COMP 6360 Wireless and Mobile Networks

Project 1 Report

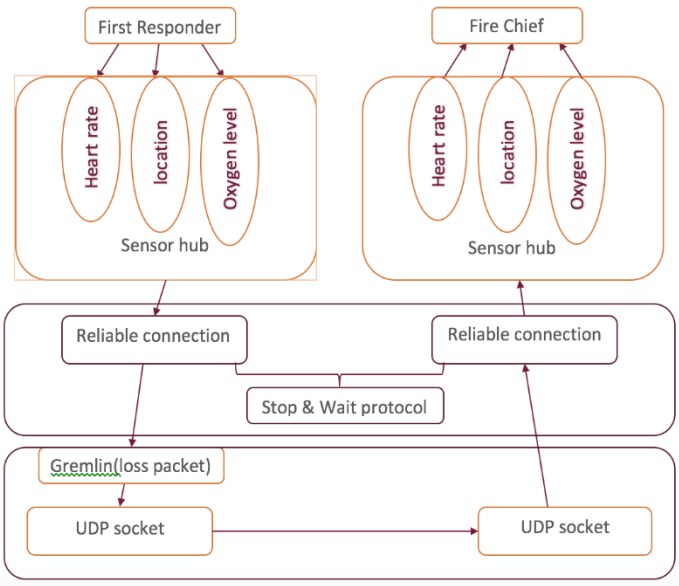
Group 3

Mar. 1 2017

Muzi Li, Chang Ren

1. Architecture

This project consists of Sensor hub program, Reliable Connection and Unreliable Connection with Gremlin Function. Those three functions work collaboratively to simulate the reliable communication between First Responder and Fire Chief through unreliable link as shown below.



2. Design

This project is written in C language. In this project, three time-varying streams of data are packed into a struct named Frame.

```
typedef struct frame{
    int seq_no;
    Packet data;
}Frame;
```

2.1 Sensor hub program

This part mainly discusses that generating three types of data by sensors and transmitting packed data through reliable connection between Fire Chief and First Responder, and showing three types of data on three separate windows.

2.1.1 Data function

To emulate data generation, we program three functions which are heart-rate, location and air-pack oxygen level. All the functions are programmed in client part.

1) Heart-rate function

Heart-rate should relate with oxygen level. When oxygen level become lower, heart-rate increase. And at one point, people will feel sleepy or dizzy or even shock due to the low oxygen level. At this time heart-rate will go down. As the commonsense showing air-pack will last no longer than half an hour, i.e. 1800 seconds. People won't

feel good after oxygen level down to 18%. So the function of heart-rate generates Rate = 0.1*t + 80, where t is time less than 1200 seconds. After t go beyond 1200s, Rate =-1*t/3 +600

2) Location function

We suppose that building is about 6 floors height and the beginning position of first responder is at six floor. After every 60 seconds, he will go down one floor and in each floor he will do the "sin() curve" movement to check every corner of this building to find alive people and save them. We suppose that there is only 30m*30m square place and coordinate system which use x, y, z as it coordinates. When we compile the file, we should add "-lm" after the file name/path.(We should define pi equals to 3.1415926, and add head file<math.h>)

y=30*sin(pi*x/60)

3) Air-pack oxygen level

The oxygen level relates to the heart-rate and time. As commonsense shown in 1) that air-pack can hold at most half an hour, we build a function which from 100% go straight down to 0 in 1800s.

Oxygen-level = -1*t/18 + 100

2.1.2 Data display

According to the requirement, the Fire Chief should monitor three windows on the screen with those three types of data mentioned above. Here we simply use curses library and code in it to draw three display windows. The Curses is a terminal control library for Unix-like systems which can provide API for programmer to write text-based user interfaces. The head file <curses.h> should be added.

First we open curses mode and refresh the screen. Then we set length, width of window and starting position for content. Next using box() to draw the window's boundary. After this, we receive the data and display it in the window. Finally, we refresh window again and delete window. When we compile the file, we should add "-lcurses" after the file name/path

```
WINDOW*win;
initscr();
refresh();
win1=newwin(10,30,1,4);
box(win1,ACS_VLINE,ACS_HLINE);
mvwprintw(win,1,1,"message" ");
wrefresh(win);
delwin(win);
```

2.2 Stop and Wait protocol

The basic idea of stop and wait protocol is that after transmitting one frame with one sequence number, the sender waits for an acknowledge(ACK) with the same sequence number before transmitting the next one. If the sender doesn't receive ACK for previous sent frame after a certain time, the sender times out and retransmits that frame again.

2.3 Gremlin Function

This function provides a way to randomly drop some frames in order to

set up an unreliable link. In the method int germlin(), a random number will be generated to compare with a certain loss rate. Then, the return value will determine if the current frame would be selected to drop.

3.Implementation issue

The main implementation issue is that the terminal window size adjustment would cause system interruption. After compiling and operating the code file in the terminal, the server end will be shut down automatically if we adjust the terminal window size. The reason why this problem occurs still remains unknown and needs time to be figured out.