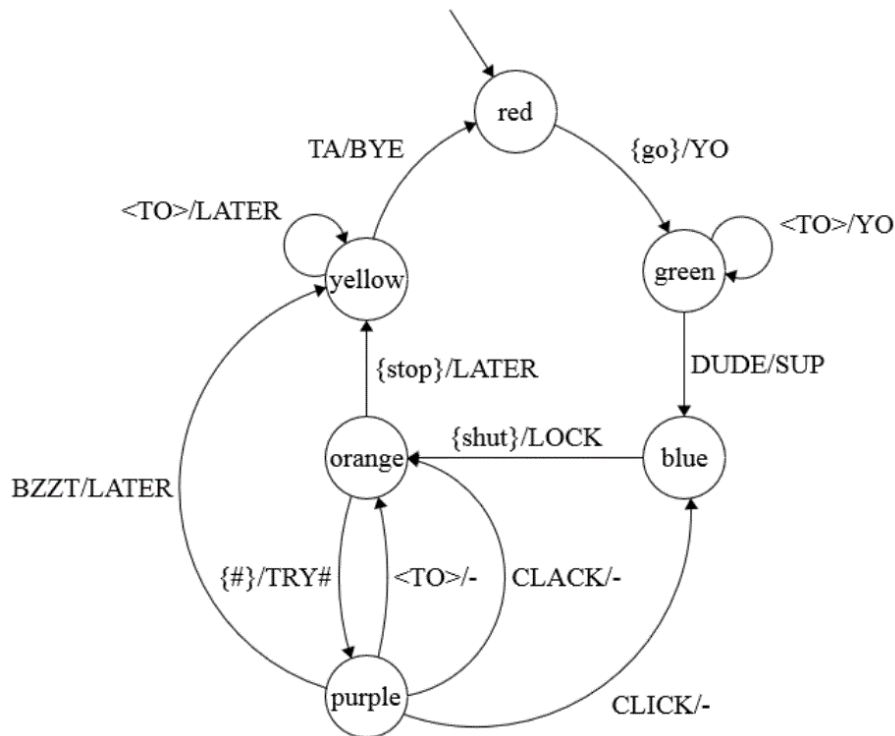
test-server	55000
-------------	-------

Remember:

The server DNS name is just a label identifying the machine.

The server port number is just a label identifying the process within that machine.

1. Implement following client (5 points total)



- Use the software template provided
- Each message is one line, ending in a “\n” character (the provided send/receive calls adds those for you)
- Your machine should start by asking the user (print out a statement, read a response) whether they’re ready to start; when they type “go”, it should begin.
 - o When you’re in the orange state:
 - The user can enter a single-digit number (0-9)
 - The user can enter “stop”
- TRY will include a single-digit number after the symbol “TRY”
 - o You’re basically guessing the correct code to go back to the blue state
 - You can be right (when you receive CLICK)
 - You can be wrong and allowed more tries (you receive CLACK)
 - You can be wrong and be locked out (you receive BZZT)
- Make sure to print out:
 - o Every message received
 - o Every message sent
 - o Every user input
 - o Every state transition
 - o Every timeout