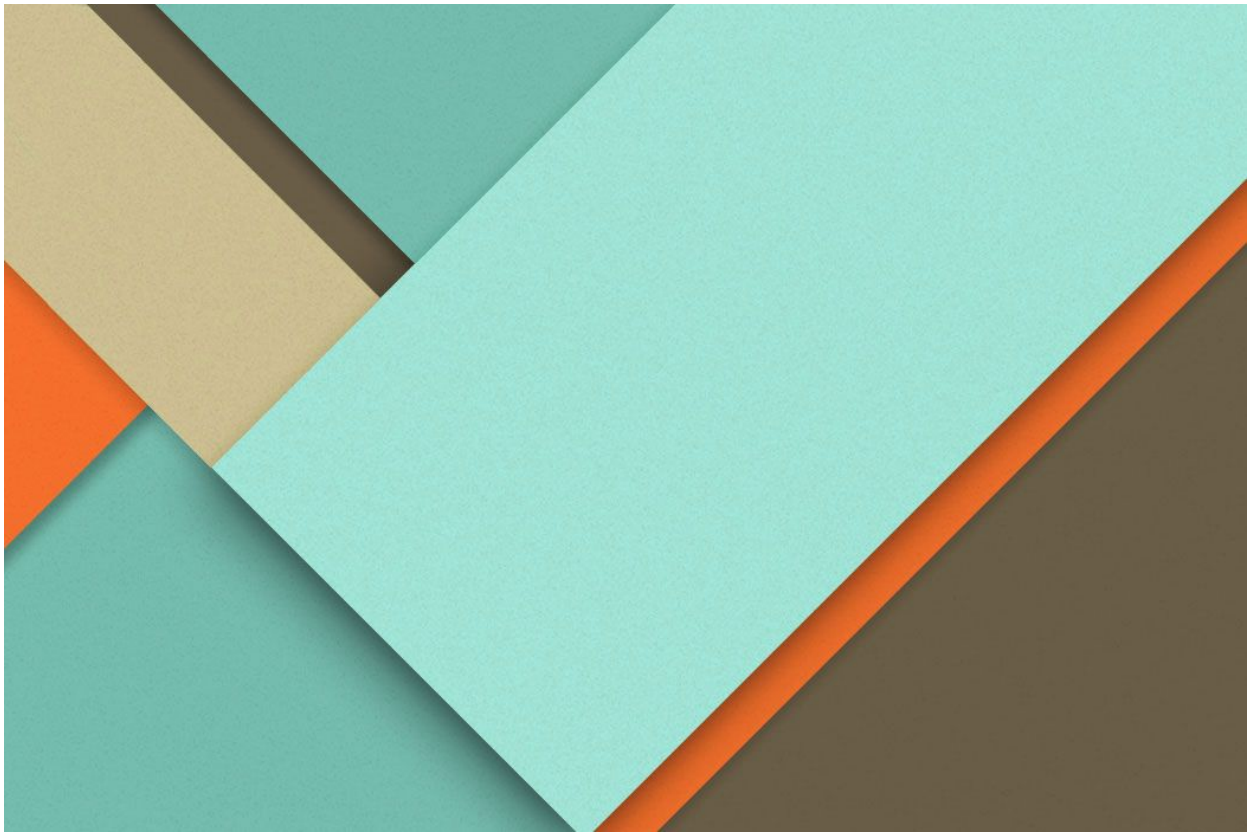


CSN-361 Lecture-3

21.08.2019

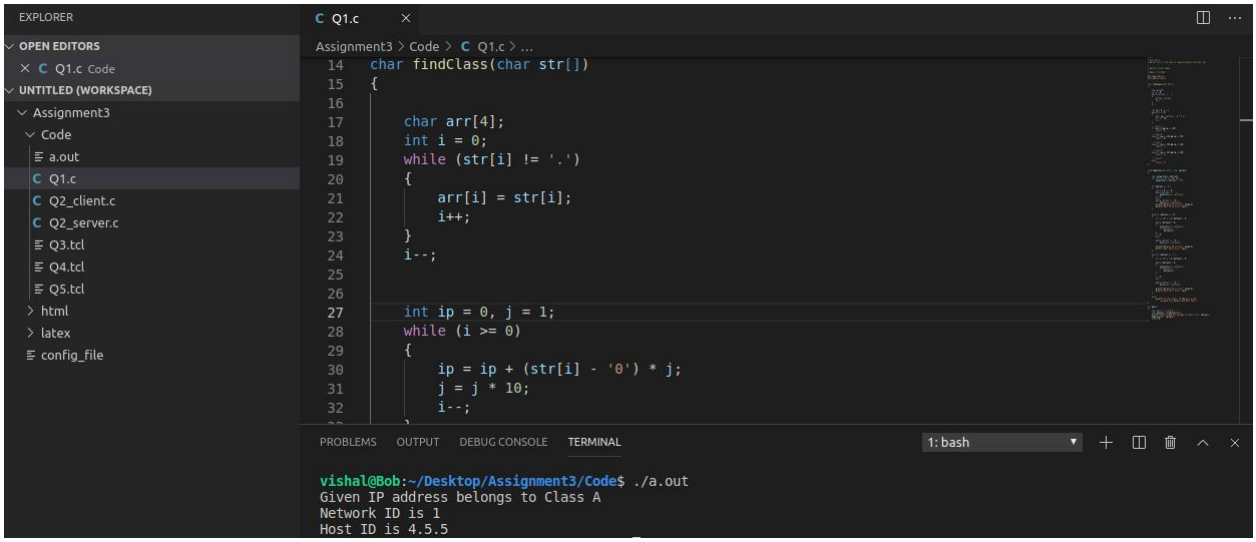
Goddu Vishal
17114034
CS-1

Write a socket program in C to determine class, Network and Host ID of an IPv4 address.

This is done by basic string parsing and value checking.

We look at the first octet (the integer before the first '.' in the IP address) and depending on the range that it lies in, we find out its class (A to E). Then depending on this class, we split the IP address into network and host addresses.

For example: class A -> first octet is network address and the remaining is host address.



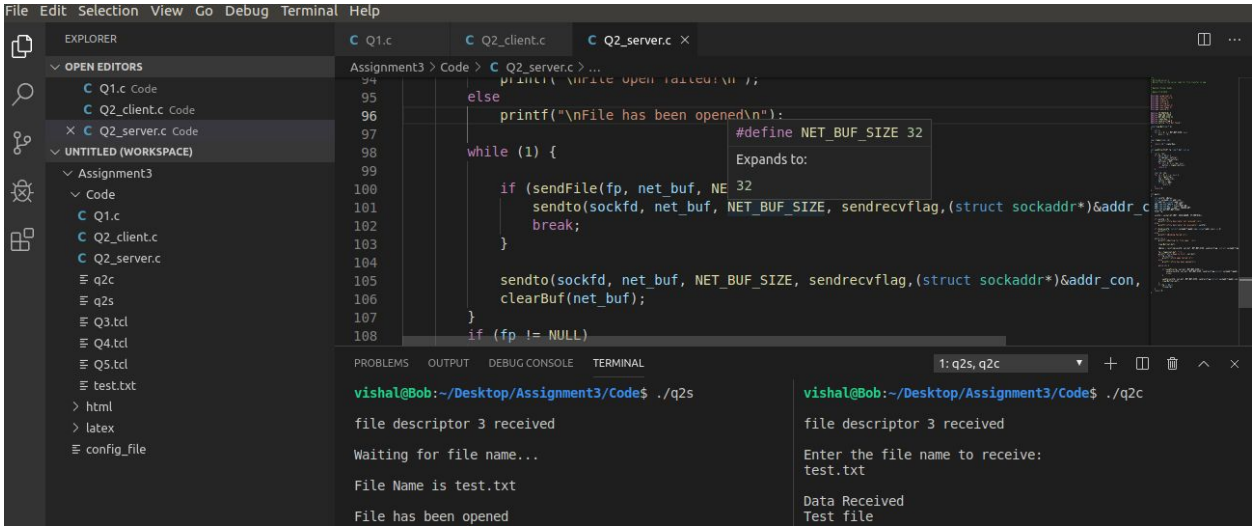
Write a C program to demonstrate File Transfer using UDP.

Similar to previous assignments, struct sockaddr_in is used to use and perform operations on a socket.

The data is ciphered using the XOR operator and using 'S' as the ciphering key -> can be seen in the Cipher() function.

The server first binds to a socket and starts listening for the client. The client checks whether the socket is available, and if available, it sends the name of the file to be read through the buffer. The server receives the file name checks whether file exists or not. If

the file exists it reads the file and sends the data to the client.



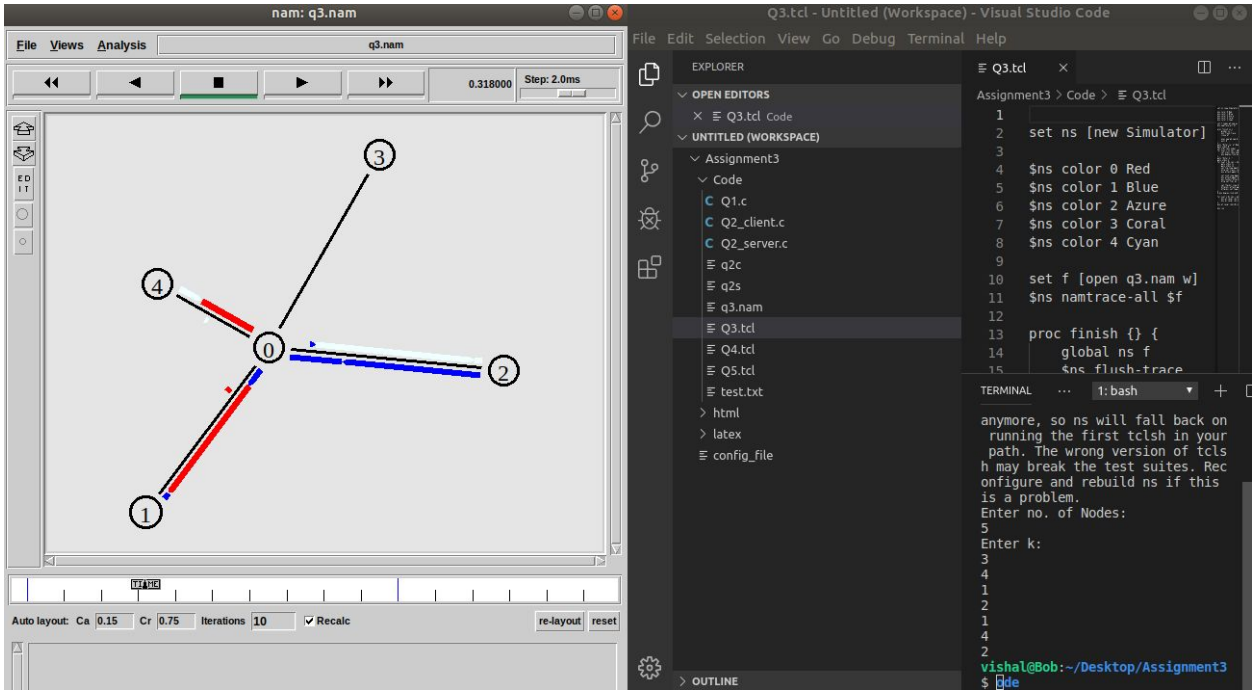
Write a TCL code for network simulator NS2 to demonstrate the STAR TOPOLOGY among a set of computer nodes. Given N nodes, one node will be assigned as the central node and the other nodes will be connected to it to form the star. You have to set up a TCP connection between k pairs of nodes and demonstrate the packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

Since we have to make a star topology, we place the 0th node as the center node and form a duplex-link to every other node. This ensures that all the data passes through node 0.

We create a nam file in our TCLcode to get the simulation.

We take the number of nodes as input and fill the node array.

We take two nodes at the ends from the user and we attach a tcp connection on the sender node and a sink on the receiver node, identically for every connection.

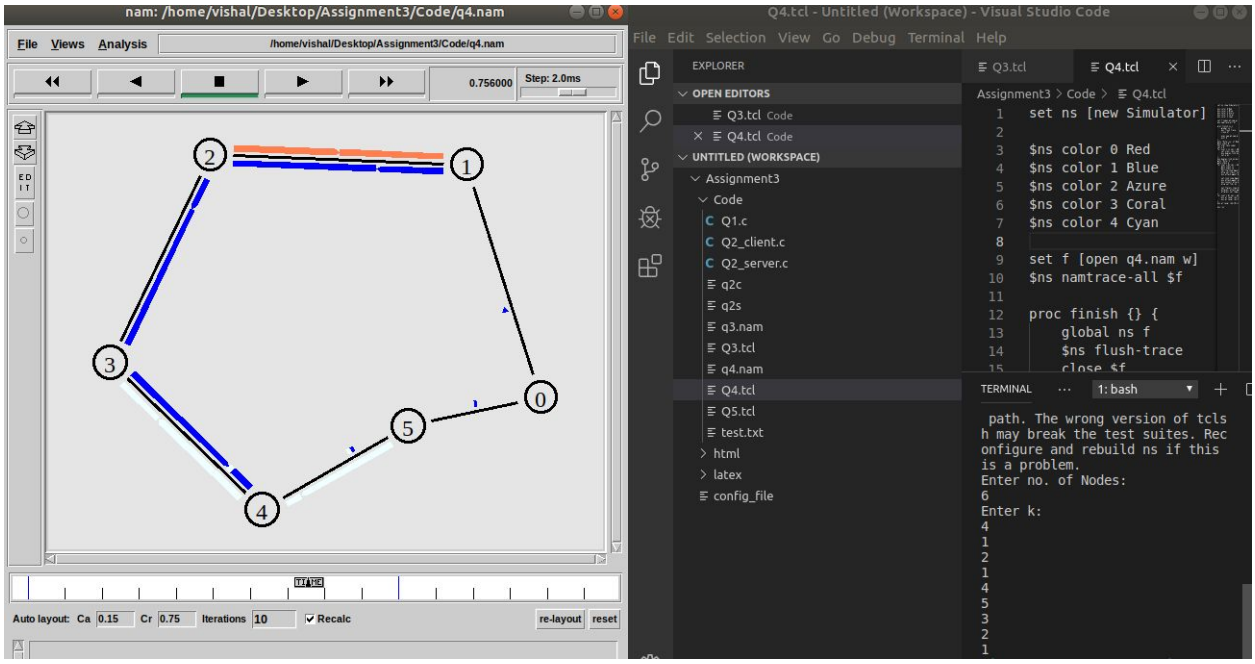


Write a TCL code for network simulator NS2 to demonstrate the RING TOPOLOGY among a set of computer nodes. Given N nodes, each node will be connected to two other nodes in the form of a ring. You have to set up a TCP connection between k pairs of nodes and demonstrate packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

Since we need to make a ring topology. Every node must be connected with it's next as well as previous node with a duplex-link, and the nth node must be joined with the 0th.

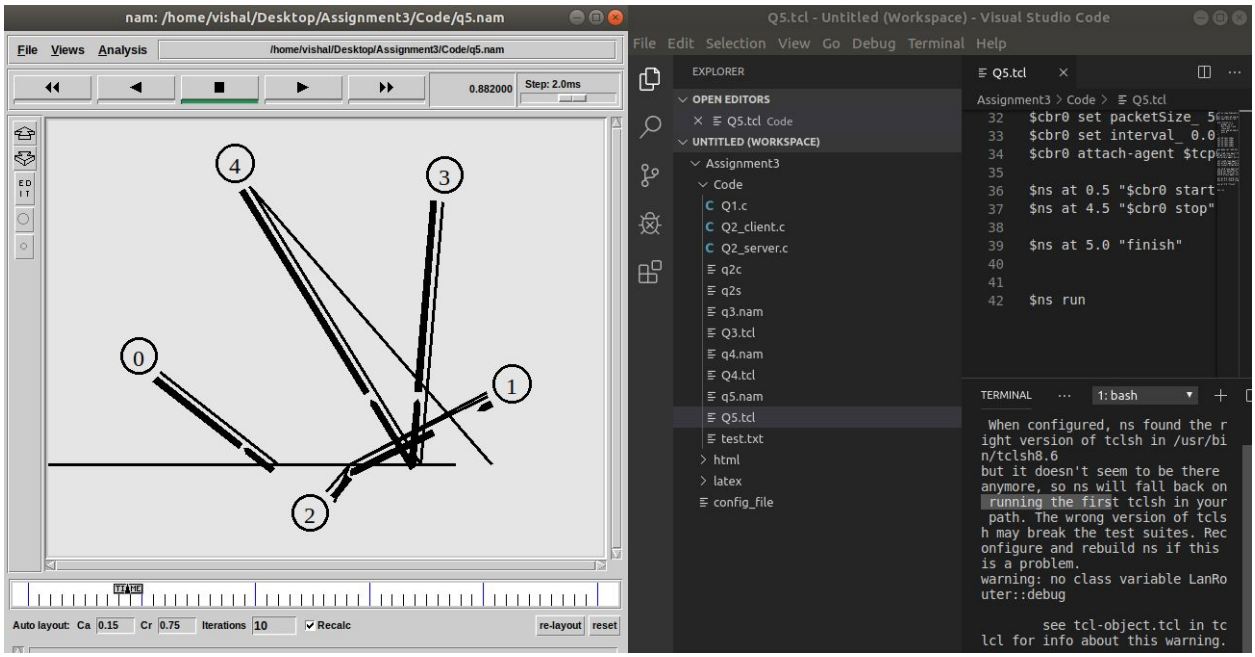
TCL ensures the nearest path of transfer from one node to another as seen in the simulation

The input and variable definitions are the same as the previous question.



Write a TCL code for network simulator NS2 to demonstrate the BUS TOPOLOGY among a set of computer nodes. Given N nodes, each node will be connected to two other nodes in the form of a ring. You have to set up a TCP connection between k pairs of nodes and demonstrate packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

As we need to build a bus topology, we create a lan using the “make-lan” command. All the subsequent nodes are then connected to this lan. The tcp pairing follows the same process as the previous two questions. The input and variable definitions are the same as the previous question.


nam: /home/vishal/Desktop/Assignment3/Code/q5.nam
File Views Analysis
/home/vishal/Desktop/Assignment3/Code/q5.nam
0.882000
Step: 2.0ms
4
3
0
1
2
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc re-layout reset
Q5.tcl - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
EXPLORER
OPEN EDITORS
Q5.tcl Code
UNTITLED (WORKSPACE)
Assignment3
Code
Q1.c
Q2_client.c
Q2_server.c
q2c
q2s
q3.nam
Q3.tcl
q4.nam
Q4.tcl
q5.nam
Q5.tcl
test.txt
html
latex
config_file
Q5.tcl
Assignment3 > Code > Q5.tcl
32 $cbr0 set packetSize_ 5
33 $cbr0 set interval_ 0.0
34 $cbr0 attach-agent $tcp
35
36 $ns at 0.5 "$cbr0 start"
37 $ns at 4.5 "$cbr0 stop"
38
39 $ns at 5.0 "finish"
40
41
42 $ns run
TERMINAL
1: bash
When configured, ns found the right version of tclsh in /usr/bin/tclsh8.6
but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh in your path. The wrong version of tclsh may break the test suites. Reconfigure and rebuild ns if this is a problem.
warning: no class variable LanRouter::debug
see tcl-object.tcl in tclcl for info about this warning.