Exercise 6: Mini Project

Algorithmic Trading

Project Description

The project is an implementation of a simple algorithmic trading system that monitors stock exchange messages. It can decide if to buy or sell stocks based on messages containing the quantity and price information of the stocks. The system receives the quantity held, the bought price and the current price of a specific stock, and returns a trade order on this stock with a corresponding quantity.

Architecture

A local machine and a Raspberry Pi (which serves as a switch), connected directly using a single Ethernet cable are used.

The P4 program is compiled and run on the Raspberry Pi.

The local machine sends packets to the Raspberry Pi by a Python program.

Packets are exchanged between the local machine and the Raspberry Pi.

Operation

The Raspberry Pi runs the P4 program, starting the P4Pi switch.

The local machine runs a Python program that sends packets to the switch. The protocol header is coded as:

```
* The Protocol header looks like this:
*
*          0                1                2                3
* +----------------+----------------+----------------+----------------+
* |       P        |       4        |    Version     |      Dec       |
* +----------------+----------------+----------------+----------------+
* |            Identifier           |            Quantity             |
* +----------------+----------------+----------------+----------------+
* |                           Bought Price                            |
* +----------------+----------------+----------------+----------------+
* |                           Current Price                           |
* +----------------+----------------+----------------+----------------+
*
* P is an ASCII Letter 'P' (0x50)
* 4 is an ASCII Letter '4' (0x34)
* Version is currently 0.1 (0x01)
* Dec is an decison to take:
*      If deciding to buy, Dec = 0
*      If deciding to sell, Dec = 1
* Identifier: name of the stock
* Quantity: quantity of the stock held, or to be transacted
* Bought Price: price of the stock when bought
* Current Price: price of the stock currently
```

The switch captures the packet and processes the information based on the header: if the bought price is higher than the current price, it decides to buy the stocks; otherwise, it decides to sell the stocks. It writes back the decision into the header and sends the packet back to the local machine.

At the same time ,the switch decides a quantity of the stock that is to be bought or sold. Currently, it is set to half of the quantity held, which is in the header received. It writes back the quantity to be transacted into the header.

The processed information is contained in the packets sent back by the Raspberry Pi to the local machine.

Performance Tests

Currently, the program is tested using a Python script run on the local machine. The testing program receives an input from the user. The input contains an identifier (corresponding to the name of the stock), a quantity held, a bought price and a current price, which are all numbers. The program then encodes these values into the header to be sent to the switch.

Upon receiving the returned packets, the program displays the new information, where the decision to buy (displayed as 0) or sell (displayed as 1), and the quantity to buy or sell (replacing the original value of quantity held), are included.

Link to the repository

https://github.com/Y-J-Xue/CWM-ProgNets/tree/main/assignment6