

CS656 Internetworking and Higher Layer Protocols

Instructor: Grace Wang

Lab Assignment 2

Due date: 11/22/08

Overview

The objective of this assignment is to deepen your understanding of how transport layer and network layer work and how an application message is sent to the destination.

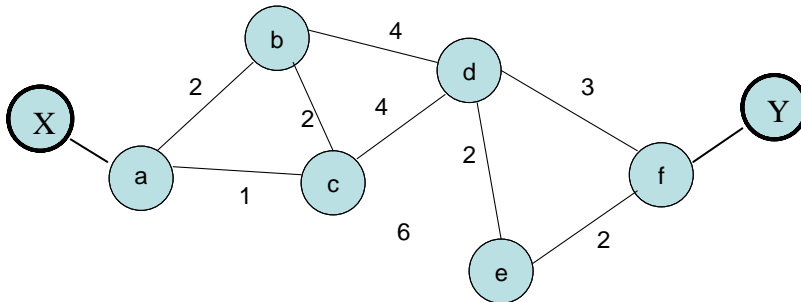
We consider a network with both end hosts and routers. All routers run Distance Vector Algorithm to calculate the forwarding table. In end hosts, a transport layer protocol called uTCP runs. uTCP will segment application layer messages if they are too large. If a message is segmented, it will be re-assembled in a reverse order before being delivered to the application. For example, if message “ABCD” is segmented into “A”, “B”, “C”, “D” by uTCP at the source host, “DCBA” will be delivered to the application by the uTCP at the destination. Here, we assume the maximum data length in a segment is 10Bytes. In addition, in end hosts, a simple application runs. It reads a file called “data.txt” and sends the data in the file to the destination specified by the user. If it receives data, it will save the data into a file called “data-reverse.txt”.

Implementation Details

To ensure every program produce the same output, we take the following assumptions. All the computers in the network run in a loosely synchronized manner. Suppose the whole time line is divided into equal intervals. For routers, they can propagate their DVs (Distance Vector) any time within the current interval if their DVs have been updated. However, by the end of each interval, every node should have received the DVs from their neighbors if there is any. Computation time can be neglected. Routers forward packets only at the end of each interval by consulting the up-to-date forwarding table. After a router sends a packet, the cost of the link is increased by \underline{c} , which certainly incur the updating of DVs.

At end hosts, uTCP only sends one segment every two intervals if there is any. Also, suppose all computers are booted up the same time. End hosts do not send any packet during the first 5 intervals to allow routers calculate their forwarding tables.

In this lab, you are required to use multi-thread programming (you already knew it from the first Lab). You should use threads to simulate routers and end hosts. A thread represents one router or one end host. You should use thread-communication to simulate one-link packet transmission.



Input/Output

The inputs of the program include one network topology file and one data file. The topology file of the above network is as follows, where “#” indicates the destination end host and “##” indicates the source end host. X and Y are end hosts, and a, b,, f are routers.

topology.txt

##	X	a
#	Y	b
a	b	2
a	c	1
b	c	2
b	d	4
d	e	2
c	e	6
d	f	3
e	f	2
c	d	4

The data file, used by the source end host is a plain text file. It can have the following content for example.

AAAAAAAAAABBBBBBBBBBB

The output of the program should be the routes of different packets, shown on the screen and a file called “data-reverse.txt”, generated at the destination host.

data-reverse.txt: BBBBBBBBBBBAAAAAAAAA

On the screen, you should see the routes taken by the packets in the following format:

Packet 1: acdf

Packet 2: acef

Here, packet 1 carries data “AAAAAAAAA” and packet 2 carries data “BBBBBBBBBBB”

Language

You must use either C/C++ or java.

Lab Report

Lab report should be concise and clear. It should include three parts:

1. General information, including the language you used, the names and IDs of your group members.
2. Readme, include how to compile your program and how to execute them.
3. Your protocol specification.
4. Problems encountered and how they are solved. Also, indicate the problems that you have not solved.

Grading

The total credit is 100 points, of which the report accounts for 20 points and the program accounts for 80 points. If the program runs correctly and you answer all the questions correctly in the demo, you will get 80 points; otherwise you will get 0 ~ 50 points, depending on how your program is written. The report is graded based on whether it is clear, concise and well-written and how you discuss the problems you meet.

TA will grade the Lab. He will use two files, also called topology.txt and data.txt, but with different content of the above example. He will check whether the output is the same as calculated.