CSEE W4119 Computer Networks - PA2

Li Yan(ly2278) ly2278@columbia.edu

Go-Back-N Protocol

1 Building intruction

```
cd src
make
```

Then run java program.

2 Document detail

`src/`

- `Accessory.java`
- `Constant.java`
- `GBN.java`
- `gbnnode.java`
- `Receiver.java`
- `Sender.java`
- `SenderTimerTask.java`
- `Transmission.java`
- `makefile`

The the main function of the project is `gbnnode.java`.

3 Project feature

1. Simulate the GBN protocol between two nodes.
2. Print packet sent/received condition.
3. Print ACK sent/received condition.

4 Data structure

To realize GBN protocol, there should be a buffer, a window and a Timer.

4.1 Buffer

If I simply set up a very large buffer, it will be much easy: 1) I don't have to sut the sending data when buffer is not large enough to hold it all; 2) I don't have to consider using buffer as a loop; 3) it will be easier to solve the problem of sending termination.

But here, in order to simulate the real condition, I set the buffer 2*window size. Thereby, it will be hard to use buffer:

1. first cut sending data into small pieces, send one piece at a time and send another when buffer is free.
2. Using mod ('%' in Java) to looply use the buffer.
3. Calculate the end (stop sending) point on the buffer, not on the data string.

The operation on buffer is in `Sender.Buffered_Send` and `Sender.GBN_Send`.

4.2 Timer

I use `java.util.Timer` and `java.util.TimerTask` here. Java Timer is not easy to use. No suspend, no stop, no kill, no restart. What I can do is to cancel it each time a proper ACK is received, and create a new one and create a new schedule for it.

Whenever it time, I set the sending point back to the window point. In this way, the sender will resend packets start at the window point.

5 Algorithm

The main GBN itself does nothing, but runs two threads: 1) Sender; 2) Receiver.

The flowchart for Sender is given in Fig 5-1:

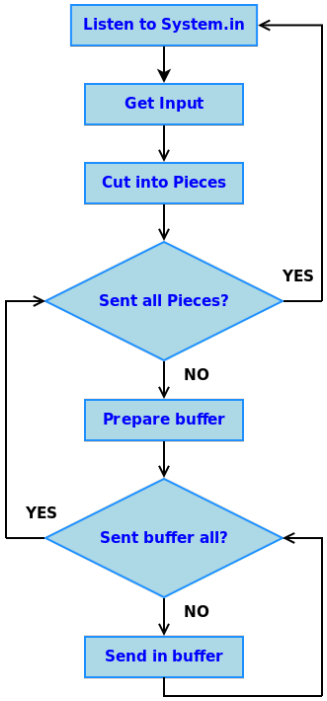

Fig 5-1

The Receiver have the flowchart of Fig 5-2:

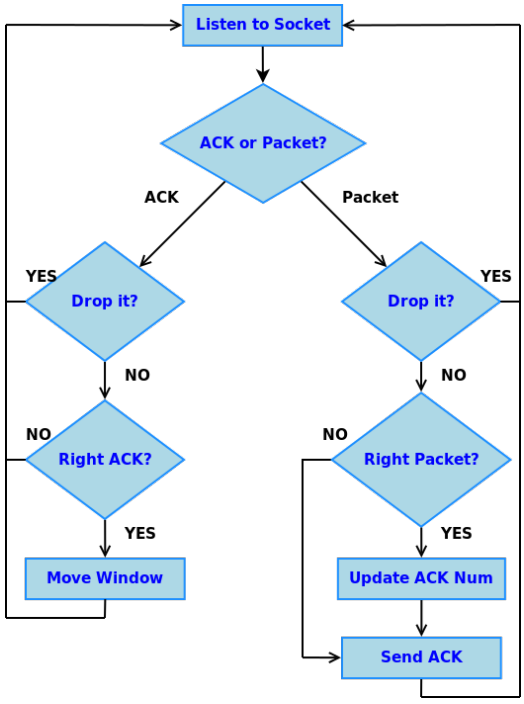

Fig 5-2

6 Usage scenario

With proper run command, each process can perform both sender and receiver. When all the user input has been sent out. It will wait for the user's next input.

7 Test

A test output is in folder "`Test/`", I start up two node:

1. `java gbnnode 1111 2222 5 -p 0.5`
2. `java gbnnode 2222 1111 10 -d 7`

In the test case, both node performs as sender as well as receiver.

Fig 7-1 is the sample of timeout testing, we see it is 500ms (there will be some noise caused by other thread and data exchange in the buffer):

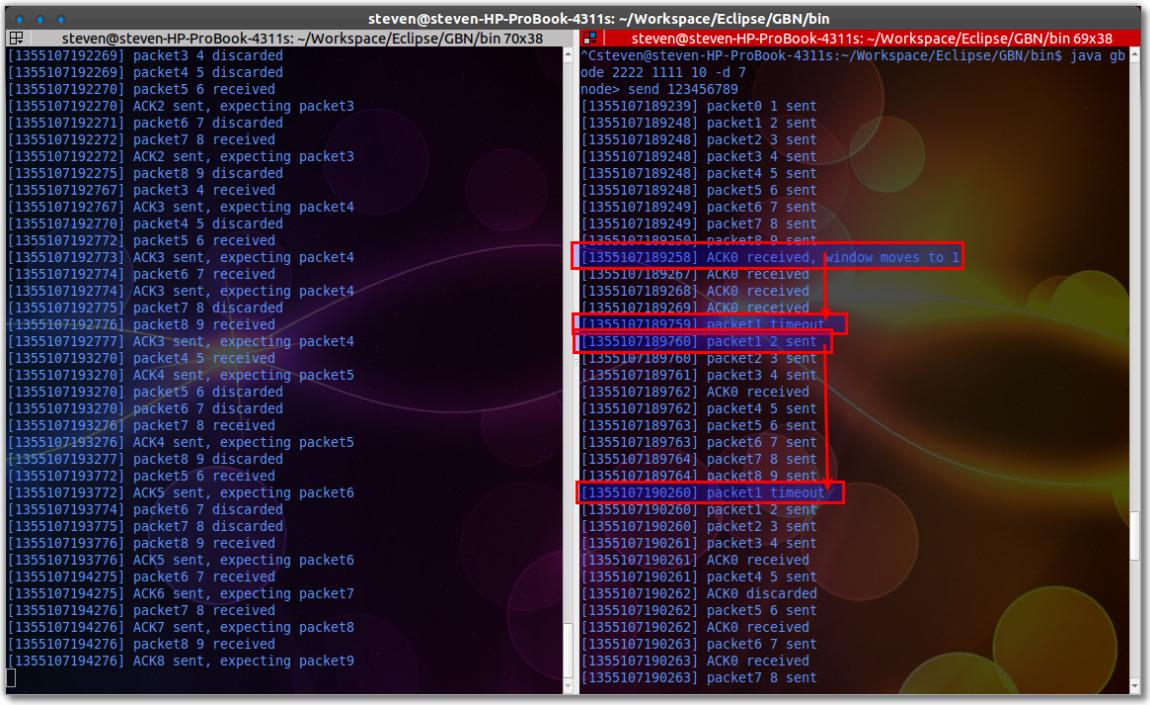
steven@steven-HP-ProBook-4311s: ~/Workspace/Eclipse/GBN/bin
steven@steven-HP-ProBook-4311s: ~/Workspace/Eclipse/GBN/bin 70x38
[1355107192269] packet3 4 discarded
[1355107192269] packet4 5 discarded
[1355107192270] packet5 6 received
[1355107192270] ACK2 sent, expecting packet3
[1355107192271] packet6 7 discarded
[1355107192272] packet7 8 received
[1355107192272] ACK2 sent, expecting packet3
[1355107192275] packet8 9 discarded
[1355107192767] packet3 4 received
[1355107192767] ACK3 sent, expecting packet4
[1355107192770] packet4 5 discarded
[1355107192772] packet5 6 received
[1355107192773] ACK3 sent, expecting packet4
[1355107192774] packet6 7 received
[1355107192774] ACK3 sent, expecting packet4
[1355107192775] packet7 8 discarded
[1355107192776] packet8 9 received
[1355107192777] ACK3 sent, expecting packet4
[1355107193270] packet4 5 received
[1355107193270] ACK4 sent, expecting packet5
[1355107193270] packet5 6 discarded
[1355107193270] packet6 7 discarded
[1355107193276] packet7 8 received
[1355107193276] ACK4 sent, expecting packet5
[1355107193277] packet8 9 discarded
[1355107193772] packet5 6 received
[1355107193772] ACK5 sent, expecting packet6
[1355107193774] packet6 7 discarded
[1355107193775] packet7 8 discarded
[1355107193776] packet8 9 received
[1355107193776] ACK5 sent, expecting packet6
[1355107194275] packet6 7 received
[1355107194275] ACK6 sent, expecting packet7
[1355107194276] packet7 8 received
[1355107194276] ACK7 sent, expecting packet8
[1355107194276] packet8 9 received
[1355107194276] ACK8 sent, expecting packet9
steven@steven-HP-ProBook-4311s: ~/Workspace/Eclipse/GBN/bin 69x38
^Csteven@steven-HP-ProBook-4311s:~/Workspace/Eclipse/GBN/bin$ java gb
ode 2222 1111 10 -d 7
node> send 123456789
[1355107189239] packet0 1 sent
[1355107189248] packet1 2 sent
[1355107189248] packet2 3 sent
[1355107189248] packet3 4 sent
[1355107189248] packet4 5 sent
[1355107189248] packet5 6 sent
[1355107189249] packet6 7 sent
[1355107189249] packet7 8 sent
[1355107189258] ACK0 received, window moves to 1
[1355107189267] ACK0 received
[1355107189268] ACK0 received
[1355107189269] ACK0 received
[1355107189759] packet1 timeout
[1355107189760] packet1 2 sent
[1355107189760] packet2 3 sent
[1355107189761] packet3 4 sent
[1355107189762] ACK0 received
[1355107189762] packet4 5 sent
[1355107189763] packet5 6 sent
[1355107189763] packet6 7 sent
[1355107189764] packet7 8 sent
[1355107189764] packet8 9 sent
[1355107190260] packet1 timeout
[1355107190260] packet1 2 sent
[1355107190260] packet2 3 sent
[1355107190261] packet3 4 sent
[1355107190261] ACK0 received
[1355107190261] packet4 5 sent
[1355107190262] ACK0 discarded
[1355107190262] packet5 6 sent
[1355107190262] ACK0 received
[1355107190263] packet6 7 sent
[1355107190263] ACK0 received
[1355107190263] packet7 8 sent

Fig 7-1