Tianyi Liang
1436616
Cmput313 Lab3

Objectives

This project is an experience for me to deal with routing algorithms and LAN protocols using CNET. As well, the assignment gives experience with obtaining simulation results suitable for plotting performance graphs for the developed algorithms.

Part1

1. In flooding2.c, what variables are used in implementing a network layer stop-and-wait protocol? What are the initial values of such variables? How does the implementation initialize such variables?

Answer:

seqno & MAXHOPS are used in implementing anetwork layer stop-and-wait protocol.
seqno's initial value = 0. MAXHOPS initial value = 4.
How does the implementation initialize such variables?
Seqno: Use function NL_nextpackettosend. p.seqno = NL_nextpackettosend(p.dest);
MAXHOPS: #define MAXHOPS 4

2. In flooding2.c, does varying MAXHOPS over the range [1, 4] have an effect on the performance of the algorithm? If yes, identify at least two specific quantities that can be used to describe the resulting effect. Each quantity should be observable either from the global network statistics produced by cnet, or the node-specific statistics displayed when a node is clicked (or when any debugging button in a node's window is pressed). Explain your answer.

Answer:

I changed MAXHOPS value from 1 to 4 in flooding2.c. Then, I use cnet -W -q -T -e 5m -s FLOODING2 to see the statistics. I found that the efficiency decreases along with MAXHOPS increases.

When MAXHOPS = 1, Frames transmitted = 2463, Frames received = 2426, Efficiency (bytes AL) / (bytes PL) = 34.24.

When MAXHOPS = 2, Frames transmitted = 2648, Frames received = 2352, Efficiency (bytes AL) / (bytes PL) = 16.80.

When MAXHOPS = 3, Frames transmitted = 2642, Frames received = 2236, Efficiency (bytes AL) / (bytes PL) = 9.83.

When MAXHOPS =4, Frames transmitted = 2994, Frames received = 2306, Efficiency (bytes AL) / (bytes PL) = 7.45.

Therefore, changing MAXHOPS over the range[1,4] have an effect on the performance of the algorithm.

3.In flooding2.c, what information is computed by “ALL_LINKS & ~(1<<arrived_on)”? How does the program use the expression?

Answer: Based on the comments in flooding2.c,”ALL_LINKS & “ ~(1<<arrived_on)” computed all links *except* the one on which it arrived. It returns an int indicates the wanted link.

4. For flooding2.c, comment on the behaviour of the program in the following two situations:

(a) The call to function CNET disable application is commented out, and the program runs on a network with 8 or more nodes for 5 or more minutes.
(b) The unmodified program runs on a network of 20 or more nodes for 5 or more minutes.

Explain the observed behaviour(s).
(a)…
(b)…

5. For flooding3.c, explain how a node processes a relayed frame (i.e., the node is neither the source nor the destination of the frame) in each of the following cases:
(a) The node has not previously received a frame with a matching source address.
(b) The node has previously received a frame with a matching source address.
(c) The node has previously received a frame with a matching destination address.
Answer:

(a) The frame will be stored in minhop
(b) Compare hopcount and minhop
(c) …

Part2

Design Overview:

According to “lab3-lan.c”, this protocol has following features:

- When a transmitted frame collides, the protocol uses an exponential backing off procedure to schedule a retransmission.
- MAX_BACKOFF determines the maximum value of the backoff counter. After this number, a frame is dropped.
- When 'CS_FLAG= 0', the protocol relies only on the backing off procedure without using carrier sensing (i.e., function CNET_carrier_sense is not used). Else, when 'CS_FLAG= 1', the protocoluses both carrier sensing and backing off before transmitting (or re-transmitting) each frame.

Program Status & test:

In this program, I just follow the instruction in the given code and implement them. I just wrote the part inside LANC_manager. I test the program by makefile. And I change cs_flag between 0 and 1 to test two situation.

Result:

For cs_flag == 0:

Network	Global Statistics			Protocol Statistics(for the first node)			
	Messages generated	Messages delivered	Frame collisions	tx_frames	success frames	dropped frames	rx_frames
LAN-5	2868	2747	1361	612	594	18	507
LAN-10	5797	4964	5921	614	539	75	487
LAN-15	8531	6287	11992	551	422	129	421
LAN-20	11283	7111	19150	585	390	195	348

For cs_flag == 1:

Network	Global Statistics			Protocol Statistics(for the first node)			
	Messages generated	Messages delivered	Frame collisions	tx_frames	success frames	dropped frames	rx_frames
LAN-5	2869	2869	0	569	569	0	566
LAN-10	5868	5867	0	561	561	0	577
LAN-15	8777	8776	0	547	547	0	590
LAN-20	11897	11896	0	648	648	0	609

Comment:

- By using cs_flag = 1,comparing with cs_flag = 0, there is a great improvement of the protocol since frame collision = 0.

- When using carrier sensing with backing off(cs_flag = 1), tx_frames = success frames, so the dropped frames is 0.

Acknowledgments:

Given lab3-lan.c, makefile example