Modelling front-running using High-Frequency Trading in the US Stock Market

by

Saagar Hemrajani

A thesis submitted in partial fulfillment for the degree of MEng Computer Science

Supervisor: Dr. Christopher Clack
Department of Computer Science

April 2016

This report is submitted as part requirement for the MEng Degree in Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text.

The report may be freely copied and distributed provided the source is explicitly acknowledged.
Modelling front-running using High-Frequency Trading in the US Stock Market
by Saagar Hemrajani
Supervisor: Dr. Christopher Clack

Abstract

Recent years have seen High Frequency Trading (HFT) become the centre of controversy in the financial markets. A particularly famous criticism of HFT is presented by financial journalist Michael Lewis in his 2014 book Flash Boys.

In Flash Boys, Lewis accuses high frequency traders (HFTs) of using their faster connection speeds to the exchanges to effectively predict orders being sent to them and order ahead of other traders. This is a technique Lewis refers to as front-running [1](pg.102).

Lewis also claims that Regulation National Market System (RegNMS), a key piece of recent financial regulation that governs the US equity market, affords HFTs an unfair advantage over other market participants. In doing so, Lewis alleges that the stock market is systematically rigged in favor of HFTs.

This paper uses a series of agent based simulations to investigate the obstacles, risks and challenges involved in front running orders within The United States National Market System (NMS).

A simulation is treated as an experiment that models a specific market scenario with a set of assumptions and various agents (brokers, HFTs, stock exchanges, CQS) interacting in a NMS context. Each experiment posits an alternative hypothesis that quantifies the conditions under which a HFT can make riskless profit by front-running orders by brokers. Every subsequent simulation is made more realistic through the involvement of additional agents, more advanced front-running techniques, and additional rules of RegNMS.

Taken cumulatively, the results of the experiments are used to form a qualitative assessment of the plausibility of Lewis’ claims that HFTs have the ability to systematically front-run orders and that RegNMS helps them do so [1](pg.107).

The project finds evidence that, in contrary to Lewis’ claims, HFTs simply don’t possess sufficient knowledge to guarantee riskless profit from front-running in the US equity markets. Their lack of certainty of order sizes, sending times, routing destinations and inter-agent delays makes front-running a highly risky strategy for HFTs. The experiments demonstrate that a front-running based trading model would result in heavily volatile returns and involve large market risk which would deter most HFTs. The project also finds that aspects of RegNMS, such as the order protection rule, act as a significant obstacle to potential front-runners and diminish the profitability of the strategy.
# Contents

Abstract i

List of Figures v

List of Tables vi

Abbreviations vii

1 Introduction 1

1.1 Preamble ................................................................. 1
1.2 Motivation ............................................................... 2
1.3 Project Proposal ......................................................... 3
1.4 Project Deliverables .................................................... 4
1.4.1 Primary Deliverables ................................................. 4
1.4.2 Secondary Deliverables .............................................. 5
1.5 Report Outline .......................................................... 5

2 Background 6

2.1 Market System .......................................................... 6
2.1.1 Exchanges and Order Books ........................................ 6
2.1.2 Market and Limit Orders .......................................... 7
2.2 The US Equity Market ................................................. 8
2.2.1 The National Market System ...................................... 8
2.2.2 Regulation National Market System (RegNMS) .................. 8
2.3 High Frequency Trading (HFT) and Co-location .................... 9
2.4 Front running ............................................................ 9
2.5 Interdyne Simulator .................................................... 10
2.5.1 The Messaging Infrastructure ..................................... 10
2.5.2 BATS Exchange and CQS ......................................... 10
2.6 Key Texts ................................................................. 11
2.6.1 Flash Boys, Michael Lewis [1] .................................... 11
2.6.2 Flash Boys: Not So Fast, Peter Kovac [2] ....................... 12

3 Experiment 1: Single Broker, Non-Routable Orders 13

3.1 Experiment Overview .................................................. 13
3.2 Background ............................................................. 13
3.3 Agents Overview ........................................................ 14
3.3.1 Agent Communication Topology .................................. 14
3.3.2 Agent Delay Matrix ................................................ 14
3.4 Assumptions ............................................................. 16
3.4.1 Market System Assumptions ...................................... 16
3.4.2 Agent Behaviour Assumptions .................................... 16
3.5 Agent Behaviour ........................................................ 17
3.5.1 Exchanges ................................................................. 19
3.5.2 Consolidated Quotation System (CQS) ............................... 19
3.5.3 Broker - ProbeV1.hs (Appendix Section B.3) ......................... 19
3.5.4 Broker - BrokerV1.hs (Appendix Section B.4) ....................... 20
3.5.5 High Frequency Trader - HFTV1.hs (Appendix Section B.5) ........ 21
  3.5.5.1 State ................................................................. 21
  3.5.5.2 Behaviour ......................................................... 21

3.6 Experiment Hypotheses ......................................................... 24
  3.6.1 Null Hypothesis ......................................................... 24
  3.6.2 Alternative Hypothesis ............................................... 25

3.7 Resolving Hypotheses ......................................................... 26
  3.7.1 Results ................................................................. 26
  3.7.2 Disproving Null Hypothesis ......................................... 26
  3.7.3 Justifying Alternative Hypothesis .................................. 27

3.8 Limitations of Experiment .................................................. 28

3.9 Conclusion ........................................................................... 29

4 Experiment 2: Single Broker, Routable Orders .......................... 30
  4.1 Experiment Overview ....................................................... 30
  4.2 Agents Overview ............................................................ 31
    4.2.1 Agent Communication Topology ................................... 31
    4.2.2 Agent Delay Matrix ................................................... 32
  4.3 Assumptions ................................................................. 32
  4.4 Agent Behaviour ............................................................. 32
    4.4.1 Broker - BrokerV1a.hs (Appendix Section B.6) ................. 34
    4.4.2 High Frequency Trader - HFTV1.hs (Appendix Section B.5) ... 35
  4.5 Experiment Hypotheses ..................................................... 37
    4.5.1 Null Hypothesis ....................................................... 37
    4.5.2 Alternative Hypothesis .............................................. 37
  4.6 Resolving Hypotheses ....................................................... 38
    4.6.1 Results ................................................................. 38
    4.6.2 Disproving Null Hypothesis ....................................... 38
    4.6.3 Justifying Alternative Hypothesis ................................ 40
  4.7 Limitations of Experiment ................................................ 41
  4.8 Conclusion ....................................................................... 41

5 Experiment 3: Multiple Brokers ............................................... 43
  5.1 Experiment Overview ....................................................... 43
  5.2 Agents Overview ............................................................ 43
    5.2.1 Agent Communication Topology ................................... 43
    5.2.2 Agent Delay Matrix ................................................... 44
  5.3 Assumptions ................................................................. 44
    5.3.1 Market System Assumptions ....................................... 44
    5.3.2 Agent Behaviour Assumptions .................................... 45
  5.4 Agent Behaviour ............................................................. 46
    5.4.1 Probe - ProbeV2.hs (Appendix Section B.7) .................... 49
    5.4.2 Broker agents .......................................................... 49
    5.4.3 High Frequency Trader - HFTV2.hs (Appendix Section B.9) ... 51
      5.4.3.1 State ................................................................. 51
      5.4.3.2 Behaviour .......................................................... 51
  5.5 Experiment Hypotheses ..................................................... 55
    5.5.1 Null Hypothesis ....................................................... 55
5.5.2 Alternative Hypotheses .............................................. 55
5.6 Resolving Hypotheses .................................................. 57
  5.6.1 Results .................................................................. 57
  5.6.2 Disproving Null Hypothesis ...................................... 57
  5.6.3 Justifying Alternative Hypotheses ................................ 59
5.7 Limitations of Experiment .............................................. 61
5.8 Conclusion ................................................................. 61

6 Conclusion ................................................................. 63
  6.1 HFTs ability and incentive to front-run ............................ 63
  6.2 Plausability of Lewis’ claims in Flash Boys [1] .................... 66
  6.3 Further Work ............................................................ 67

A Appendix A. Supporting Material ..................................... 68
  A.1 BYX: Messages Sent/Received ...................................... 68
  A.2 System Manual - Running Experiments ......................... 69
    A.2.1 Running Experiments .............................................. 69
    A.2.2 Modifying Experiment Setup .................................. 70
    A.2.3 Modifying Agent Behaviour .................................... 70

B Appendix B. Simulator Code ............................................ 72
  B.1 BATSMessages.hs ..................................................... 72
  B.2 SaagarExperiments.hs ................................................ 73
  B.3 ProbeV1.hs ............................................................... 76
  B.4 BrokerV1.hs ............................................................. 77
  B.5 HFTV1.hs ................................................................. 78
  B.6 BrokerV1a.hs ............................................................. 81
  B.7 ProbeV2.hs ............................................................... 82
  B.8 BrokerV2.hs ............................................................. 83
  B.9 HFTV2.hs ................................................................. 84

C Appendix C. Simulator Trace Files .................................. 88
  C.1 frExp1-trace ............................................................ 88
  C.2 frExp1a-trace .......................................................... 91
  C.3 frExp2-trace ............................................................ 94

Bibliography ................................................................. 99
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Schematic of an order Book</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Agent Communication Topology Diagram for Experiment 1</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Swim Lanes Diagram of Successful Front-running outcome of Experiment 1</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Pseudocode representation of the internal logic of HFT1 and HFT2 for Experiment 1</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Agent Communication Topology Diagram for Experiment 2</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Swim Lanes Diagram of Successful Front-running outcome of Experiment 2</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>Agent Communication Topology Diagram for Experiment 3</td>
<td>45</td>
</tr>
<tr>
<td>5.2</td>
<td>Swim Lanes Diagram of HFT successfully Front-running both brokers in Experiment 3 (Milestone 2 Onwards)</td>
<td>48</td>
</tr>
<tr>
<td>5.3</td>
<td>Pseudocode representation of the internal logic of HFT1 and HFT2 for Experiment 3</td>
<td>55</td>
</tr>
<tr>
<td>A.1</td>
<td>Main.hs</td>
<td>70</td>
</tr>
<tr>
<td>A.2</td>
<td>SaagarExperiments.hs</td>
<td>70</td>
</tr>
<tr>
<td>A.3</td>
<td>Comm.hs</td>
<td>71</td>
</tr>
</tbody>
</table>
# List of Tables

3.1 Agents in Experiment 1 ............................................. 14
3.2 Agent Delay Matrix for Experiment 1 .............................. 15
3.3 Probe Agent - Orders Sent ........................................ 19
3.4 Limit Order Book of BYX1 and BYX2 at Milestone 1 in timeline of Figure 3.2 ........................................ 20
3.5 Broker Agent - Orders Sent ........................................ 20
3.6 Limit Order Book of BYX1 and BYX2 at Milestone 2 in timeline of Figure 3.2 ........................................ 21
3.7 HFT1/2 State For Experiment 1 .................................... 22
3.8 HFT Agent - Orders Sent ........................................... 22
3.9 Limit Order Book of BYX1 and BYX2 at Milestone 3 in timeline of Figure 3.2 ........................................ 23
3.10 Breakdown of components in delay inequality for HFT’s profit-making conditions in Experiment 1 ........................................ 25
3.11 Results of certain trials of Experiment 1 ......................... 26

4.1 Agent Delay Matrix for Experiment 2 .................................. 32
4.2 Probe Agent - Orders Sent ........................................... 34
4.3 Limit Order Book of BYX1 and BYX2 at Milestone 1 in timeline of Figure 3.2 ........................................ 34
4.4 Broker Agent - Order Sent ........................................... 35
4.5 Broker Agent - Order Split (portions) ................................. 35
4.6 Limit Order Book of BYX1 and BYX2 at Milestone 2 in timeline of Figure 4.2 ........................................ 36
4.7 HFT Agent - Orders Sent ........................................... 36
4.8 Limit Order Book of BYX1 and BYX2 at Milestone 3 in timeline of Figure 4.2 ........................................ 37
4.9 Breakdown of components in delay inequality for HFT’s profit-making conditions in Experiment 2 ........................................ 38
4.10 Results of certain trials of Experiment 2 ......................... 39

5.1 Agents in Experiment 3 ............................................. 44
5.2 Agent Delay Matrix for Experiment 3 ................................ 44
5.3 Probe Agent - Orders Sent ........................................... 49
5.4 Limit Order Book of BYX1 and BYX2 at Milestone 1 in timeline of Figure 5.2 ........................................ 49
5.5 Broker1 Agent - Orders Sent ........................................ 50
5.6 Broker2 Agent - Orders Sent ........................................ 50
5.7 Limit Order Book of BYX1 and BYX2 at Milestone 2 in timeline of Figure 5.2 ........................................ 51
5.8 HFT1/2 State For Experiment 3 .................................... 52
5.9 HFT Agents - Orders Sent ........................................... 53
5.10 Limit Order Book of BYX1 and BYX2 at Milestone 3 in timeline of Figure 5.2 ........................................ 54
5.11 Table of delay events and outcomes for HFT .................... 58
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBO</td>
<td>Best Bid and Offer</td>
</tr>
<tr>
<td>BYX</td>
<td>BATS BYX Exchange Inc.</td>
</tr>
<tr>
<td>CQS</td>
<td>Consolidated Quotation System</td>
</tr>
<tr>
<td>FIX</td>
<td>Financial Information eXchange</td>
</tr>
<tr>
<td>FOK</td>
<td>Fill Or Kill</td>
</tr>
<tr>
<td>GTC</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>HFT</td>
<td>High Frequency Trading/Trader</td>
</tr>
<tr>
<td>HFTs</td>
<td>High Frequency Traders</td>
</tr>
<tr>
<td>IOC</td>
<td>Immediate or Cancel</td>
</tr>
<tr>
<td>NBBO</td>
<td>National Best Bid and Offer</td>
</tr>
<tr>
<td>NMS</td>
<td>National Market System</td>
</tr>
<tr>
<td>RegNMS</td>
<td>Regulation National Market System</td>
</tr>
<tr>
<td>SEC</td>
<td>U.S Securities and Exchange Commission</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Preamble

Ever since the creation of financial markets, there has been a debate about whether they are rigged in favor of those with faster access to news and prices.

In 1790, James Jackson spoke at the house of representatives of an emerging trend where speculative traders would hire fast-moving boats to travel across American cities and make trades based on political or financial news that hadn’t yet reached those regions. Jackson brandished these speculators as “rapacious wolves, preying upon the misfortunes of their fellowmen.”[3] This is one of the earliest examples of traders taking advantage of faster access to relevant news or prices to make trades ahead of the rest of the market; a practice that would come to be known as front-running.

In the coming centuries, the world would experience a series of revolutions in travel and communication technology. Traders obsession with speed would cause them to be the earliest adopters of technologies like stage-coaches, private horse expresses and even the telegraph. With each new technologically advanced trading technique came a wave of criticism from slower adopters who felt the new technology offered its users an unfair advantage in the financial markets.

The evolution of the financial markets continued with the invention of pneumatic tubes in 1896, the telephone in 1876, the computer in the 1950s and the development of electronic trading in 1971 [3].

Now, 225 years since Jackson’s speech, the world’s financial markets have evolved beyond recognition. Nevertheless, society’s dim view of the speed-obsessed speculative trader remains unchanged. Vicious, greedy and predatory are some of the terms used to describe today’s high-frequency traders who employ mathematical algorithms and fast trading systems to execute tens of thousands of orders a second.

In financial markets, information asymmetry will always exist and faster access to the markets or relevant news will continue to be highly sought. History shows that as long as high-speed traders operate, there will be a vocal crowd who demonize their methods and claim the market is rigged.
1.2 Motivation

High Frequency Traders (HFTs) are characterized by their trading strategies which involve using computer programs to make numerous orders, trading in high volumes with extremely short holding periods. Their ability to trade with such speed comes from their advanced connectivity to the exchanges they trade on. HFTs are the main consumers of the premium co-location services offered by most US stock exchanges. This service places the HFT servers in the same data center as the exchange servers, allowing the HFT to receive quotations from and submit orders to the exchange with minimal latency.

In recent years High Frequency Traders (HFTs) have amassed significant interest from politicians, regulators and the media. Despite their notoriety, there is still much uncertainty amongst the financial community and the general public regarding the impact that HFTs have on the market.

Proponents of HFT assert that HFTs are generally beneficial to the market and all its participants. They argue that the HFT’s high order volumes add liquidity to the markets, causing smoother price movements and reduced volatility. They also claim that the presence of HFTs reduce bid and offer spreads, leading to more accurate price discovery and improving overall market quality.

Nonetheless, HFTs have been “besieged by accusations that they cheat slower investors” [4]. HFT firms are accused of using their faster co-located connections to take advantage of other market participants. It is not just the media who criticize HFTs, some regulatory bodies have also intervened, with several European countries seeking a ban on HFTs in 2012. In 2010, a Dutch regulatory body even called upon high frequency traders justify “their behaviour and demonstrate that they make a positive contribution to the market as a whole” [5].

A particularly outspoken critic of high frequency trading is the financial journalist and author Michael Lewis, who devoted his 2014 book Flash Boys [1] to the topic. In Flash Boys, Lewis alleges that HFTs use their faster connections speeds to engage in front-running.

“*A small class of insiders with the resources to create speed were now allowed to preview the market and trade on what they had seen.*” – Michael Lewis, Flash Boys, 2014 [1] pg.100

Lewis also claims that Regulation National Market System (RegNMS), a key piece of recent financial regulation that governs the US equity market, favors HFTs over other market participants. In doing so, Lewis makes the argument that that the stock market is systematically rigged in favour of HFTs.

“*Reg NMS was intended to create equality of opportunity in the U.S. stock market. Instead it institutionalized a more pernicious inequality*” – Michael Lewis, Flash Boys, 2014 [1] pg.100

“*Every systemic market injustice arose from some loophole in a regulation created to correct some prior injustice*” – Michael-Lewis, Flash Boys, 2014 [1] pg. 102

While there is some support for Lewis’s allegations regarding RegNMS and HFT firms, the evidence
provided in Flash Boys remains largely anecdotal. The book also shows a significant bias which is evident from the fact that it contains no interviews with any experienced HFTs themselves.

Despite the lack of concrete evidence, Flash Boys casts doubt on the fairness of the US financial market. The sheer size of the financial system and the complex nuances involved in RegNMS makes the task of assessing Lewis’s claims immensely challenging. Nevertheless, considering the severity of Lewis’s accusation and the fact that HFTs have accounted for upwards of 50% of all US equity transactions in recent years [6], the importance of conducting an unbiased empirical analysis of the behaviour of HFTs within a market governed by RegNMS cannot be understated.

There is a precedence of researchers at UCL using agent based simulation to model interactions between participants in a market scenario. Court and Clack [2013] [7] used a simulator to model the events of the 2010 Flash Crash. Chopra [2015] [8] uses an agent based simulator to investigate the conditions required for profitable front-running in a pre-RegNMS market place. Bakshi [2015] [9] extends the aforementioned simulator used by Chopra by building a BATS exchange replica, creating Financial Information Exchange (FIX) protocol messaging between agents and implementing some key portions of Regulation NMS including a Consolidated Quotation System (CQS) agent that upholds the National Best Bid and Offer (NBBO).

This goal of this project is to use the simulator, last updated by Bakshi, to run experiments that investigate the profit-making ability of a HFT engaged in front-running orders in a market governed by RegNMS.

The results of these experiments should illustrate the conditions under which a HFT can make a profit and the obstacles faced when front-running under RegNMS. These results can be extrapolated to make educated judgments regarding Lewis’s accusations that HFTs are able to make risk-free profits through front-running and that RegNMS favors HFTs.

This project will not be sufficient to outright refute or endorse any claims made by Lewis in 'Flash Boys'. Instead, the focus of this project will be on presenting a repeatable procedure for designing and analysing the behaviours of agents in a market simulation. This shall hopefully add scientific rigor to the analysis of market simulations and raise the standard for future research in this area.

1.3 Project Proposal

The focus of this project will be on defining and categorizing the challenges associated with front-running as described by Michael Lewis in Flash Boys. Some foreseeable intricacies to front-running that will be considered include timing issues, order restrictions, liquidity problems and market risk. The project will also discuss the validity of Lewis’ claim that a HFT firm has the knowledge and ability to make risk-free profit by front-running orders in a market place governed by RegNMS. Additionally, there will be some investigation into Lewis’ secondary accusation that RegNMS favors HFTs over slower traders.
The bulk of the analysis will stem from a series of agent based simulations that model a RegNMS governed market of a given security. These simulations will reproduce normal market conditions and aim to replicated the price-shifting phenomena experienced by the broker in Flash Boys; exploring how it could be caused by a HFT engaged in front-running. Agents in these simulation will behave as brokers, HFTs engaging in front-running, Exchanges or the CQS.

The analysis of these simulations will provide a quantitative measure of the conditions under which the HFT can make a profit by front running and the market/agent assumptions required. Furthermore, the analysis should yield some conclusions regarding the risk involved in front-running.

Although each simulation will have a self-contained analysis, the results will be used collectively to posit general obstacles and profit-making conditions for a HFT engaging in front-running in a market governed by RegNMS. Each simulation will also contribute to the discussion regarding the plausibility of Lewis’ claim that HFTs have the ability to systematically front-run orders. Lastly, the project will deliver a qualitative assessment of the level of incentive for an HFT to follow a business model based on front running.

1.4 Project Deliverables

1.4.1 Primary Deliverables

1. **Experiment 1.** A simulation of a two-exchange, RegNMS market with a HFT and a Broker submitting non-routable orders

   - Design and Implement the front-running experiment including its assumptions and the agent’s behaviour
   - Create quantitative null and alternative hypotheses relating to the conditions necessary for profitable front-running under the given assumptions
   - Repeat simulation with different inter-agent delay permutations and use the results of the experiment to justify or disprove the appropriate hypotheses
   - Provide concluding remarks of general conditions necessary for profitable front-running

2. **Experiment 2.** Experiment 2 uses the same number of agents as Experiment 1 except the broker now submits routable orders. This experiment investigates the order protection Rule (Rule 611) of RegNMS and its impact on the profitability of front-running.

3. **Experiment 3.** Experiment 3 investigates the intricacies involved in front-running two orders from different brokers simultaneously. Like experiment 1, it is a simulation of a two-exchange, RegNMS market with one HFT but with two brokers submitting non-routable orders to both exchanges.

4. Qualitative yet justified assessment of the following questions using the results of the experiment
• To what extent do HFTs operating the in the US stock market possess the correct knowledge and ability to front-run orders in a riskless manner in a market governed by RegNMS?
• To what extent is a front-running based business model an effective use of capital for a high frequency trading firm operating in a market governed by RegNMS?
• To what extent could the price shifting phenomenon outlined in Flash Boys be attributed to HFT firms engaging in front-running on a mass scale?
• To what extent does RegNMS assist HFTs in front-running?

1.4.2 Secondary Deliverables

1. Agent wrappers functions created using modular design such that future users can use modify the same code when adapting the behaviour of the agents

2. Additional simulator methods to allow for the faster creation of BATS orders in FIX format

3. Outline of other possible simulations that could add value to the analysis and be taken up as a further project

1.5 Report Outline

This report is structured such that in addition to the introduction, background and conclusion, each experiment forms its own chapter. Each experiment is split into sections including an overview of the experiment, its assumptions, the behaviour of agents involved, a set of hypotheses, the results and some conclusions. For instructions on running the experiments, refer to the System Manual in Appendix section A.2
Chapter 2

Background

2.1 Market System

2.1.1 Exchanges and Order Books

A stock exchange is simply a market place where stocks/shares of public companies are bought and sold. The exchange’s job is to manage an order book for every security traded and enforce certain rules to ensure market fairness.

A limit order book (LOB) for a given security consists of a list of bids and list of asks/offers. Each list stores price and quantity tuples that indicate how many shares are willing to be sold/bought at that price point. The order books are populated when market participants send in limit orders. The limit orders at each price point are sorted by the time the exchange receives them, meaning that the oldest order at a given price point is matched/executed first. Figure 2.1 shows the structure of an order book.
In an order book, the lowest priced ask is referred to as the best ask/offer and the highest price bid is referred to as the best bid. When the best ask is lower than the best bid, it results in what is known as a ‘crossed book’. Some exchanges do not accept orders that would cross the order book while others will uncross the book by executing the crossed ask and bid orders with each other. The gap between the best bid price and best ask price in an uncrossed order book is called the bid-ask spread.

A public exchange publishes market data including the limit order book for all securities. Additionally, an exchange will publish the execution details of any trade that occurs on the exchange. This information is anonymized so the counter-parties of the trade cannot be identified but the price and quantity of the trade is published. There are certain private exchanges (known as dark pools) that are not available to the investing public and do not publish such details.

2.1.2 Market and Limit Orders

There are two main types of orders accepted by public exchanges; limit and market orders. An easy way to visualize the difference is by considering that limit orders populate the LOB of a given security while market orders remove limit orders from the LOB. Unless explicit expiry conditions are given, limit orders remain on the exchange until they are matched with another limit order (in the case of a crossed book) or a market order.

A market order is an intention to buy/sell a security at market price. For buying the security, the market price is the best ask on the LOB, for selling the security the market price is the best bid on the LOB. As such, a market order specifies the quantity of shares but not the price. It is the job of the exchange to match market orders to limit orders. In order to ensure fairness, incoming market orders are arranged in a queue based on their arrival time. Therefore, the first market order to arrive will be matched first.

In submitting an order, a market participant specifies whether they want to buy (bid) the security or sell (offer) the security as well as the number of shares they want to buy/sell. In a limit order, the price must also be specified.

In addition, the sender can also specify certain other conditions for their order. Firstly, in the US national market system (NMS), one can specify whether an order is routable or not. If it is routable, then the order can be partially or entirely forwarded on to be executed at other exchanges at a better price. A sender can also specify certain expiry conditions for their order.

The following are some examples of expiry conditions:

- **Immediate or Cancel (IOC):** The order is cancelled if it cannot be matched immediately
- **Fill or Kill (FOK):** If the entire order cannot be executed entirely and immediately, it is cancelled
- **Good till cancelled (GTC):** The order remains on the book, or in the queue until it is fulfilled
- **Day (D):** If not filled, the order is cancelled at the end of the trading day
Chapter 2. Background

2.2 The US Equity Market

The US secondary equity market is made up of 8 public exchanges, more than 20 alternative trading systems (another term for dark pools) and dozens of agency-execution brokers. US stocks are one of the most heavily traded securities on the global markets, with more than 13.9 billion shares being exchanged in 2015 [10].

The popularity of US stocks amongst investors stems from the fact that the market is heavily regulated and transparent.

The regulation of the stock market at a federal level began when the US congress passed the Securities Exchange Act in 1934 [11]. This legislation is what established the Securities and Exchange Commission (SEC), which remains responsible for enforcing securities law till today.

2.2.1 The National Market System

The regulations that govern the US stock market today were designed over a series of amendments to the Securities Act passed by congress throughout the decades. In 1975, congress directed the SEC to facilitate the creation of a national market system (NMS) where clearance and settlement of securities transactions were conduction on a national scale [12]. This led to the forming of the Consolidated Quotation System (CQS), which collected market data from the individual exchanges and disseminated information on the national best bid and offer (NBBO) for any given security [13]. The ultimate goal of the NMS was to permit each investor the opportunity for the best possible execution of their order, regardless of where in the system it originated [14].

2.2.2 Regulation National Market System (RegNMS)

The next big change to the regulation of the US equity market came when the SEC’s proposal to modernize the regulatory structure of the U.S. equity markets were approved in 2005. The new regulatory structure would come to be known as Regulation National Market System (RegNMS). Of the 12 rules that make up RegNMS, one of them (rule 611) is vital to the context of this project.

Rule 611 is the Order Protection Rule that aims to promote intermarket price protection. An important part of rule 611 is the maintenance NBBO of a given NMS security. The NBBO is disseminated to any investor or exchange with a subscription by the CQS. Rule 611 requires the exchange to take responsibility for ensuring that every order they receive is executed at the best price offered by the market. According to rule 611, if a market order is marked as routable and not IOC or FOK then it must be broken up into smaller quantities and routed to other public exchanges, if this is necessary to get the best execution price [15].
The exchanges do this by keeping track of the NBBO for every NMS security and routing orders for those securities to the exchanges that are offering the best price. The goal of 611 is to help the investor by guaranteeing that regardless of the destination and size of their order, each share will be executed at the best price available in the entire market at the time their order arrives.

### 2.3 High Frequency Trading (HFT) and Co-location

HFT is a term used to describe a trading strategy which involve using computer programs to make numerous orders, trading in high volumes with extremely short holding periods. High frequency traders are able to execute millions of trades in a matter of seconds because of their fast connection to the various stock exchanges. In recent years, HFT has become a large part of the equity market, responsible for around 50% of the entire trading volume in 2012 [4].

Many HFTs establish this fast connection by taking advantage of a service offered by many exchanges called co-location. Co-location is when the HFT servers (that house the trading algorithms) are places in the same data center as the exchange servers (that house LOB and trade execution data). Through this service, the HFT is able to minimize the latency when sending orders to the exchange and when receiving market data from the exchange.

### 2.4 Front running

In finance, front-running is a practice where a stock broker executes their own orders on a security by taking advantage of the “advance knowledge of pending orders from its customer” [16]. If the customer submits an order to buy, the broker will submit their own order to buy to drive the price up before filling the customer’s order. If the customer submits to sell, the broker will do the same, submitting their own order before filling the customer’s. The practice of front-running is unethical and clearly illegal.

In his 2014 book Flash Boys [1], Michael Lewis builds a criticism of HFT, accusing them of ‘front-running’ the orders of slower investments. Lewis’ use of the word ‘front-running’ to describe the alleged actions of HFTs does not reflect the traditional definition of the word. Lewis alleges that HFTs use their faster connections speeds to the exchanges to effectively predict orders being sent to them. He claims that the HFTs use these informed predictions to order ahead of other traders. This is the technique Lewis refers to as front-running. For the remainder of the report, ‘front-running’ will refer to Lewis’ description of the term.
2.5 Interdyne Simulator

Interdyne is the agent-based simulator that will be used to build the experiments on which the central thesis of this project will be based. The simulator was originally created in 2011 using the functional language Miranda by Christopher Clack of University College London. In 2014 the simulator was ported to Haskell by Liu and Clack in 2014 [17]. In 2015, Chopra [8] enhanced the simulator by removing bugs and creating new agents for further use. In the same year, Bakshi [9] further improved the simulator by implementing FIX messaging between agents, and creating an exchange agent which mimics the BATS BYX exchange and an agent replica of the CQS.

Bakshi’s updates allow for the creation of experiments that simulate the functioning of a market governed by RegNMS. It is this latest version of the simulator that will be used to investigate front-running in within a RegNMS governed market.

The simulator functions in discrete time, meaning that each simulation lasts for a certain number of time steps and each agent performs one round of actions at each time step. The simulator is also deterministic, meaning that it will behave the same way every time it runs.

2.5.1 The Messaging Infrastructure

The most important part of the simulator is the messaging infrastructure. The simulator supports one-to-one and one-to-many communication. At each time-step, each agent performs logic that analyses received messages and creates messages to be sent. In designing experiments, a precise communication topology can be developed to determine the nature of interactions between components.

One vital feature when designing the communication topology between agents is the information delay. Interdyne allows the experimenter to decide the exact number of time steps of delay in any one-to-one or one-to-many interaction. The delay is enforced by postponing the sending of any messages/broadcasts by the number of time-steps corresponding to the delay. The use of such delays will be integral in designing and analyzing the front-running experiments. In designing the experiment, delays can be used to create a condition where a certain agent (i.e. a HFT) has a much smaller delay to the exchange than another (i.e a broker). The manipulation of such delays and the behaviour of the agents can then yield valuable information regarding the technical intricacies associated with front-running.

2.5.2 BATS Exchange and CQS

The BATS Exchange is an agent created by Bakshi to simulate the BATS BYX exchange. The agent uses the FIX messaging system to communicate with other market participants and operates in accordance with RegNMS. This means that it publishes its best bid and offer (BBO) to the CQS agent and receives
Chapter 2. Background

an NBBO at each time-step. Additionally, it also upholds rule 611 of RegNMS meaning that it splits and re-routes any orders it receives to the exchange with the best execution price [18].

In order to facilitate easier sending of orders, an extra function was added to the code of the BATS exchange agent. As a result of this update, agents can now from orders to exchanges concisely with a single like of code. For details on the update, consult appendix section B.1.

The CQS agent was also created by Bakshi and it aims to function identically to the CQS in the US equity market. The main role of the CQS agent is to maintain the NBBO. At each time step, it collates the BBO at each exchange and broadcasts the updated NBBO to all subscribed agents. Since each exchange consults the NBBO when deciding whether to route incoming orders, the CQS plays an important role in the investigation of front-running.

Consult Bakshi’s report [9] for a further explanation of the behaviour and implementation of the BATS Exchange or CQS Agent.

2.6 Key Texts

In the initial stages of the project, an extensive literature review was conducted into the topics of high frequency trading and agent based simulations. This section provides a brief overview of the two texts that served as integral background reading and stimulus for this project.

2.6.1 Flash Boys, Michael Lewis [1]

Flash Boys is 2014 creative non-fiction book that chronicles the rise of high frequency trading in the US equity market. It follows the experiences of Brad Katsuyama in his role as the global head of electronic sales and trading at Royal Bank of Canada (RBC). Flash Boys describes how Katsuyama begins suspecting that his orders were being predicted by HFTs with fast connections to various exchanges. When trading, Katsuyama would split large orders into smaller ones and send them to multiple exchanges to obtain the best execution price. In 2008, Brad began noticing that price of these securities would shift almost immediately as he submitted the orders. This led him to suspect that he was being taken advantage of by stock scalpers.

Katsuyama theorized that HFTs would notice his first order execute and predict the destinations of the other orders. Then, using their fast connections, the HFTs would send orders to buy the securities ahead of Katsuyama. Thus, by the time Katsuyama’s orders arrived, the offers he saw when submitting the order would no longer be available on the Limit Order Book and his trade would execute at a higher price. The shift in price seemed almost immediate to Katsuyama since the entire process would take milliseconds. Lewis refers to this strategy as front-running and then goes on to explain how Katsuyama assembles a team that works to outmanoeuvre these front-runners and get better execution prices.
Chapter 2. Background

Lewis bases much of his criticism of high frequency trading on the experiences and suspicions of Brad Katsuyama and claims that the US stock market is rigged by HFT traders. Lewis estimates that the HFT’s unfair advantage amounts to a cost of over $5 Billion a year to other investors.

Flash Boys’ entertaining story telling technique appeals to the mass audience. This is evident by the fact that it topped the New York Times Best Seller list for four consecutive weeks. As such, Flash Boys has the unique honour of simultaneously being the harshest criticism High Frequency Trading and the most widely read book on the topic.

The motivation for this project stems from a concern for the severity of Lewis’ accusations and the need to verify the true impact of High Frequency Trading on the US equity market. Most of the controversy surrounding HFT is caused by the misinformation and general ignorance on the topic. This project hopes to delve deeper into the technical realities of the US equity market, RegNMS and high frequency trading to rigorously evaluate the claims set out by Lewis in Flash Boys.

2.6.2 Flash Boys: Not So Fast, Peter Kovac [2]

‘Flash Boys: Not So Fast’ is a point-by-point rebuttal of Michael Lewis’ Flash Boys written by Peter Kovac. Kovac’s background as a programmer and then Chief Operating Officer at a HFT firm gives him a unique perspective in the debate regarding the impact of HFT on the market. In his book, Peter illustrates the faults in Lewis’ claims. Kovac’s main criticism is that Lewis does not present the views and experiences of any actual high frequency traders. Kovac argues that Michael Lewis does not fully understand how HFTs operate and how orders are processed in the national market system.

Kovac claims that there are significant obstacles to front-running in a RegNMS market, making it illegal and definitely not worthwhile for any HFT to engage in the practice. Kovac explains that in order to front-run an order successfully, the HFT would need to perform five steps. “1) Determine the price and quantity of shares of your order 2) Buy the same amount of shares you want, before you do 3) Manipulate the market price upward 4) Sell the shares back to you at the higher price 5) Avoid anyone else in the market who could disrupt the scam” [2]. Kovac explains how the electronification of stock markets and the introduction of RegNMS in 2007 have made these barriers illegal and almost impossible to breach. Kovacs also explains order routing and how difficult it is for a HFT to accurately predict the destination of a routed order.

Overall, Kovac’s book is a much needed second opinion on the impact of HFT on the US equity market. Reading the opposing views of both Lewis and Kovac puts one in a perfect position to begin thorough experimentation into the impact of HFTs and the profitability of a front-running trading strategy in a RegNMS market.
Chapter 3

Experiment 1: Single Broker, Non-Routable Orders

The objective of this experiment is to investigate whether it is possible, and if so under what conditions and with what risk is it possible, for a HFT to make a profit by front running a non-routable order from a single traditional broker in a RegNMS market system.

3.1 Experiment Overview

This experiment is an agent based simulation of an attempt at order front running within the US secondary equity market. The experiment depicts the interactions between two exchanges, a broker and a high frequency trader (HFT) who has co-located access to both exchanges. The broker is tasked with executing trades for its clients and sends two orders, one to each exchange, to buy a given security at market price. The HFT aims to use its fast access to each exchange and knowledge of trade execution and communication delays to submit its own orders and shift the price before the broker’s second order is executed. If the HFT is successful, the second market order will execute at a higher price than originally seen by the broker and the HFT will make a profit from the difference.

3.2 Background

This experiment was designed to replicate the experiences of Brad Katsuyama, head of Royal Bank of Canada’s stock desk in New York in 2008. Brad started to suspect the US stock market was rigged when he noticed that the prices seemed to move up the moment he placed a market buy order. In his book Flash Boys, Michael Lewis outlines how Brad began to question whether the exchanges were providing a certain group of traders (high frequency trader) with a faster access to prices and trade execution data.
3.3 Agents Overview

The real-life entities involved in this scenario are represented as agents in our simulation and shown in Table 3.1.

<table>
<thead>
<tr>
<th>Agent Type</th>
<th>Agent Name</th>
<th>Agent Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchanges</td>
<td>BYX1</td>
<td>The stock exchanges in our experiment are representative of any US secondary equities market. The behaviour of these exchanges are designed to resemble the American BYX exchange, which is an SEC approved BATS exchange. The BYX exchange functions within a RegNMS market system and is capable of upholding the NBBO (National Best Bid and Offer) through order routing. The BYX exchange also offers a premium co-location service. BYX1 is the first instance of the BATS BYX and BYX2 is the second instance. BYX2 functions independently of BYX1.</td>
</tr>
<tr>
<td></td>
<td>BYX2</td>
<td></td>
</tr>
<tr>
<td>CQS (Consolidated Quotation System)</td>
<td>CQS</td>
<td>The CQS agent consolidates the limit order books of all exchanges to provide the best overall price for any traded equity security. This information is known as the National Best Bid and Offer (NBBO) and is used by the individual exchanges to route market orders to the destination with the best price.</td>
</tr>
<tr>
<td>Probe</td>
<td>ProbeV1</td>
<td>The probe agent populates the limit order book for the security on both exchanges by sending bids and asks. This creates a price for the instrument and sets up a market environment that will be the basis for our front running scenario.</td>
</tr>
<tr>
<td>Broker</td>
<td>BrokerV1</td>
<td>The broker agent represents a traditional broker in the equities market. The broker is responsible for executing a client’s orders and ensuring the client receives the best price available.</td>
</tr>
<tr>
<td>High Frequency Trader</td>
<td>HFT1</td>
<td>In this experiment the HFT (high frequency trader) is represented as two agents, HFT1 and HFT2. This is done in order to distinguish between the two trading applications, each co-located (stored) at a different exchange’s data center. The two HFT agents also have a communication link between them that will be used to exchange information facilitate their goal of front-running. While the two co-located applications are portrayed as separate agents, it is assumed that they both belong to a single high frequency trading firm. HFT1 is stored in the same data center as BYX1 while HFT2 is stored in the same data center as BYX2.</td>
</tr>
<tr>
<td></td>
<td>HFT2</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Agents in Experiment 1

3.3.1 Agent Communication Topology

Figure 3.1 is a communication topology diagram illustrating the connection and delays in communications between the agents in this experiment.

3.3.2 Agent Delay Matrix

Table 3.2 is a matrix representation of the delays in communication (measured in abstract units of time) between the various agents in the experiment. Note that only connections relevant to this experiment are shown. In interpreting this matrix two points should be noted:

- Firstly, it is important to note that the delays are not necessarily symmetric, meaning that the time taken for a message to go from Agent A to B is not necessarily the same as the time taken for a message to go from B to A.
Additionally, every pair of agents are not guaranteed a direct line of communication. This accounts for the empty cells in the matrix. An empty cell mean that the two agents do not have a line of communication or never communicate during the experiment, making the delay unimportant.

![Figure 3.1: Agent Communication Topology Diagram for Experiment 1](image)

<table>
<thead>
<tr>
<th>From</th>
<th>BYX1</th>
<th>BYX2</th>
<th>CQS</th>
<th>Probe</th>
<th>Broker</th>
<th>HFT1</th>
<th>HFT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1</td>
<td>-</td>
<td>-</td>
<td>δ8</td>
<td>δ1b</td>
<td>δ3b</td>
<td>δ5b</td>
<td>-</td>
</tr>
<tr>
<td>BYX2</td>
<td>-</td>
<td>-</td>
<td>δ9</td>
<td>δ2b</td>
<td>δ4b</td>
<td>-</td>
<td>δ6b</td>
</tr>
<tr>
<td>CQS</td>
<td>δ8b</td>
<td>δ9b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Probe</td>
<td>δ1</td>
<td>δ2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Broker</td>
<td>δ3</td>
<td>δ4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HFT1</td>
<td>δ5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>δ7</td>
<td>-</td>
</tr>
<tr>
<td>HFT2</td>
<td>-</td>
<td>δ6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>δ7b</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 3.2: Agent Delay Matrix for Experiment 1**
3.4 Assumptions

“The only constraint was how fast an electronic signal could travel between Chicago and New York, more-precisely, between the data center in Chicago that housed the Chicago Mercantile Exchange and a data center beside the Nasdaq’s stock exchange in Carteret, New Jersey.” – Michael Lewis, Flash Boys, 2014 [1] (pg.15)

3.4.1 Market System Assumptions

Assumption 1 – Only the listed agents exist in this market

From this assumption we can extrapolate the following conclusions:

- The given security is only traded on two exchanges, BYX1 and BYX2
- The Bid/Ask prices across these two exchanges represent the NBBO for this given security
- The broker agent is the only broker submitting orders for this specific security
- There is only one HFT with co-located access to both exchanges

Assumption 2 – The trade execution information published by the exchange is anonymous and does not identify the counter parties

The BATS BYX exchange is required to publish information on any trade executed on the exchange. This information includes the symbol of the security, the size and price of the trade. This is true in reality and is mentioned by Michael Lewis in Flash Boys. This assumption makes it more complicated for the HFT to spot a front running opportunity by scanning executed trades as there is no way to identify which broker made the trade.

3.4.2 Agent Behaviour Assumptions

Assumption 3 -- The HFT knows the total order size being executed by the Broker

In order to give the HFT some way to gauge when a front running opportunity is available, the experiment assumes that the HFT knows the total number of shares that the broker is looking to buy. By knowing the order quantity, the HFT can implement some logic to deduce what orders the broker will send in order to get the best price for their clients. Knowing this information allows the HFT to identify the front running opportunity and act accordingly.

Assumption 4 – Both HFT1 and HFT2 know the delays between the following agents:

- The broker and each exchange
- Itself and its co-located exchange
Chapter 3. Experiment 1: Single Broker, Non-Routable Orders

- Itself and the other HFT agent (co-located at the other exchange)
- The CQS and its co-located exchange

Although not entirely realistic, we assume that the HFT knows the above delays as they give the HFT a fighting chance at gauging front running opportunities. At each time-step, the BYX exchange sends the details of its limit order book (LOB) to all those that subscribe to it. Using the LOB and the delays, the HFT can figure out what LOB was seen by the broker at each time-step. This can be used to infer what orders the broker would have sent to obtain the best execution price. Knowing what orders the broker sent is the first step to identifying and acting upon a possible front running opportunity.

Below is an example of the type of logic that would be performed by the HFT:

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Delay (time steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX</td>
<td>HFT</td>
<td>A</td>
</tr>
<tr>
<td>BYX</td>
<td>Broker</td>
<td>B</td>
</tr>
<tr>
<td>Broker</td>
<td>BYX</td>
<td>C</td>
</tr>
</tbody>
</table>

Given the above delays we can infer that:

- The limit order book seen by HFT at time Y was seen by Broker at time $Y - A + B$
- Trade execution on BYX at time Z was sent by Broker at time $Z - C - B + A$

**Assumption 5** – The broker sends all its market orders at the same time-step

This is an assumption that was necessary to allow the HFT to properly gauge a front running opportunity. Since all orders were sent at the same time, when the first order gets executed, the HFT can figure out when the broker sent the orders and know what order books the Broker saw at that time. Then, by applying some logic, the HFT can predict the other order sent by the broker. Having certainty as to what orders will be arriving at what exchange is necessary for the HFT to engage in front running.

### 3.5 Agent Behaviour

This section will explain the role and behaviour of all of the agents in the simulation. The explanation will refer to the a swim lanes diagram which illustrates the messages sent and actions taken by each agent in the simulation. Figure 3.2 reflects the timeline of actions in the event where the front running effort is successful (i.e the HFT makes a profit).
Figure 3.2: Swim Lanes Diagram of Successful Front-running outcome of Experiment 1.
3.5.1 Exchanges

BYX1 and BYX2 are the two equities exchanges that trade a given security which the broker has been instructed to buy. Both agents behave in an identical manner and observe the regulations of a RegNMS market system. Their behaviour, including the types of messages each sends/receives, its price matching and its order execution method, have been modelled after the existing BATS BYX exchange. Like the BATS BYX exchange, the BYX1 and BYX2 agents are assumed to offer a premium co-location service where traders can get minimal latency access by having their applications stored in the same data centre to the exchange.

The specific responsibilities of agents BYX1 and BYX2 are best outlined through an understanding of the type of messages they send and receive. An outline of this can be found in Appendix A.1

3.5.2 Consolidated Quotation System (CQS)

The Consolidated Quotation System is an integral part of the market of exchange traded securities in the US and is represented by the CQS agent in our experiment. The CQS agent chiefly responsible for maintaining the NBBO for every given security traded on the existing exchanges. To do this, the CQS receives LOB and BBO broadcast from each exchange and each time-step. It then collates this data and broadcasts its own message, outlining the best bid and offer price for each traded instrument (also known as the NBBO).

3.5.3 Probe - ProbeV1.hs (Appendix Section B.3)

The probe agent in this experiment is responsible for populating the limit order books of each exchange and setting up a scenario where a Broker can submit market orders which the HFT will attempt to front run. All of the limit orders are marked Good Till Cancelled because they need to remain on the LOB throughout the simulation.

Table 3.3 shows the orders sent by the probe agent to the exchanges

<table>
<thead>
<tr>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>23.95</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>400</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.00</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>600</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.10</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>23.95</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.05</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

**Table 3.3: Probe Agent - Orders Sent**

Table 3.4 shows the limit order book of BYX1 and BYX2 at Milestone 1 when all of the probe’s limit orders have been received and placed on the LOB.
Chapter 3. Experiment 1: Single Broker, Non-Routable Orders

20

Table 3.4: Limit Order Book of BYX1 and BYX2 at Milestone 1 in timeline of Figure 3.2

3.5.4 Broker - BrokerV1.hs (Appendix Section B.4)

The broker agent in this experiment represents a typical broker performing agency trades for their clients. The broker is not co-located to the either exchange but does have a faster connection to BYX1 than BYX2. The broker also has a level 3 data subscription to each exchange, giving it access to the (albeit slightly delayed) limit order book of each exchange.

In this experiment, we assume that the broker is instructed by a client to purchase 600 shares of the given security at market price. The broker inspects the limit order book at each exchange and notes the following:

- The best offer/ask for this specific security can be found on BYX1 where there are 400 shares available at $24.00.
- The second best offer/ask is found on BYX2 where 200 shares are available for $24.05.

As such, in order to obtain the best price for its client, the broker chooses to split its 600 share purchase into two orders, one sent to each exchange. Table 3.5 shows the market orders sent by the broker.

<table>
<thead>
<tr>
<th>Order #</th>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>BYX1</td>
<td>Market</td>
<td>400</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Immediate Or Cancel</td>
</tr>
<tr>
<td>#2</td>
<td>BYX2</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Immediate Or Cancel</td>
</tr>
</tbody>
</table>

Table 3.5: Broker Agent - Orders Sent

The experiment is designed such that the broker has a faster connection to BYX1 than BYX2. As a result, the broker’s market order to BYX1 will arrive and execute first. After the execution of the trade for 400 shares on BYX1 between broker and probe, we arrive at Milestone 2 in the swim lanes diagram (Figure 3.2).

Table 3.6 shows the LOB of both exchanges after the execution of the broker’s first order (at Milestone 2). The empty yellow cell illustrates the absence of the 400 shares that were just executed.
Chapter 3. Experiment 1: Single Broker, Non-Routable Orders

### Table 3.6: Limit Order Book of BYX1 and BYX2 at Milestone 2 in timeline of Figure 3.2

<table>
<thead>
<tr>
<th>Bid Size</th>
<th>Price</th>
<th>Ask Size</th>
<th>Bid Size</th>
<th>Price</th>
<th>Ask Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.10</td>
<td>600</td>
<td></td>
<td>24.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.09</td>
<td></td>
<td></td>
<td>24.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.05</td>
<td></td>
<td></td>
<td>24.05</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td></td>
<td>24.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td></td>
<td>23.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td></td>
<td>23.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td></td>
<td>23.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td></td>
<td>23.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td></td>
<td>200</td>
<td>23.95</td>
<td></td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
<td>23.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
<td>23.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.5.5 High Frequency Trader - HFTV1.hs (Appendix Section B.5)

In this experiment, there is one high frequency trader who is portrayed by two agents, HFT1 and HFT2. Both HFT agents are assumed to have a level 3 data subscription and thus receive a complete LOB from both exchanges at each time-step. It should also be noted that both HFT agents are instances of the generic HFT agent type, who’s behavior is defined in HFTV1.hs (Appendix Section B.5).

The goal of the HFT is to make a profit by successfully front running the broker’s orders.

#### 3.5.5.1 State

In understanding the behavior of the HFT agents, it is important to understand what kind of information they store and collect. Table 3.7 shows the state variables of each HFT agent:

#### 3.5.5.2 Behaviour

At each time-step, the HFT agent checks for any trades being executed on its local exchange. Just before Milestone 2, HFT1 receives a trade execution broadcast from BYX1 describing a 400 share trade execution.

Upon learning this, HFT1 then figures out at what time-step the order was sent. Upon finding out when the order was sent, it then uses its knowledge of delays to figure out what LOB was seen by the broker at the time the order was sent. If this is done successfully, the HFT will arrive at the conclusion that the broker saw the order books at Milestone 1 when it sent the order.

Using information about the LOB and the number of remaining shares, HFT1 figures out where the next order headed. If HFT1 finds that the next order is going to BYX2 (ie. The next best price is at BYX2), it sends the following debug message to its counterpart HFT2:

“Y, Executed Shares 400. Best Local Ask Price 24.10”
Chapter 3. Experiment 1: Single Broker, Non-Routable Orders

Experiment 1: Single Broker, Non-Routable Orders

Table 3.7: HFT1/2 State For Experiment 1

<table>
<thead>
<tr>
<th>State Variable/Function</th>
<th>Haskell Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1Bids</td>
<td>Map.Map Int BYXOrderlist</td>
<td>The list of bids on BYX1 received by the HFT agent at a given time-step. Stored according to time step received.</td>
</tr>
<tr>
<td>BYX1Asks</td>
<td>Map.Map Int BYXOrderlist</td>
<td>List of asks on BYX1 mapped by time step received.</td>
</tr>
<tr>
<td>BYX2Bids</td>
<td>Map.Map Int BYXOrderlist</td>
<td>List of bids on BYX2 mapped by time step received.</td>
</tr>
<tr>
<td>BYX2Asks</td>
<td>Map.Map Int BYXOrderlist</td>
<td>List of asks on BYX2 mapped by time step received.</td>
</tr>
<tr>
<td>TradeExecutions</td>
<td>Map.Map Int (Price, Shares, AgentSimID, AgentSimID)</td>
<td>List of trade executions on collated exchange (i.e. BYX1 for HFT1, BYX2 for HFT2). Mapped by time step received.</td>
</tr>
<tr>
<td>BrokerOrdSize</td>
<td>Int</td>
<td>According to assumption 3, each HFT agent knows the total order size being executed by the Broker. The BrokerOrdSize is the state variable that stores the number of shares yet to be executed.</td>
</tr>
<tr>
<td>Delay Function</td>
<td>Int -&gt; Int -&gt; Int</td>
<td>According to assumption 4 the HFT has access to all the delays between (to and from) itself, the broker and both exchange. This function returns integer the delay between any two agents given the agent IDs.</td>
</tr>
</tbody>
</table>

The ‘Y’ represents that there is a front-running opportunity. The number of executed shares are the number of shares executed at BYX1. The best local ask price is the next best price at BYX1, which is the case at Milestone 2.

At each time-step, both HFT agents also checks for messages from their counterpart. Upon receiving the above message, HFT2 then confirms the front running opportunity and executes both front running orders. When front-running a buy order, the aim is to increase the best offer on the exchange by sending a market buy order for the current best price and a limit sell order at a higher price. However, since the US equities market is governed by RegNMS, HFT2 cannot increase the price above the National Best Offer otherwise the order will simply be re-routed (if GTC) or cancelled (if IOC), causing the front run to fail. Therefore, the HFT agent is programmed to increase the price to $0.01 below the best offer/ask price at the other exchange. At Milestone 2, the best ask at BYX1 is $24.10, the broker’s next order is a 200 share market buy order to BYX2, which has a best ask of $24.05. It is the goal of HFT2 to raise the best asking price at BYX2 while keeping it less than that of BYX1. As such, HFT2 submits a sell limit order at price $24.09 to BYX2 ($0.01 less than the best ask at BYX1).

Table 3.8 shows the front running orders sent by the HFT agent in the scenario of a successful front run.

<table>
<thead>
<tr>
<th>Sent By</th>
<th>Sent to (Exchange)</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFT2</td>
<td>BYX2</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>HFT2</td>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.09</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

Table 3.8: HFT Agent - Orders Sent
After these orders are submitted and the first market order is executed, we arrive at Milestone 3 and the LOB on each exchange is shown in Figure 3.9. The empty yellow cell illustrates the absence of the 200 shares at $24.05 which was bought by HFT2 from Probe. The blue cell represents the sell limit order placed by the HFT. Note that although the best ask price at BYX2 has been raised, BYX2 still has the National Best Offer (at $24.09) for this given security.

<table>
<thead>
<tr>
<th>Milestone 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BYX1</strong></td>
</tr>
<tr>
<td>Bid Size</td>
</tr>
<tr>
<td>24.10</td>
</tr>
<tr>
<td>24.09</td>
</tr>
<tr>
<td>24.05</td>
</tr>
<tr>
<td>24.00</td>
</tr>
<tr>
<td>23.99</td>
</tr>
<tr>
<td>23.98</td>
</tr>
<tr>
<td>23.97</td>
</tr>
<tr>
<td>23.96</td>
</tr>
<tr>
<td><strong>200</strong></td>
</tr>
<tr>
<td>23.90</td>
</tr>
<tr>
<td>23.85</td>
</tr>
</tbody>
</table>

Table 3.9: Limit Order Book of BYX1 and BYX2 at Milestone 3 in timeline of Figure 3.2

Immediately after milestone 3, the broker’s market buy order for 200 shares finally arrives at BYX2. It is immediately matched and executed against the 200 shares at $24.09. When the Broker first sent the order to BYX2 (at Milestone 1), the broker was expecting its trade to execute against the probe, who supplied the best ask of $24.05. Nevertheless, HFT1 and HFT2 used their knowledge of delays, order sizes and their faster connection to raise the best asking price at BYX2. When the broker’s order finally arrives at BYX2 it executed against HFT2’s limit order at $24.09.

In the scenario of a successful front-run, the HFT makes a profit. HFT2 first buys 200 shares at $24.05 then sells the same 200 shares for $24.09 resulting in a profit of $8.00. While that may not seem like a huge profit, in reality such trades would be automated and repeated throughout the day.

In the broker’s point of view, the price seems to move immediately as the order is sent. This parallels what Brad Katsuyama claims to have experienced on numerous trades in 2008 and 2009. While some anecdotal evidence is provided in Flash Boys; Michael Lewis’s accusation that High Frequency Traders are conducting large scale front-running operations are yet unverified.

Figure 3.3 is a pseudocode representation of the internal logic of HFT1 and HFT2.
3.6 Experiment Hypotheses

3.6.1 Null Hypothesis

The following is a null hypothesis that we aim to disprove. The hypothesis is subject to the assumptions specified at the beginning of the experiment.

**Null Hypothesis** – The HFT will make a profit regardless of any latency (delay) conditions between the agents.

Our experiment aims to illustrate that specific delay conditions are necessary within an agent based system to allow for successful front-running to occur. As such, the results of this experiment should dispel the notion that the HFT can always make a profit.

According to the null hypothesis, there should be no minimum difference necessary in the delay between the broker and exchange BYX1 and the broker and exchange BYX2 for the HFT to make a profit from front-running.

In our experiment, the HFT aims to front run the second order sent by the broker. As we have seen from the explanation of its behaviour, the HFT only begins the process of front-running after it has learnt about the broker’s first trade execution. Additionally, the success of the front-running attempt is dependent on the HFT’s orders arriving at the exchange before the broker’s second order. Therefore, there needs to be some difference in the delays between each exchange and the broker to give time for the HFT’s orders to be executed.
3.6.2 Alternative Hypothesis

Using our knowledge of the experiment setup and the agent behaviours, we can hypothesize the logical conditions under which the HFT will make a profit through front running. These logical conditions can be represented as an inequality of the delays (in time-steps) between the agents.

**Alternative hypothesis** – The HFT’s front-running attempt is successful if the following inequality is fulfilled:

\[(\delta_3 + 1) + (\delta_{5b} + 1) + (\delta_7 + 1) + (\delta_6 + 1) + (\delta_9 + 1) + (\delta_{9b} + 1) \leq (\delta_4 + 1)\]

Table 3.10 shows the breakdown of the components in the inequality above.

<table>
<thead>
<tr>
<th>Time-steps for front-running to occur</th>
<th>Time-steps for Broker’s second order to arrive</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\delta_3 + 1)</td>
<td>(\delta_{5b} + 1)</td>
</tr>
<tr>
<td>(\delta_7 + 1)</td>
<td>(\delta_6 + 1)</td>
</tr>
<tr>
<td>(\delta_9 + 1)</td>
<td>(\delta_{9b} + 1)</td>
</tr>
<tr>
<td>Broker’s first order executed at BYX1</td>
<td>BYX1 Trade execution broadcast received by HFT1</td>
</tr>
<tr>
<td></td>
<td>Notification message from HFT1 received by HFT2</td>
</tr>
<tr>
<td></td>
<td>HFT2 Front-running orders executed at BYX2</td>
</tr>
<tr>
<td></td>
<td>BYX2 sends updated BBO to CQS and receives updated NBBO from CQS</td>
</tr>
<tr>
<td></td>
<td>Broker’s second order executed at BYX2</td>
</tr>
</tbody>
</table>

**Table 3.10**: Breakdown of components in delay inequality for HFT’s profit-making conditions in Experiment 1

The following should be noted about this delay inequality:

- The delays in the inequality refer to the time-step delays between agents shown in the delay matrix in table 3.2

- In the simulator, the delays are implemented by postponing the sending of a given message by a certain number of time steps. After the postponing, the message is sent and arrives at the recipient at the following time-step. In order to account for the time that the message is received, each delay in the above inequality is incremented by 1.

- For the front-running to be successful BYX2 needs send its updated BBO to the CQS and receive its updated NBBO after the front-running orders are received. If the broker’s second order arrives before BYX2 receives the updated NBBO, it will still assume that the best offer is $24.05 instead of $24.09 and cancel the broker’s order. Therefore, the inequality for the HFT to make a profit also depends on the delays to and from BYX2 and CQS.
3.7 Resolving Hypotheses

3.7.1 Results

Table 3.11 showing the results of running Experiment using certain delay combinations. These delay combinations were integral in resolving the hypotheses and understanding the nature of the HFT’s profitability.

<table>
<thead>
<tr>
<th>Delay Adjustment</th>
<th>Delays (Time-steps)</th>
<th>Delay Difference (broker to each exchange)</th>
<th>Alternative Hypothesis Fulfilled</th>
<th>HFT makes profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increasing $\delta_4 - \delta_3$</td>
<td>$0$ $0$ $0$ $0$ $0$ $0$ $4$</td>
<td>$4$</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>Increasing $\delta_3$</td>
<td>$1$ $0$ $0$ $0$ $0$ $0$ $6$</td>
<td>$6$</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>Increasing $\delta_4$</td>
<td>$2$ $1$ $1$ $1$ $1$ $11$</td>
<td>$10$</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>Decreasing $\delta_{5b}$</td>
<td>$2$ $1$ $1$ $1$ $1$ $12$</td>
<td>$10$</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>Increasing $\delta_6$</td>
<td>$6$ $0$ $0$ $5$ $0$ $0$ $16$</td>
<td>$10$</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Table 3.11: Results of certain trials of Experiment 1

3.7.2 Disproving Null Hypothesis

Null Hypothesis – The HFT will make a profit regardless of any latency (delay) conditions between the agents

From the explanation of the front-running process, it is clear that a successful front-run requires a sequence of events to occur. Firstly, the broker’s first order needs to be executed at BYX1. Next, HFT2 needs to receive confirmation of a front-running opportunity. Then HFT2 needs to send its front-running orders to BYX2 wait for the NBBO to be updated. The HFT only makes a profit if all this happens before the broker’s second order arrives at BYX2. Therefore, it is clear that the HFT will not make a profit under all delay permutations. This is further justified by table 3.11 which shows that certain delay combinations restrict the HFT from making a profit.

The null hypothesis implies that there is no minimum difference necessary in the delay between the broker BYX and the broker and BYX2 for the HFT to make a profit from front-running.

First let’s assume this is correct. This would mean that the HFT can make a profit regardless of the difference in the delay between broker and BYX1 and the broker and BYX2.
In our delay matrix in figure 3.2, \( \delta_3 \) refers to the time taken for an order to go from the broker to BYX1 and \( \delta_4 \) refers to the time taken for an order to go from the broker to BYX2. Therefore, \( \delta_4 - \delta_3 \) refers to the difference in delays between the broker and each exchange. If we assume that \( \delta_4 - \delta_3 = 0 \) then the broker has the same delay to both exchanges. For the front-running to be successful, the HFT needs to complete the front-running maneuver before the broker’s second order reaches BYX2.

In the scenario where \( \delta_4 - \delta_3 = 0 \), after BYX1 receives the broker’s first order, the HFT needs to complete the rest of the front-running actions in 0 time-step. This includes receiving the trade execution from BYX1 \((\delta_{5b} + 1)\), sending the debug message with front-running instructions \((\delta_7 + 1)\), sending and executing the front-running orders \((\delta_6 + 1)\) and waiting for BYX2 to send and receive the updated NBBO \([(\delta_9 + 1) + (\delta_{9b} + 1)]\). Even if each of the following delays \{\( \delta_{5b}, \delta_7, \delta_6, \delta_9, \delta_{9b} \}\} were set to 0, the front-running process would still take:

\[
(\delta_{5b} + 1) + (\delta_7 + 1) + (\delta_6 + 1) + (\delta_9 + 1) + (\delta_{9b} + 1) = 1 + 1 + 1 + 1 + 1 = 5 \text{ time-steps.}
\]

Table 3.11 supports this analysis, showing that when all other delays are 0, as the difference in the broker’s delays to each exchange \( \delta_4 - \delta_3 \) increases beyond 4, the HFT stops making a profit.

After the broker’s first order is executed at BYX1, the HFT’s front-running processes take a minimum of 5 time-steps to complete and need to be done before the broker’s second order arrives at BYX2 for the HFT to make a profit.

Therefore, it is clear that there needs to be some minimum difference in the delay between broker-BYX1 and broker-BYX2 for the HFT to make a profit. The minimum value for this delay is \((\delta_4 - \delta_3)_{\text{min}} = 5\). This can be shown as follows:

\[
(\delta_{3} + 1) + 5 \leq (\delta_{4} + 1) \\
\delta_{4} - \delta_{3} \geq 5
\]

Since the delay between the broker and each exchange needs to be a minimum of 5 time-steps, the null hypothesis is disproven.

3.7.3 Justifying Alternative Hypothesis

**Alternative hypothesis** – The HFT’s front-running attempt is successful if the following inequality is fulfilled:

\[
(\delta_{3} + 1) + (\delta_{5b} + 1) + (\delta_{7} + 1) + (\delta_{6} + 1) + (\delta_{9} + 1) + (\delta_{9b} + 1) \leq (\delta_{4} + 1)
\]

In disproving the null hypothesis, we illustrated that the HFT will not make a profit in all delay scenarios. We found that the difference in delays between Broker-BYX1 and Broker-BYX2 needs to be greater than a certain minimum value for the HFT to make a profit. We found that this minimum value depends on the number of time-steps necessary for the front-running to take place.
We calculated the necessary number of time-steps by splitting up the front-running process into five steps. The number of time-steps taken for each step is shown below:

1. \((\delta_3 + 1)\) = Broker’s first order is arrives and is executed at BYX1
2. \((\delta_{5b} + 1)\) = HFT1 receives broker’s first order trade execution broadcast from BYX1
3. \((\delta_7 + 1)\) = HFT2 receives debug message from HFT1 outlining front-running opportunity
4. \((\delta_6 + 1)\) = BYX2 receives and executes/acknowledges both front-running orders from HFT2
5. \((\delta_9 + 1) + (\delta_{9b} + 1)\) = BYX2 sends updated BBO to CQS and receives updated NBBO from CQS

The total time needed for the front-running manoeuvre is the sum of the time taken for the above steps. For the front-running attempt to be successful, this total needs to be less than or equal to the time taken for the broker’s second to arrive and be executed at BYX2. The number of time steps taken for an order to go from Broker to BYX2 and then be executed is \((\delta_4 + 1)\). Therefore, the HFT only make a profit if the total time taken for the front-running manoeuvre does not exceed \((\delta_4 + 1)\). This condition is reflected in the inequality stated in the alternative hypothesis.

Table 3.11 illustrates the systematic adjustment of delays performed in order to arrive at this conclusion. The results clearly illustrate that each individual delay component in the inequality has an impact on the overall success of the front-running manoeuvre.

### 3.8 Limitations of Experiment

Although this simulation has been designed to replicate a realistic front-running scenario, there are some limitations that diminish the reliability of these results.

**Limitations of the set-up**

The number of agents present in the market

- Our simulation assumed that there are only two exchanges (BYX1 and BYX2) that have a market for this given security. In reality, a broker will have many exchanges (NYSE, NASDAQ, BATS) at which to execute a market order. Having only two exchanges makes it much easier for the HFT to predict where the next order will go. In reality, a HFT will have many more exchanges to consider and will need to be co-located at each of them.

- This experiment assumes there is only one broker placing market orders for this specific security. In reality, there will undoubtedly be many other market participants placing orders at the same exchanges. Having multiple brokers will make it more challenging to recognize the broker’s trade and front run it effectively.
Lastly, this experiment set up assumes only one HFT which is unlikely in reality. The presence of multiple HFTs collocated at each exchange makes it more complicated for each of them to front-run successfully. Each HFT can expect to run into scenarios where they end up being the victim of front-running by another HFT.

Limitations of the HFT

The HFT in this experiment is able to identify certain front-running scenarios but is limited in its abilities.

- Firstly, the HFT assumes that the broker sends both market orders at the same time step. If the broker decides to delay one of the orders, this may cause a confusion for the HFT. If the broker decides to send its order to BYX2 first and then BYX1, both orders may arrive at very similar times and cause the HFT to send its front-running orders late.

- The HFT assumes that both orders are being sent to separate exchanges. While this may be reasonable, the RegNMS system gives the broker the opportunity to mark its orders as ( Routable, Good Till Cancelled) and submit them to both to the same exchange. This way, the exchange is obliged to route the broker’s orders to the destination with the best price. It may be that routing the order to BYX2 through BYX1 is quicker for the broker than submitting the order directly to BYX2. In this case, the HFT may be too late in submitting its front-running orders as it is not calibrated to consider the time delays of routable orders.

- Additionally, the other assumptions made about the HFT are also quite unrealistic. The first assumption suggests that the HFT knows the broker’s total order size. In reality, this is highly unrealistic is unlikely to be available by legal means. Furthermore, the assumption that the HFT knows the delays between the broker and each exchange is also unrealistic. Despite unsubstantiated claims in Flash Boys, there is little evidence of an HFT having access to accurate latency times between broker and each exchange.

3.9 Conclusion

This experiment shows that even in a high simplified simulation within limited agents and generous assumption, a HFT is not guaranteed to make a profit from front-running. The results demonstrate that the HFT’s profit making ability are highly dependent on certain factors that it cannot reliably know or control. Such factors include the delay between the broker and each exchange ($\delta_3, \delta_4$) and the delay between the exchange and CQS ($\delta_9, \delta_{9b}$). As such, the experiment illustrates that the HFT is operating on incomplete information. This incomplete information causes the HFT to risk having their orders arrive too late, leading them to miss the front-running opportunity. A missed front-running opportunity results in the HFT taking on a unplanned market positions and risking prices shifting in the opposite direction.
Chapter 4

Experiment 2: Single Broker, Routable Orders

4.1 Experiment Overview

The objective of this experiment is to investigate whether, and if so under what conditions and with what risk is it, possible, for a HFT to make a profit by front running a routable order from a single traditional broker in a RegNMS market system.

Experiment 2 is an extension of the previous experiment (experiment 1, see Chapter 3) which explores the specific case where a high frequency trader (HFT) is trying to front run a routed order.

In the previous experiment, the order protection rule, one of the central features of a RegNMS market system, is disregarded. The order protection rule is a provision of RegNMS that ensures that investors receive an execution price that is equivalent or better than what is being quoted on any other exchange. It requires all exchanges to route orders to the exchange with the best execution price if they are not able to match the price.

In Experiment 2, a broker submits a single Good Till Cancelled (GTC) market order to one exchange. A traditional broker may only have the access to one exchange. Nevertheless, the broker expects the exchange to split and route the order such that they receive the best execution price for each share. If the broker’s GTC order cannot be executed in one batch, it will need to be split and a portion of the order will be routed. The HFT will aim to use its fast access to each exchange, knowledge of trade executions and communication delays to submit its own orders and shift the price at the routed (destination) exchange. If the HFT is successful, the routed portion will execute at a higher price than seen by the broker and the HFT will make a profit from the difference.
4.2 Agents Overview

While their behaviours may be different, the agents in this experiment are the same as those in Experiment 1.

For a full explanation of their roles, refer to section 3.3.

4.2.1 Agent Communication Topology

Figure 5.1 is a communication topology diagram illustrating the connection and delays in communications between the agents in this experiment.
4.2.2 Agent Delay Matrix

Table 4.1 is a matrix representation of the delays in communication (measured in abstract units of time) between the various agents in the experiment. Note that only connections relevant to this experiment are shown.

<table>
<thead>
<tr>
<th>From</th>
<th>BYX1</th>
<th>BYX2</th>
<th>CQS</th>
<th>Probe</th>
<th>Broker</th>
<th>HFT1</th>
<th>HFT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1</td>
<td>-</td>
<td>δ10</td>
<td>δ8</td>
<td>δ1b</td>
<td>δ3b</td>
<td>δ5b</td>
<td>-</td>
</tr>
<tr>
<td>BYX2</td>
<td>δ10b</td>
<td>-</td>
<td>δ9</td>
<td>δ2b</td>
<td>δ4b</td>
<td>-</td>
<td>δ6b</td>
</tr>
<tr>
<td>CQS</td>
<td>δ8b</td>
<td>δ9b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Probe</td>
<td>δ1</td>
<td>δ2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Broker</td>
<td>δ3</td>
<td>δ4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HFT1</td>
<td>δ5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>δ7</td>
<td>-</td>
</tr>
<tr>
<td>HFT2</td>
<td>-</td>
<td>δ6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.1: Agent Delay Matrix for Experiment 2

4.3 Assumptions

The assumptions made for experiment 2 are identical to those made for the previous simulation (experiment 1). Only one new assumption is made.

Assumption 5 – The broker is only able to execute trades on BYX1

This experiment investigates a scenario of a small scale broker who can only afford trading access to one exchange (BYX1). However, knowing about the order protection rule in RegNMS, the broker expects his orders to be split and routed so to execute at the best available price in the market.

4.4 Agent Behaviour

This section will explain the role and behaviour of all of the agents in the simulation. The explanation will refer to the a swim lanes diagram which illustrates the messages sent and actions taken by each agent in the simulation. Figure 4.2 reflects the timeline of actions in the event where the front running effort is successful (i.e. the HFT makes a profit).
Figure 4.2: Swim Lanes Diagram of Successful Front-running outcome of Experiment 2.
Chapter 4. *Experiment 2: Single Broker, Routable Orders*  

### Exchanges

BYX1 and BYX2 are identical to each other and behave in the same manner as in experiment 1. For a full explanation of their Behavior types of messages sent and received by the exchanges, refer to section 3.5.1.

### Consolidated Quotation System (CQS)

The Consolidated Quotation System is identical to the CQS agent in experiment 1. A further explanation of its behavior can be found in section 3.5.2.

#### Probe - ProbeV1.hs (Appendix Section B.3)

The probe agent in this experiment is responsible for populating the limit order books of each exchange.

Table 4.2 shows the orders sent by the probe agent to the exchanges

<table>
<thead>
<tr>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>23.95</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>400</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.00</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>600</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.10</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>23.95</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.05</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

**Table 4.2: Probe Agent - Orders Sent**

Table 4.3 shows the limit order book of BYX1 and BYX2 at Milestone 1 when all of the probe’s limit orders have been received and placed on the LOB.

<table>
<thead>
<tr>
<th>Milestone 1</th>
<th>BYX1</th>
<th></th>
<th>BYX2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid Size</td>
<td>Price</td>
<td>Ask Size</td>
<td>Bid Size</td>
<td>Price</td>
</tr>
<tr>
<td>24.10</td>
<td>600</td>
<td></td>
<td>24.10</td>
<td></td>
</tr>
<tr>
<td>24.09</td>
<td></td>
<td></td>
<td>24.09</td>
<td></td>
</tr>
<tr>
<td>24.05</td>
<td></td>
<td></td>
<td>24.05</td>
<td>200</td>
</tr>
<tr>
<td>24.00</td>
<td>-400</td>
<td></td>
<td>24.00</td>
<td></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td></td>
<td>23.99</td>
<td></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td></td>
<td>23.98</td>
<td></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td></td>
<td>23.97</td>
<td></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td></td>
<td>23.96</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td></td>
<td>200</td>
<td>23.95</td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
<td>23.90</td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
<td>23.85</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.3: Limit Order Book of BYX1 and BYX2 at Milestone 1 in timeline of Figure 3.2**

#### 4.4.1 Broker - BrokerV1a.hs (Appendix Section B.6)

In this experiment, we assume that the broker is instructed by a client to purchase 600 shares of the given security at market price. The broker inspects the limit order book at each exchange and notes the following:
Chapter 4. Experiment 2: Single Broker, Routable Orders

- The best offer/ask for this specific security can be found on BYX1 where there are 400 shares available at $24.00.
- The second best offer/ask is found on BYX2 where 200 shares are available for $24.05
- RegNMS guarantees the national best execution price for any orders submitted

As such, the broker sends a 600 share market purchase order to BYX1\(^1\). Table 4.4 shows the market orders sent by the broker.

<table>
<thead>
<tr>
<th>Order #</th>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>BYX1</td>
<td>Market</td>
<td>600</td>
<td>BUY</td>
<td>TRUE</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

Table 4.4: Broker Agent - Order Sent

The main difference in this experiment is that both the orders are marked as routable, Good Till Cancelled and both are sent to BYX1. From inspecting the LOBs, the broker expects the order to be split into two portions. The broker expects that 400 shares (portion #1) will be executed against the national best offer of 400 shares at $24.00 in BYX1. Since the next best ask is 200 shares offered at $24.05 in BYX2, the broker then expects the order 200 shares (portion #2) to be routed from BYX1 to BYX2.

Table 4.5 shows how the broker’s original order is split into execution portions.

<table>
<thead>
<tr>
<th>Portion #</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Execution Destination</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Market</td>
<td>400</td>
<td>BUY</td>
<td>BYX1</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>#2</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>BYX2 (Routed)</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

Table 4.5: Broker Agent - Order Split (portions)

The swim lanes diagram (Figure 4.2) shows that portion #1 is executed at BYX1 just before Milestone 2. At this same point, portion #2 (the 200 share routed order) is sent from BYX1 but has yet to arrive at BYX2.

Table 4.6 shows the LOB of both exchanges after the execution of the broker’s first portion (Milestone 2 in Figure 4.2). The empty yellow cell illustrates the absence of the 400 shares on the LOB of BYX1 that were just executed.

4.4.2 High Frequency Trader - HFTV1.hs (Appendix Section B.5)

The HFT in this experiment is identical in behaviour and goal to the HFT agent in experiment 1. Refer to section 3.5.5 for details.

At each time-step, the HFT agent checks for any trades being executed on its local exchange. Just before milestone 2, HFT1 receives a trade execution broadcast from BYX1 describing the 400 share trade that

\(^1\)The current version of the simulation does not permit splitting orders. As such, in the code, the broker’s order is sent as three 200 share orders. This does not impact the interpretation of the experiment and is simply an implementation detail.
just occurred between the broker and probe agents. HFT1 then does some internal calculations to figure out whether there is a front running opportunity and where portion #2 is headed.

HFT1 suspects that portion #2 is being routed to BYX2. As such, HFT1 sends a message to HFT2 telling it of the front-running opportunity. Upon receiving this message, HFT2 then sends the following front-running orders (shown in Table 4.7) to BYX2.

<table>
<thead>
<tr>
<th>Sent By</th>
<th>Sent to (Exchange)</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFT2</td>
<td>BYX2</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>HFT2</td>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.09</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

Table 4.7: HFT Agent - Orders Sent

HFT2’s market order removes the 200 shares offered by the probe at $24.05 from the LOB of BYX2. The next limit order then adds another 200 shares to BYX2’s LOB at $24.09. This also updates the national best offer to $24.09. A successful front-run occurs when both these orders arrive at BYX2 before the broker’s routed portion #2 arrives at BYX2. In this case, the broker’s portion #2 will execute at $24.09, a higher price than the broker expected. In reality, this entire process would occur in a matter of seconds and in the broker’s perspective, the price would seem to move immediately as the order is sent. At Milestone 3, HFT2’s orders would have arrived at BYX2 and the market order would be executed. Figure 4.8 shows what the order books would look like at Milestone 3.

The empty yellow cell illustrates the absence of the 200 shares at $24.05 which was bought by HFT2 from Probe. The blue cell represents the sell limit order placed on the LOB by the HFT. It should be noted that although the best ask price at BYX2 has been raised, BYX2 still has the National Best Offer (at $24.09) for this given security.
4.5 Experiment Hypotheses

4.5.1 Null Hypothesis

The following null hypothesis is a statement that we aim to disprove.

Null Hypothesis – The HFT will make a profit regardless of any latency (delay) conditions between the agents

Our experiment aims to illustrate that specific delay conditions are necessary within an agent based system to allow for successful front-running to occur. As such, the results of this experiment should dispel the notion that the HFT can always make a profit.

This experiment will be specifically trying to disprove the notion that there is no minimum delay needed between the two exchanges (BYX1 and BYX2) for the HFT to make a profit from front-running.

In our experiment, the HFT aims to front run the routed portion (portion #2) of the order originally sent by the broker. The success of the front-running attempt is dependent on the HFT’s orders arriving at BYX2 before the broker’s portion #2 can be routed from BYX1 to BYX2. Logically, there needs to be some minimum delay between BYX1 and BYX2 to give time for the HFT’s front-running orders to be executed.

4.5.2 Alternative Hypothesis

Our alternative hypothesis is an inequality of the delays (in time-steps) between the agents that, if fulfilled, will allow the HFT to make a profit through front running.

---

Table 4.8: Limit Order Book of BYX1 and BYX2 at Milestone 3 in timeline of Figure 4.2

<table>
<thead>
<tr>
<th>Bid Size</th>
<th>Price</th>
<th>Ask Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.10</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>24.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td></td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bid</strong></td>
<td><strong>Price</strong></td>
<td><strong>Ask Size</strong></td>
</tr>
<tr>
<td>24.10</td>
<td><strong>200</strong></td>
<td></td>
</tr>
<tr>
<td>24.09</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>24.05</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td><strong>200</strong></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td></td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
Alternative hypothesis – The HFT’s front-running attempt is successful if the following inequality is fulfilled:

$$(\delta_3 + 1) + (\delta_{5b} + 1) + (\delta_7 + 1) + (\delta_6 + 1) + (\delta_9 + 1) + (\delta_{9b} + 1) \leq (\delta_3 + 1) + (\delta_{10} + 1)$$

Table 4.9 shows the breakdown of of the components in the inequality above

<table>
<thead>
<tr>
<th>Time-steps for front-running to occur</th>
<th>Time-steps for Broker’s routed order (portion #2 to arrive at BYX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$(\delta_3 + 1)$$</td>
<td>$$(\delta_{10} + 1)$$</td>
</tr>
<tr>
<td>$$(\delta_{5b} + 1)$$</td>
<td>$$(\delta_3 + 1)$$</td>
</tr>
<tr>
<td>$$(\delta_7 + 1)$$</td>
<td>$$(\delta_{5b} + 1)$$</td>
</tr>
<tr>
<td>$$(\delta_6 + 1)$$</td>
<td>$$(\delta_3 + 1)$$</td>
</tr>
<tr>
<td>$$(\delta_9 + 1)$$</td>
<td>$$\leq (\delta_3 + 1)$$</td>
</tr>
<tr>
<td>$$(\delta_{9b} + 1)$$</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.9: Breakdown of components in delay inequality for HFT’s profit-making conditions in Experiment 2

Since the term $$(\delta_3 + 1)$$ is common to both sides of the inequality, it can be removed to simplify the inequality like so:

$$(\delta_{5b} + 1) + (\delta_7 + 1) + (\delta_6 + 1) + (\delta_9 + 1) + (\delta_{9b} + 1) \leq (\delta_{10} + 1)$$

4.6 Resolving Hypotheses

4.6.1 Results

Table 4.10 showing the results of running Experiment using certain delay combinations. These delay combinations were integral in resolving the hypotheses and understanding the nature of the HFT’s profitability.

4.6.2 Disproving Null Hypothesis

Null Hypothesis – The HFT will make a profit regardless of any latency (delay) conditions between the agents

From the explanation of the front-running process, it is clear that a successful front-run requires a sequence of events to occur. Firstly, portion #1 needs to be executed at BYX1. Next, HFT2 needs to receive confirmation of a front-running opportunity. Then HFT2 needs to send its front-running orders to BYX2 wait for the NBBO to be updated. The HFT only makes a profit if all this happens before the portion #2 is routed and arrives at BYX2. Therefore, it is clear that the HFT will not make a
profit under all delay permutations. This is further justified by Table 4.10 which shows that certain delay combinations restrict the HFT from making a profit.

Let’s assume for instance that there is no minimum delay needed between the two exchanges (BYX1 and BYX2) for the HFT to make a profit from front-running. This would mean that the HFT can make a profit regardless of the delay between BYX1 and BYX2. In our delay matrix in Figure 4.1, \( \delta_{10} \) refers to the time taken for an order to be routed from BYX1 to BYX2. If we assume that \( \delta_{10} = 0 \) then the broker’s order can be routed from BYX1 to BYX2 and executed in \( \delta_{10} + 1 = 0 + 1 = 1 \) time-step.

In this scenario, the HFT needs to complete all its front-running actions in 1 time-step. This includes receiving the trade execution from BYX1 \( (\delta_{5b} + 1) \), sending the debug message with front-running instructions \( (\delta_{7} + 1) \), sending and executing the front-running orders \( (\delta_{6} + 1) \) and waiting for BYX2 to send and receive the updated NBBO \( [(\delta_{9} + 1) + (\delta_{9b} + 1)] \). Even if each of the following delays \( \{\delta_{5b}, \delta_{7}, \delta_{6}, \delta_{9}, \delta_{9b}\} \) were set to 0, the front-running process would still take \( (\delta_{5b} + 1) + (\delta_{7} + 1) + (\delta_{6} + 1) + (\delta_{9} + 1) + (\delta_{9b} + 1) = 1 + 1 + 1 + 1 + 1 = 5 \) time-steps.

The HFT’s front-running processes take a minimum of 5 time-steps to complete and need to be done before the portion #2 is routed from BYX1 to BYX2 for the HFT to make a profit.
Chapter 4. Experiment 2: Single Broker, Routable Orders

Therefore it is clear that there needs to be some minimum delay between BYX1 and BYX2 for the HFT to make a profit. The minimum value for this delay is $\delta_{10} = 4$. This can be shown as follows:

$$5 \leq \delta_{10_{\text{min}}} + 1$$
$$\delta_{10_{\text{min}}} \geq 4$$

Table 4.10 supports this analysis, showing that when all other delays are 0, as the routing delay between the exchanges $\delta_{10}$ decreases below 4, the HFT stops making a profit.

4.6.3 Justifying Alternative Hypothesis

**Alternative hypothesis** – The HFT’s front-running attempt is successful if the following inequality is fulfilled:

$$(\delta_{5b} + 1) + (\delta_{7} + 1) + (\delta_{6} + 1) + (\delta_{9} + 1) + (\delta_{9b} + 1) \leq (\delta_{10} + 1)$$

In disproving the null hypothesis, we illustrated that the HFT will not make a profit in all delay scenarios. We found that the delay between BYX1 and BYX2 ($\delta_{10}$) needs to be greater than a certain minimum value for the HFT to make a profit. We found that this minimum value depends on the number of time-steps necessary for the front-running to take place.

We calculated the necessary number of timesteps by splitting up the front-running process into four steps. The number of timesteps taken for each step is shown below:

1. $(\delta_{5b} + 1) =$ HFT1 receives trade execution broadcast of portion #1 from BYX1
2. $(\delta_{7} + 1) =$ HFT2 receives debug message from HFT1 outlining front-running opportunity
3. $(\delta_{6} + 1) =$ BYX2 receives and executes/acknowledges both front-running orders from HFT2
4. $(\delta_{9} + 1) + (\delta_{9b} + 1) =$ BYX2 sends updated BBO to CQS and receives updated NBBO from CQS

The total time needed for the front-running manoeuvre is the sum of the time taken for the above steps. For the front-running attempt to be successful, this total needs to be less than or equal to the time taken for the portion #2 to be routed. The number of time steps taken for an order to be routed from BYX1 to BYX2 and then executed is $(\delta_{10} + 1)$. Therefore, the HFT only make a profit if the total time taken for the front-running manoeuvre does not exceed $(\delta_{10} + 1)$.

This condition is reflected in the inequality stated in the alternative hypothesis above. Table 4.10 illustrates the systematic adjustment of delays and repeated simulation used to verify the alternative hypothesis. The results clearly illustrate that each individual delay component of the inequality has in impact on the overall success of the front-running manoeuvre.
4.7 Limitations of Experiment

Although this simulation has been designed to replicate a realistic front-running scenario, there are some limitations that diminish the reliability of these results.

Limitations of the set-up

The number of agents present in the market

- Our simulation assumed that there are only two exchanges (BYX1 and BYX2) that have a market for this given security. This means that there is only one choice for the exchange to route its order, making it easier for the HFT to predict the destination of the routed order.

- This experiment also assumes there is only one broker in this market. In reality, there will undoubtedly be multiple brokers placing orders at the same exchanges. Having multiple brokers will make it more challenging to recognize the brokers trade and front run it effectively.

Limitations of the HFT

- Firstly, the HFTs behaviour is based on the assumption that the broker sends only one market order. If the broker decides split and send the orders at different times, this will cause a confusion for the HFT.

- The HFT’s strategy only works if we assume that the broker only sends orders to BYX1. While, this is an assumption for our current experiment, it may not be true in reality. If the broker decides to split the order and send them to separate exchanges, the HFT will have more difficulty predicting the routing path of the orders.

4.8 Conclusion

The results of this experiment show even in a high simplified simulation within limited agents and generous assumption, a HFT is not guaranteed to make a profit from front-running. More generally, this experiment demonstrates that the HFTs profit making ability are highly dependent on certain factors that it cannot reliably know or control. Such factors include the delay between the broker and each exchange ($\delta_3, \delta_4$) and the delay between the exchange and CQS ($\delta_9, \delta_{9b}$).

In Flash Boys, Michael Lewis talks about a firm called Spread Networks, that built an ultra-low latency connection between the Chicago Mercantile exchange and the Nasdaq in New Jersey [1]. Lewis uses this as an example of a scenario where HFT’s can pay to have much faster connection between exchanges and use this to front-run routed orders. Nevertheless, despite the speculation, Lewis provides no evidence that it would be possible to use this connection to reliably front-run orders or that any firm are actually engaging in such strategy.
The alternative hypothesis shows that the HFTs profit-making ability rests on a certain inequality being true. The right side of the inequality has the term \((\delta_{10} + 1)\) which represents the time taken for an order to be routed from one exchange to the other and then executed. In reality there are numerous paths the routed order may take, making it impossible for the HFT to realistically predict the delay. As such, each time the HFT decides to front-run, it runs the risk of having its orders arrive too late, resulting in it missing the front-running opportunity and taking on a greater market position. The main point this experiment illustrates is that the HFT is also operating on incomplete information and taking significant risks each time it decides to front-run. The main risk that the HFT takes in this experiment is the chance that the routed order will arrive faster than its own front-running orders. Despite Lewis’s claims, there is no evidence of there being any reliable or legal way of knowing the exact delay between public exchanges in the US equity markets. Without this key piece of information, the HFT cannot be certain that its front-running attempt will be successful, making it impossible to engage in the strategy without significant risk.
Chapter 5

Experiment 3: Multiple Brokers

5.1 Experiment Overview

The objective of this experiment is to investigate whether it is possible, and if so under what conditions and with what risk is it possible, for a HFT to make a profit by simultaneously front running non-routable orders from multiple traditional brokers in a RegNMS market system.

Experiment 3 is another agent based simulation of a front-running attempt within the US secondary equity market. This experiment is an extension of the previous simulations that aims to portray a more realistic market scenario with multiple brokers. Both brokers are tasked with executing trades for their clients and each one sends two orders, one to each exchange, to buy a given security at market price. The HFT aims to use its fast access to each exchange and knowledge of trade execution and communication delays to submit its own orders and shift the price before each broker’s second order is executed. The presence of multiple brokers in the market is expected to make the front-running process more complicated for the HFT.

5.2 Agents Overview

The real-life entities involved in this scenario are represented as agents in our simulation and shown in Table 5.1.

5.2.1 Agent Communication Topology

Figure 5.1 is a communication topology diagram illustrating the connection and delays in communications between the agents in this experiment.
Chapter 5. Experiment 3: Multiple Brokers

### Table 5.1: Agents in Experiment 3

<table>
<thead>
<tr>
<th>Agent Type</th>
<th>Agent Name</th>
<th>Agent Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchanges</td>
<td>BYX1</td>
<td>The stock exchanges in our experiment are representative of any US secondary equities market. The behaviour of these exchanges are designed to resemble the American BYX exchange, which is an SEC approved BATS exchange. The BYX exchange functions within a RegNMS market system and is capable of upholding the NBBO (National Best Bid and Offer) through order routing. The BYX exchange also offers a premium co-location service. BYX1 is the first instance of the BATS BYX and BYX2 is the second instance. BYX2 functions independently of BYX1.</td>
</tr>
<tr>
<td>BYX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CQS (Consolidated Quotation System)</td>
<td>CQS</td>
<td>The CQS agent consolidates the limit order books of all exchanges to provide the best overall price for any traded equity security. This information is known as the National Best Bid and Offer (NBBO) and is used by the individual exchanges to route market orders to the destination with the best price.</td>
</tr>
<tr>
<td>Probe</td>
<td>ProbeV1</td>
<td>The probe agent populates the limit order book for the security on both exchanges by sending bids and asks. This creates a price for the instrument and sets up a market environment that will be the basis for our front running scenario.</td>
</tr>
<tr>
<td>Broker</td>
<td>Broker1</td>
<td>This experiment focuses on the front-running of non-routable, immediate or cancel orders. As such, both brokers will split their total order volume and send market buy orders to each exchange. It is assumed that broker1 is closer to BYX1 and broker2 is closer to BYX2. Both broker 1 and broker 2 are instructed to buy 600 shares of a particular security</td>
</tr>
<tr>
<td></td>
<td>Broker2</td>
<td></td>
</tr>
<tr>
<td>High Frequency Trader</td>
<td>HFT1</td>
<td>As in the previous experiment, the HFT is represented by two agents HFT1 and HFT2. The HFT's aim is to front-run each broker's second order, by raising the execution price and profiting the difference.</td>
</tr>
<tr>
<td></td>
<td>HFT2</td>
<td></td>
</tr>
</tbody>
</table>

### 5.2.2 Agent Delay Matrix

Table 5.2 is a matrix representation of the delays in communication (measured in abstract units of time) between the various agents in the experiment. Note that only connections relevant to this experiment are shown.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>BYX1</th>
<th>BYX2</th>
<th>CQS</th>
<th>Probe</th>
<th>Broker1</th>
<th>Broker2</th>
<th>HFT1</th>
<th>HFT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>δ8b</td>
<td>δ1b</td>
<td>δ3b</td>
<td>δ10b</td>
<td>δ5b</td>
<td>-</td>
</tr>
<tr>
<td>BYX2</td>
<td>-</td>
<td>-</td>
<td>δ9b</td>
<td>-</td>
<td>δ2b</td>
<td>δ4b</td>
<td>δ11b</td>
<td>-</td>
<td>δ6b</td>
</tr>
<tr>
<td>CQS</td>
<td>δ8b</td>
<td>δ9b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Probe</td>
<td>δ1</td>
<td>δ2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Broker1</td>
<td>δ3</td>
<td>δ4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Broker1</td>
<td>δ10</td>
<td>δ11</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HFT1</td>
<td>δ5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>δ7</td>
<td>-</td>
</tr>
<tr>
<td>HFT2</td>
<td>-</td>
<td>-</td>
<td>δ6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>δ7b</td>
<td>-</td>
</tr>
</tbody>
</table>

### 5.3 Assumptions

#### 5.3.1 Market System Assumptions

**Assumption 1** – Only the listed agents exist in this market
Assumption 2 – The trade execution information published by the exchange is anonymous and does not to identify the counter parties

5.3.2 Agent Behaviour Assumptions

Assumption 3 – The HFT knows the total order size being executed by each broker

Assumption 4 – Both HFT1 and HFT2 know the delays between the following agents:

- Each broker and each exchange
- Itself and its co-located exchange
- Itself and the other HFT agent (co-located at the other exchange)
- The CQS and its co-located exchange

Assumption 5 – Broker1 is closer to BYX1 than Broker2, broker2 is closer to BYX2 than broker1. This is known by the HFT agent

Using this assumption, we create LOBs in each exchange such that, broker1’s second order is sent to BYX2 and broker2’s second order is sent to BYX1. Since the HFT is trying to front-run each broker’s
second order, it will need to perform its front-running manoeuvre on both exchanges. This creates an added challenge for the HFT and reflects a more realistic market scenario where there are multiple brokers to front-run across multiple exchanges.

**Assumption 6** – Both brokers send all orders market orders at the same time-step and the HFT agent knows this

There are certain times in the market where the HFT can be confident of a high volume of orders being executed. For example, right after a major earnings surprise or a federal reserve announcement, it is clear that there will be heightened trading activity as the market reacts to this new piece of public information. This is the logic behind this assumption. We assume that an important announcement is being made and that the HFT expects both brokers to react to this announcement, and send orders simultaneously.

Even though trade execution data from the exchanges is anonymized, the HFT can use assumption 4, 5 and 6 to deduce which trade execution refer to which broker. For example, the HFT can reliably know that the first execution it witnesses on BYX1 is the market order sent by broker1 since broker1 is closer to BYX1 than broker2. This allows the HFT to circumvent the challenge posed by assumption 2 and improve its prediction of the size and destinations of upcoming orders.

### 5.4 Agent Behaviour

This section will explain the role and behaviour of all agents and refer to the a swim lanes diagram which illustrates the messages sent and actions taken by each agent in the simulation. Figure 5.2 reflects the timeline of actions in the event where the HFT is successful in front-running each broker’s second order (i.e the HFT makes a profit off both brokers).

To enhance readability, Figure 5.2 has been printed over two pages. Nonetheless, it should be considered as one diagram depicting a single simulation. The first page shows the events from the start of the simulation till Milestone 2 while the second page shows the events from Milestone 2 till the end of the simulation.
Figure 5.2: Swim Lanes Diagram of HFT successfully Front-running both brokers in Experiment 3. (Till Milestone 2)
Figure 5.2: Swim Lanes Diagram of HFT successfully Front-running both brokers in Experiment 3. (Milestone 2 Onwards)
Exchanges

BYX1 and BYX2 behave in the same manner as in previous experiments. For a full explanation refer to section 3.5.1.

Consolidated Quotation System (CQS)

The Consolidated Quotation System is identical to the CQS agent in previous experiments, refer to section 3.5.2.

5.4.1 Probe - ProbeV2.hs (Appendix Section B.7)

The probe agent is responsible for populating the limit order books of each exchange.

Table 5.3 shows the orders sent by the probe agent to the exchanges

<table>
<thead>
<tr>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>23.95</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>400</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.00</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.05</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX1</td>
<td>Limit</td>
<td>600</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.10</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>23.95</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>400</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.00</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.05</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>BYX2</td>
<td>Limit</td>
<td>600</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.10</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

Table 5.3: Probe Agent - Orders Sent

Table 5.4 shows the limit order book of BYX1 and BYX2 at Milestone 1 when all of the probe’s limit orders have been received and placed on the LOB.

<table>
<thead>
<tr>
<th>Milestone 1</th>
<th>BYX1</th>
<th>BYX2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid Size</td>
<td>Price</td>
<td>Ask Size</td>
</tr>
<tr>
<td>24.10</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>24.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.05</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td>200</td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Limit Order Book of BYX1 and BYX2 at Milestone 1 in timeline of Figure 5.2

5.4.2 Broker agents

Broker1 - BrokerV2.hs (Appendix Section B.8)
Broker1 represents the first broker in this experiment. It is assumed that broker1 is closer to BYX1 than BYX2 and is instructed by a client to purchase 600 shares of the given security at market price. Broker1 begins to take action at Milestone 1.

Broker1 notes that both BYX1 and BYX2 have 400 shares at the NBO price. In order to obtain the best execution price for the orders, Broker1 splits its 600 share purchase into two market orders of 400 and 200 shares. Broker1 sends the larger (400 share) order to BYX1 as it expects a lower delay in execution. Broker1 sends the smaller (200 share) order to BYX2. Broker1 expects both orders to execute at the national best asking price of $24.00. Table 5.5 shows the market orders sent by broker1 directly after Milestone 1.

<table>
<thead>
<tr>
<th>Order #</th>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>BYX1</td>
<td>Market</td>
<td>400</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Immediate Or Cancel</td>
</tr>
<tr>
<td>#2</td>
<td>BYX2</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Immediate Or Cancel</td>
</tr>
</tbody>
</table>

Table 5.5: Broker1 Agent - Orders Sent

Both orders are specified to be non-routable and are marked as Immediate or Cancel thus they should be cancelled if not fulfilled at the national best offering price immediately on arrival.

**Broker2 - BrokerV2.hs (Appendix Section B.8)**

Broker2 represents the second broker in this experiment. Broker2 is closer to BYX2 than BYX1 and is instructed by a client to purchase 600 shares of the given security at market price. Broker2 takes at Milestone 1.

In order to obtain the best execution price for the orders, Broker2 splits its 600 share purchase into two market orders of 400 and 200 shares. Broker2 sends the larger (400 share) order to BYX2 as it expects a lower delay in execution. Broker2 sends the smaller (200 share) order to BYX1. Broker2 expects both to execute at the national best asking price of $24.00. Below are the orders sent by broker2 directly after Milestone 1. Table 5.6 shows the market orders sent by broker2 directly after Milestone 1.

<table>
<thead>
<tr>
<th>Order #</th>
<th>To Exchange</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>BYX2</td>
<td>Market</td>
<td>400</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Immediate Or Cancel</td>
</tr>
<tr>
<td>#2</td>
<td>BYX1</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Immediate Or Cancel</td>
</tr>
</tbody>
</table>

Table 5.6: Broker2 Agent - Orders Sent

Given that broker1 is closer to BYX1 and broker2 is closer to BYX2, it is assumed that for each broker, order #1 will execute earlier than order #2. Broker1’s order #1 executes on BYX1 while broker2’s order #1 executes on BYX2.

After executing broker1’s order #1 against the probe’s limit order (400 shares at $24.00), BYX1 sends out a trade execution broadcast specifying the price and quantity of shares exchanged in the trade. Since HFT1 is co-located at BYX1 and has a level 3 data subscription to BYX1, it is one of the first agents to receive this message.
Similarly, BYX2 sends out a trade execution broadcast after executing broker2’s order #1 against the probe’s limit order (400 shares at $24.00). HFT2 is one of the first to receive the message since it is co-located at BYX2 and has a level 3 data subscription.

Milestone 2 (in figure 5.2) occurs when the each broker’s first order (order #1) has been executed and all trade confirmations, execution broadcast have been sent.

Table 5.7 shows the LOB of both exchanges at Milestone 2.

<table>
<thead>
<tr>
<th>Milestone 2</th>
<th>BYX1</th>
<th></th>
<th>BYX2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid Size</td>
<td>Price</td>
<td>Ask Size</td>
<td>Bid Size</td>
<td>Price</td>
</tr>
<tr>
<td>24.10</td>
<td>600</td>
<td></td>
<td>24.10</td>
<td>600</td>
</tr>
<tr>
<td>24.09</td>
<td></td>
<td></td>
<td>24.09</td>
<td></td>
</tr>
<tr>
<td>24.05</td>
<td>200</td>
<td></td>
<td>24.05</td>
<td>200</td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td></td>
<td>24.00</td>
<td></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td></td>
<td>23.99</td>
<td></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td></td>
<td>23.98</td>
<td></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td></td>
<td>23.97</td>
<td></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td></td>
<td>23.96</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td></td>
<td>200</td>
<td>23.95</td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
<td>23.90</td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
<td>23.85</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.7: Limit Order Book of BYX1 and BYX2 at Milestone 2 in timeline of Figure 5.2

The empty blue cell in the ask book of BYX1 represents the absence of the 400 shares which were bought by broker1. The empty yellow cell in the ask book of BYX2 represents the absence of the 400 shares that bought by broker2. At Milestone 2, the new NBO is $24.05, with 200 shares available on BYX1 and BYX2.

Both broker1 and broker2’s behaviours are defined by the same code base (BrokerV2.hs in the Appendix Section B.8). Apart from the specific orders they send, broker1 and broker2 behave identically. This is due to assumption 6 which states that both brokers send both their orders at the same time-step.

5.4.3 High Frequency Trader - HFTV2.hs (Appendix Section B.9)

5.4.3.1 State

In understanding the behavior of the HFT agents, it is important to understand what kind of information they store and collect. Table 5.8 shows the state variables of each HFT agent:

5.4.3.2 Behaviour

At each time-step, the HFT agent checks for any trades being executed on its local exchange. Given an eminent market announcement, the HFT is expecting trade volume to spike.
Just prior to milestone 2, both HFT agents receive a trade execution broadcast from their respective local exchange. HFT1 receives a broadcast from BYX1 describing 400 shares bought at $24.00. HFT2 receives and identical broadcast from BYX2.

Because of assumption 4, the HFT knows that broker1 is closer to BYX1 while broker2 is closer to BYX2. Assumption 6 states that there is an eminent market announcement that both brokers are expected simultaneously to react to. Thus way, the HFT also knows that both brokers send their market orders at the same time-step. Although the trade execution data is anonymous, the HFT is able to use its assumed knowledge to deduce the following:

- The first order to get executed on BYX1 would have to come from broker1 since broker1 is closer to BYX1 than broker2.

- The first order to get executed on BYX2 would have to come from broker2 since broker2 is closer to BYX2 than broker1.

Upon learning of the executed trades, each HFT agent uses its knowledge of delays to figure when the orders were sent and what LOB each broker saw at the sending time. If this is done successfully, both HFT1 and HFT2 will arrive at the conclusion that both brokers sent their orders directly after Milestone 1 based on LOBs seen at Milestone 1.

Using its knowledge of each broker’s total order size (assumption 3), the HFT aims to predict where each broker’s next order headed. HFT1 investigates where broker1 would send its second order. HFT2 investigates where broker2 would send its second order.

If either HFT1 or HFT2 find that the next order is possibly headed to the other (its non-local) exchange, it informs its HFT counterpart of the front-running opportunity through a debug message.
For example, HFT1 assumes that broker1 sent a 200 share buy order to BYX2. As such, it sends the following debug message to its counterpart HFT2:

“BRK1 Y, Executed Shares 400 Second Best Local Ask Price 24.10”

The ‘BRK1 Y’ represents that there is an opportunity to front-run broker1. The number of executed shares is the portion of broker1’s total order size that was executed. The ‘Second Best Local Ask Price’ is the second best price at BYX1 at the current time (after Milestone 2). The reason that the message quotes the second best price (‘$24.10) instead of the best price ($24.05) is because HFT1 expects to shift the best price upwards when it receives its front-running orders.

In our simulation, a similar message would be sent from HFT2 to HFT1 noting broker2’s purchase of 400 shares at BYX2.

At each time-step, both HFT agents also checks for notification messages from their counterpart. Upon receiving the above messages, HFT1 and HFT2 confirm the front running opportunity and execute both front running orders.

As in the previous experiments, the front-running orders consist of a market buy order for the current best price and a limit sell order at a higher price. The HFT agent is programmed to increase the price to $0.01 below the second best offer/ask price at the other exchange.

At Milestone 2, BYX1 and BYX2 both have best ask of $24.05 and the second best ask of $24.10. Broker1’s order #2 is a 200 share market buy order to BYX2 while Broker2’s order #2 is a 200 share market buy order to BYX1.

In order to front run broker1, HFT2 submits a sell limit order at price $24.09 to BYX2 ($0.01 less than the second best ask at BYX1). In order to front run broker2, HFT1 submits a sell limit order at price $24.09 to BYX1 ($0.01 less than the second best ask at BYX2).

Table 5.9 shows the orders sent by HFT1 and HFT2 in the scenario of a successful front run of both broker’s order #2.

<table>
<thead>
<tr>
<th>Sent By</th>
<th>Sent to (Exchange)</th>
<th>Type</th>
<th>Quantity</th>
<th>Side</th>
<th>Routable</th>
<th>Price</th>
<th>Time In Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFT2</td>
<td>BYX2</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>HFT2</td>
<td>BYX2</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.09</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>HFT1</td>
<td>BYX1</td>
<td>Market</td>
<td>200</td>
<td>BUY</td>
<td>FALSE</td>
<td>N/A</td>
<td>Good Till Cancelled</td>
</tr>
<tr>
<td>HFT1</td>
<td>BYX1</td>
<td>Limit</td>
<td>200</td>
<td>SELL</td>
<td>FALSE</td>
<td>24.09</td>
<td>Good Till Cancelled</td>
</tr>
</tbody>
</table>

Table 5.9: HFT Agents - Orders Sent

After both HFT agents have submitted their front-running orders, both exchanges send their updated best bid and offer (BBO) and receive the national best bid and offer (NBBO). Although the exchange of BBO and NBBO is done at every time-step, it is highlighted here because it plays an important role in determining whether the front-running attempt will be successful. At this point the simulation arrives at Milestone 3 in figure 5.2.
Figure 5.10 shows the LOBs at each exchange at Milestone 3.

<table>
<thead>
<tr>
<th>Milestone 3</th>
<th>BYX1</th>
<th></th>
<th>BYX2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid Size</td>
<td>Price</td>
<td>Ask Size</td>
<td>Bid Size</td>
<td>Price</td>
</tr>
<tr>
<td>24.10</td>
<td>600</td>
<td></td>
<td>24.10</td>
<td>600</td>
</tr>
<tr>
<td>24.09</td>
<td>200</td>
<td></td>
<td>24.09</td>
<td>200</td>
</tr>
<tr>
<td>24.05</td>
<td></td>
<td></td>
<td>24.05</td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td></td>
<td></td>
<td>24.00</td>
<td></td>
</tr>
<tr>
<td>23.99</td>
<td></td>
<td></td>
<td>23.99</td>
<td></td>
</tr>
<tr>
<td>23.98</td>
<td></td>
<td></td>
<td>23.98</td>
<td></td>
</tr>
<tr>
<td>23.97</td>
<td></td>
<td></td>
<td>23.97</td>
<td></td>
</tr>
<tr>
<td>23.96</td>
<td></td>
<td></td>
<td>23.96</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>23.95</td>
<td></td>
<td>200</td>
<td>23.95</td>
</tr>
<tr>
<td>23.90</td>
<td></td>
<td></td>
<td>23.90</td>
<td></td>
</tr>
<tr>
<td>23.85</td>
<td></td>
<td></td>
<td>23.85</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.10: Limit Order Book of BYX1 and BYX2 at Milestone 3 in timeline of Figure 5.2

The empty yellow and blue cells illustrates the absence of 200 shares offered at $24.05 on each exchange which were was bought by HFT2 and HFT1 respectively in order to raise the best offer to $24.09.

Immediately after Milestone 3, broker1’s market buy order for 200 shares (order #2) finally arrives at BYX2 and broker2’s market buy order for 200 shares (order #2) finally arrives at BYX1.

Broker2’s order #2 is immediately matched and executed on BYX1 against the 200 shares at $24.09 offered by HFT1. Broker1’s order #2 is immediately matched and executed on BYX2 against the 200 shares at $24.09 offered by HFT2.

When broker1 sent order #2 to BYX2 (at Milestone 1), broker1 was expecting its trade to execute at the best ask price at the time of $24.00. Similarly, broker2 was expecting its order #2 to BYX1 to execute against the best ask of $24.00.

Nevertheless, HFT1 and HFT2 raised the best asking price at BYX1 and BYX2 and both broker’s second orders executed $24.09.

In the timeline shown in figure 5.2, the HFT makes a profit off both brokers. HFT2 first buys 200 shares at $24.05 then sells the same 200 shares to broker1 for $24.09 resulting in a profit of $8.00. HFT1 also buys 200 shares at $24.05 then sells the same 200 shares to broker1 for $24.09 resulting in a profit of $8.00. While a total profit of $16.00 may not seem huge profit performing thousands of such trades across multiple securities and exchanges could end up being a hugely profitable strategy for a HFT.

In both broker’s point of view, the price seems to move immediately as the order is sent. This parallels what Brad Katsuyama claims to have experienced on numerous trades in 2008 and 2009.

Figure 5.3 is a pseudocode representation of the internal logic of HFT1 and HFT2.
5.5 Experiment Hypotheses

5.5.1 Null Hypothesis

The following null hypothesis is a statement that we aim to disprove.

**Null Hypothesis** – The HFT agent can front-run at least one broker’s orders regardless of latency (delay) conditions between agents

This hypothesis states that the HFT will always be successful at making a profit from at least one of the brokers. This experiment shall aim to dispel this hypothesis.

5.5.2 Alternative Hypotheses

The following at a set of formulae that will be used in our alternative hypotheses. Each formulae represents the time-steps taken for a sequence of actions to occur. The delay components of these formulae refer to the time-step delays between agents shown in the delay matrix in Table 5.2.

1. \[ A = \text{Time-steps necessary to front-run broker1’s order #2} \]
   
   \[ A = (\delta_3 + 1) + (\delta_{56} + 1) + (\delta_7 + 1) + (\delta_6 + 1) + (\delta_9 + 1) + (\delta_{96} + 1) \]

   Breakdown of components in A

---

Figure 5.3: Pseudocode representation of the internal logic of HFT1 and HFT2 for Experiment 3
Chapter 5. Experiment 3: Multiple Brokers

<table>
<thead>
<tr>
<th>Time-steps taken</th>
<th>A = Time-steps needed for front-running maneuver on broker1 (Time-steps needed to raise the BBO at BYX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Broker1’s order #1 executed at BYX1</td>
</tr>
</tbody>
</table>

2. \( B = \) Time-steps necessary for Broker1’s order #2 to arrive at BYX2

\[
B = (\delta_4 + 1)
\]

3. \( C = \) Time-steps necessary to front-run broker2’s order #2

\[
C = (\delta_{11} + 1) + (\delta_{6b} + 1) + (\delta_{7b} + 1) + (\delta_5 + 1) + (\delta_8 + 1) + (\delta_{8b} + 1)
\]

Breakdown of components in \( C \)

<table>
<thead>
<tr>
<th>Time-steps taken</th>
<th>( C = ) Time-steps needed for front-running maneuver on broker2 (Time-steps needed to raise the BBO at BYX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Broker2’s order #1 executed at BYX2</td>
</tr>
</tbody>
</table>

4. \( D = \) Time-steps necessary for Broker2’s order #2 to arrive at BYX1

\[
D = (\delta_{10} + 1)
\]

For brevity, the following hypotheses will mention \( A, B, C, D \) referring to the sequence of steps outlined above.

**Alternative Hypothesis 1** – The HFT will be successful at making a profit from broker1 by front-running broker1’s order #2 if the following conditional statement is fulfilled:

\[
(A \leq B) \land (C \leq B)
\]
**Alternative Hypothesis 2** – The HFT will be successful at making a profit from broker2 by front-running broker2’s order #2 if the following conditional statement is fulfilled:

\[(C \leq D) \land (A \leq D)\]

**Alternative Hypothesis 3** – The HFT will be successful at making a profit from both brokers simultaneously by front-running both broker1’s and broker2’s if the following logical statement is fulfilled:

\[(A \leq B) \land (C \leq B) \land (C \leq D) \land (A \leq D)\]

### 5.6 Resolving Hypotheses

#### 5.6.1 Results

The analysis of the simulation found that HFT’s success is highly dependent on delay conditions. Table 5.11 explores the outcome for the HFT given different permutations of delay conditions. A,B,C,D refer to the delay formulas calculated in section 5.5.2.

Table 5.11 highlights situations of potential failure for the HFT. The outcomes show that a failed front-run can have adverse consequences for the HFT. It can result in them having open positions that expose them to the market. Often, a failed front-run also leaves the HFT with a started sell limit order that is left unfilled.

#### 5.6.2 Disproving Null Hypothesis

**Null Hypothesis** – The HFT agent can front-run at least one broker’s orders regardless of latency (delay) conditions between agents

This null hypothesis can be refuted by extending an argument made in experiment 1. Experiment 3 is identical to experiment 1, except with with an extra broker agent. This means that if we remove broker2 from experiment 3, we are left with the identical circumstances as experiment 1. In experiment 1 we showed that the HFT cannot front-run a single broker’s order under all latency conditions (section 3.7.2). Adding an extra broker does not change this finding. Using the arguments made in section 3.7.2, we can state that there are some delay conditions under which the HFT will not be able to front-run either broker’s orders, showing that the null hypothesis is false.

Table 5.11 again illustrates how there are certain delay combinations that restrict the HFT from front running either broker.
<table>
<thead>
<tr>
<th>((A \leq B))</th>
<th>((C \leq B))</th>
<th>((C \leq D))</th>
<th>((A \leq D))</th>
<th><strong>Outcome</strong></th>
<th><strong>HFT front-runs Broker1</strong></th>
<th><strong>HFT front-runs Broker2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>- <strong>HFT front-runs both brokers' second order</strong>, profits $8 from each front-run</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
| TRUE | TRUE | TRUE | FALSE | - **HFT successfully front-runs broker1, profits $8**  
  - Broker2's order #2 to BYX1 doesn't match NBBO and is cancelled  
  - HFT ends up with  
    o 200 share long position, bought at $24.05  
    o Unfilled limit order to sell 200 shares at $24.09 on BYX1 | TRUE | FALSE |
| TRUE | TRUE | FALSE | TRUE | - **HFT successfully front-runs broker1, profits $8**  
  - HFT2's front-running orders to BYX2 are too late to front-run broker2  
  - HFT ends up with unfilled limit order to sell 200 shares at $24.09 on BYX2 | TRUE | FALSE |
| TRUE | TRUE | FALSE | FALSE | - **HFT successfully front-runs broker1, profits $8**  
  - Broker2's order #2 to BYX1 doesn't match NBBO and is cancelled  
  - HFTZ's front-running orders to BYX2 arrive too late  
  - HFT ends up with unfilled limit order to sell 200 shares at $24.09 on BYX2 | TRUE | FALSE |
| TRUE | FALSE | TRUE | TRUE | - **HFT successfully front-runs broker2, profits $8**  
  - Broker1's order #2 to BYX2 doesn't match NBBO and is cancelled  
  - HFT ends up with  
    o 200 share long position, bought at $24.05  
    o Unfilled limit order to sell 200 shares at $24.09 on BYX2 | FALSE | TRUE |
| FALSE | FALSE | TRUE | TRUE | - **HFT successfully front-runs broker2, profits $8**  
  - Broker1's order #2 to BYX2 doesn't match NBBO and is cancelled  
  - HFT1's front-running orders to BYX1 arrive too late  
  - HFT ends up with unfilled limit order to sell 200 shares at $24.09 on BYX1 | FALSE | TRUE |
| FALSE | TRUE | TRUE | TRUE | - **HFT successfully front-runs broker2, profits $8**  
  - HFT2's front-running orders to BYX2 are too late to front-run broker1  
  - HFT ends up with unfilled limit order to sell 200 shares at $24.09 on BYX1 | FALSE | TRUE |
| FALSE | TRUE | FALSE | FALSE | - **HFT fails at front-running either broker**, does not make a profit  
  - HFT ends up with  
    o 200 share long position, bought at $24.05  
    o Unfilled limit order to sell 200 shares at $24.09 on BYX2 | FALSE | FALSE |
| TRUE | FALSE | FALSE | FALSE | - **HFT fails at front-running either broker**, does not make a profit  
  - HFT ends up with  
    o Two 200 share long position, bought at $24.10  
    o Two unfilled limit orders to sell 200 shares at $24.09 | FALSE | FALSE |
| FALSE | FALSE | TRUE | FALSE | Logically impossible events | N/A | N/A |

**Table 5.11**: Table of delay events and outcomes for HFT
5.6.3 Justifying Alternative Hypotheses

**Alternative Hypothesis 1** – The HFT will be successful at making a profit from broker1 by front-running broker1’s order #2 if the following conditional statement is fulfilled: $(A \leq B) \land (C \leq B)$

One of the HFT’s aim is to front-run broker1’s order #2, which is on its way to BYX2. The HFT’s manoeuvre involves two key actions. Firstly, HFT2 sends a market buy order to BYX2 in order to remove the 200 shares available at price $24.05 from the LOB. The second step involves HFT2 sending a 200 share limit sell order to BYX2 at a price of $24.09, which is just below the previous BBO of $24.10. By performing its front-running manoeuvre, the HFT has raised the market price at BYX2 by buying 200 shares at $24.05 to and selling them at $24.09. The HFT’s success at front-running broker1 hinges on the two following conditions:

1. HFT2’s front-running orders need to arrive and execute at BYX2 before broker1’s order #2 arrives at BYX2.

   In section 5.5.2 we found that:
   
   - $A =$ Time-steps necessary to front-run broker1’s order #2
   - $B =$ Time-steps necessary for Broker1’s order #2 to arrive at BYX2

   Since $A$ needs to happen by the time $B$ does, this condition can be stated as $A \leq B$

2. When broker1’s order #2 arrives at BYX2, the national best offer for the security must equal to or greater than $24.09. If this is not the case, then broker1’s order cannot be fulfilled under RegNMS. Given that all the broker orders in this experiment are marked IOC, the order will be cancelled if not executed immediately.

   The NBBO will only shift to $24.09$ after both co-located HFTs execute their first front-running market order and both exchanges send their updated BBO and receive the updated NBBO. Condition 1 already covers the process of shifting the BBO at BYX2, therefore what remains is to calculate the time taken to shift the BBO at BYX1.

   In section 5.5.2 we found that:
   
   - $C =$ Time-steps needed to raise the BBO at BYX1
   - $B =$ Time-steps necessary for Broker1’s order #2 to arrive at BYX2

   Since $C$ needs to happen by the time $B$ does, this condition can be stated as $C \leq B$

The conjunction of condition 1 and 2 can be represented as the following conditional statement: $(A \leq B) \land (C \leq B)$. Given that this is identical to the conditional statement posited in alternative hypothesis 1, we can consider this hypothesis justified.
Table 5.11 provides empirical justification of this hypothesis, demonstrating that the HFT is only able to front run broker1 when both \((A \leq B)\) and \((C \leq B)\) are TRUE.

**Alternative Hypothesis 2** – The HFT will be successful at making a profit from broker2 by front-running broker2’s order #2 if the following conditional statement is fulfilled: \((C \leq D) \land (A \leq D)\)

In order to front-run broker2’s order #2 before it arrives at BYX1 the HFT must take two key actions. Firstly, HFT1 sends a market buy order to BYX1 in order to remove the 200 shares available at price $24.05 from the LOB. The second step involves HFT1 sending a 200 share limit sell order to BYX1 at a price of $24.09, which is just below the previous BBO of $24.10. In performing this front-running manoeuvre, the HFT has raised the market price at BYX1 by buying 200 shares at $24.05 to and selling them at $24.09. The HFT’s success in front-running broker2 hinges on two following conditions

1. HFT1’s front-running orders need to arrive and execute at BYX1 before broker2’s order #2 arrives at BYX1

   In section 5.5.2 we found that:
   
   - \(C\) = Time-steps necessary to front-run broker2’s order #2
   - \(D\) = Time-steps necessary for Broker2’s order #2 to arrive at BYX1

   Since \(C\) needs to happen by the time \(D\) does, this condition can be stated as \(C \leq D\)

2. When broker2’s order #2 arrives at BYX1, the national best offer for the security must equal to or greater than $24.09.

   The NBBO will only shift to $24.09 after both co-located HFTs execute their first front-running market order and both exchanges send their updated BBO and receive the updated NBBO. Condition 1 already covers the process of shifting the BBO at BYX2, therefore what remains is to calculate the time taken to shift the BBO at BYX2. In section 5.5.2 we found that:

   - \(A\) = Time-steps needed to raise the BBO at BYX2
   - \(D\) = Time-steps necessary for Broker2’s order #2 to arrive at BYX1

   Since \(A\) needs to happen by the time \(D\) does, this condition can be stated as \(A \leq D\)

The conjunction of condition 1 and 2 can be represented as the following conditional statement: \((C \leq D) \land (A \leq D)\).

Given that this is identical to the conditional statement posited in alternative hypothesis 2, we can consider this hypothesis justified.

The experiment results in Table 5.11 also justifies this hypothesis, showing how the HFT is only able to front run broker2 when both \((C \leq D)\) and \((A \leq D)\) are TRUE.
**Alternative Hypothesis 3** – The HFT will be successful at making a profit from both brokers simultaneously by front-running both broker1’s and broker2’s if the following logical statement is fulfilled:

\[(A \leq B) \land (C \leq B) \land (C \leq D) \land (A \leq D)\]

Justifying alternative hypothesis 3 is quite trivial as it involves deduction using the previous justifications.

In justifying alternative hypothesis 1 and 2 we found the conditions necessary to front-run (profit from) broker1 and broker2 respectively. As such, the logical conjunction of these scenarios should lead us to the delay conditions necessary for the HFT to front-run both brokers simultaneously.

Therefore, the HFT should be successful at front-running (making a profit from) both brokers simultaneously if the conjunction of the two above delay conditions hold.

\[(A \leq B) \land (C \leq B) \land (C \leq D) \land (A \leq D)\]

Since the above logical statement is identical to the conditional statement posited in alternative hypothesis 3, we can consider this hypothesis justified.

### 5.7 Limitations of Experiment

This experiment was an extension of the previous experiments and was designed to reflect a more realistic market scenario where there are multiple front-running targets. Nevertheless, the experiment omits details and makes assumptions that diminish its realism.

- **The set up limits the number of agent’s present**
  
  In a true market system, there would be many more market participants. There would be numerous possible front-running opportunities and the HFT would have to perform greater analysis to judge which ones to engage in.

- **The behaviour of the brokers are highly simplified**
  
  This experiment makes the assumption that the two brokers are each closer to opposite exchanges and that both brokers send their orders at the same time. These assumptions are highly generous as they allow the HFT to deduce with which orders come from which brokers.

### 5.8 Conclusion

The results of this experiment support the conclusions drawn from previous simulations, which suggest that even in a high simplified simulation within limited agents and generous assumption, front-running
Chapter 5. Experiment 3: Multiple Brokers

does not guarantee a profit for the HFT. On the contrary, we find that making the simulation more realistic is a hindrance to the HFT’s profit making abilities.

The inclusion of a second broker in our simulation created more obstacles in HFT’s path to a successful front-run and resulted in it’s the further tightening of its profit-making window. The increased complexity of our delay inequalities demonstrate the added difficulty in front-running two orders simultaneously.

Once again, this experiment demonstrates that the HFT’s profit making ability are highly dependent on certain factors that it cannot reliably know or control. Such factors include the delay between each broker and each exchange (\(\delta_3, \delta_4, \delta_{10}, \delta_{11}\)) and the delay between the exchange and CQS (\(\delta_9, \delta_{9b}\)). In Flash Boys, Michael Lewis does claim that certain HFT’s were able to obtain reliable information on exact delays between the CME and the Nasdaq [1] (pg.140), however this claim is not substantiated by any credible source.

This experiment also begins to illustrate the risks involved in front-running orders in a busy market place. Adding one broker results in an expansion of the possible outcomes of the simulation. In figure 5.11, we see that only in one case (when all four delay inequalities are true) does the HFT profit from both front-running attempts. In all the other cases, the simultaneous front-run is either partially or completely unsuccessful. It should be noted that in most of these unsuccessful cases, the HFT ends up with a risk-bearing open market position or an unfilled limit order.

These results also call into question the viability of front-running as a business model for HFT firms. The enforcement of the NBBO means that a front-running HFT is not able to raise prices above the NBO. For highly liquid securities with low bid/ask spreads, the profit from a successful front-run is greatly diminished. The profits are even lower when one considers the trading costs involved in each front-running manoeuvre. The volume of daily transactions needed to sustain the trading costs of a front-running business model is in the order of hundred of thousands. This volume of front-running comes with severe market risk, illustrating that a business model based on front-running is likely to be low in earning and highly uncertainty.

Juxtaposing the results from experiment 1 and experiment 3 illustrate the exponential growth in uncertainty for the HFT with each additional front-running target. Extrapolating these results to a market system of the magnitude of the US stock market is an effective thought experiment in realizing the difficulty involved in systematic front-running.
Chapter 6

Conclusion

6.1 HFTs ability and incentive to front-run

To what extent do HFTs operating the in the US stock market possess the correct knowledge and ability to front-run orders in a riskless manner in a market governed by RegNMS?

Experiment 1 (Chapter 3) simulates a market (governed by RegNMS) for a given security consisting of two exchanges, one broker who sends orders to both exchanges and a HFT who aims to front-run the brokers second order. Despite assuming that the HFT knows the brokers combined order size and that the broker submits all orders at the same time; our results show that the HFTs odds of making a profit are still uncertain.

In experiment 1, the HFTs profit making ability rests on the satisfiability of a delay inequality comprised of several delay components, most of which the HFT has no way of controlling.

For the HFT to make riskless profit from front-running non-routable orders from a single broker, they must guarantee that the delay inequality shown in section 3.7.3 is true. To be able to do this, the HFT must know the following:

- The exact delay between the broker and each exchange
- The delay between each exchange and CQS
- The delay between both its co-located servers
- The delay between each server and its co-located exchange

It is theoretically plausible for the HFT to figure out the delay between its two servers and between itself and the exchanges. However, it is much more difficult for the HFT to reliably predict delays between any two other agents in the market. This is especially the case if the data is transferred through a packet-based network opposed to a single cable. Either way, the network interception needed to figure out the delays between each broker, each exchange and CQS is unlikely to be legal.
Experiment 2 (Chapter 4) investigates the HFTs ability to front-run routable orders. By virtue of RegNMS, any exchange that receives a routable order is obliged to forward it on to a destination offering the best execution price.

The experiment found that front-running routable orders is even more complicated. For the HFT to make riskless profit from front-running routable orders from a single broker, they must guarantee that the delay inequality shown in section 4.6.3 is true. To be able to do this, the HFT must know the delays for Experiment 1 as well as the delay between the exchanges.

Lewis suggests that the work of Spread Networks and other such firms, has made it possible for HFTs to guarantee an ultra low latency connection between exchanges [1]. Nevertheless, there is very little evidence to this fact. On the contrary, in his rebuttal to Flash Boys, Kovac [2] explains that the only way a HFT can reliably know all the above delays is if they were tracking each individual broker and exchange and were monitoring all intermediate network traffic between them. The improbable and illegal nature of that scenario means that there is very little chance that the HFT can achieve riskless profit through front-running.

Experiment 3 (Chapter 5) involves multiple brokers and a HFT who aims to front-run both brokers simultaneously.

The assumptions in experiment 3 are quite generous towards the HFT. It was assumed that the HFT knew the total order size being executed by each broker, the delays between each broker and each exchange. It is also assumed that each broker was closer (in terms of delay) to a different exchange and that both brokers send all orders at the same time.

This time, the experiment produced multiple logical conditional statements, each consisting of delay inequalities. The experiment found that a series of delay inequalities, shown in section 5.6.3 need to be fulfilled in order for the HFT to be able to guarantee a riskless profit from front-running both brokers. If the inequality is not fulfilled the HFT is only successful at front-running one or none of the brokers. In the case of a fully or partialled failed front run, the HFT is often left with an open market position and/or an unfilled limit order.

The experiment showed that a real world HFT is unable to guarantee profit from front-running multiple brokers. It also shows that attempted front-runs often result in the HFT potentially taking on unplanned market positions.

The present US equity market consists of 8 public exchanges, about 5000 listed stocks, hundreds of HFT firms and millions of traditional brokers. With a market of this size, it is hard see how any given HFT firm, even one with a co-located connection every stock exchange in the country, could possess the ability to systematically engage in riskless front-running.

Other than delays some other obstacles to real world front-running are:
• The Brokers total order size.

When a HFT notices the first trade execution, it is not certain if there are more orders coming or how many shares are still outstanding.

• The sending time of the broker’s order

In the experiments we assume that the HFT knows that all brokers send orders at the same time. This is not true in reality. It is more likely for an experienced broker to split a large client order into smaller denominations and execute them over several minutes of trading.

• The destination of a routed order

This is especially significant in a RegNMS market where multiple exchanges have the same BBO. In such a case, it is up to the exchange to decide how to split and where to route the portions.

Another obstacle to front-running in the US equity market would be the abundance of liquidity at a given price point. Chances are there will be more shares offered at the best price than the broker is willing to buy. Thus, in order to make a profit, the HFT will still need to buy up all the shares at that price point (in order to move the price up) but will only be able to sell back a small portion of those.

Lastly, one should consider the challenging of front-running in a market with other front-runners. If riskless front-running using co-located connections was indeed possible; almost every HFT firm on the planet would adopt that strategy. This would mean that each individual HFT would have to compete to be faster than both the broker and all other front-runners. If there are enough HFTs engaging in front-running, the slower ones with less rigorous algorithms will soon find themselves being the victims of the practice themselves.

**To what extent is a front-running based business model an effective use of capital for a high frequency trading firm operating in a market governed by RegNMS?**

Judging from the experiments conducted and an analysis of RegNMS, it is clear that the risks associated with front-running greatly outweigh any potential profits. Firstly, there is the huge strategic risk involved with performing front-running in a market with so many uncertain factors. The HFTs lack of knowledge of the brokers total order size, sending time and the delays between other agents; a front-running strategy is unlikely to yield consistent profits.

There is an argument that certain measures such as wiretapping and hacking can be used to predict a brokers orders however these are highly illegal and thus come with their own set of risks.

Experiment 3 demonstrated that front-running manoeuvres can easily fail if the correct conditions arent met. These failed front-runs often leave the HFT with an open market position or an unfilled order. In that sense, front-running definitely doesnt guarantee profit and comes with significant market risk.
Taking all these factors into account, it is obvious that a front-running based business model is not an effective use of capital for a high frequency trading firm; especially one operating in the highly liquid and heavily regulated markets.

### 6.2 Plausability of Lewis’ claims in Flash Boys [1]

To what extent could the price shifting phenomenon outlined in Flash Boys be attributed to HFT firms engaging in front-running on a mass scale?

It is highly unlikely that that the price-shifting phenomenon experienced by Brad Katsuyama in Flash Boys was due to high frequency traders systematically front-running his orders. The experiments conducted in this project clearly demonstrate that it is very difficult for a HFT to identify a specific brokers orders. Additionally, the project found that front-running strategies result in uncertain profits and high market risk.

As such, it just doesn’t seem plausible that HFTs would be engaging in such activity and that they would be successful at front-running all of Brad Katsuyama’s orders.

A more reasonable explanation for the price shift is that Brad was trading in a highly liquid market with many HFTs and brokers present. Since Brad did not have a co-located connection, his slower connection meant that the exchanges caused him to see prices that were several milliseconds old. When Brad executed his orders, they joined the back of a queue of pending orders. By the time his orders were processed, the market price had shifted slightly and he got a worse execution price than he expected.

Clearly, Brad had been impacted by the influx of HFTs in the market; but he was not being specifically targeted or taken advantage of by predatory stock scalpers.

To what extent does RegNMS assist HFTs in front-running?

Many argue that RegNMS measures such as penny price intervals and the order protection rule have made it easier for HFTs to operate. Additionally, there are several RegNMS critics including Kieth Bliss, a former NYSE trader and veteran of electronic trading, who argue argument that the splitting and re-routing of orders has actually increased overall trading costs for the average investor as a single order is now executed in many smaller chunks[19].

While the benefit of RegNMS to the average investor can be debated, it is clear from the analysis that RegNMS does not help HFTs in front-running. In actuality, the experiments have found that aspects of RegNMS, such as the order protection rule, are an obstacle to riskless front-running. The experiments show that a routable order is much harder to front-run than a non-routable one (Section 4.8). With a routable order, there are many more unknowns including the destination of the order and the delay between the routing exchange and the receiving one. RegNMS also splits up the responsibilities for enforcing the order protection rule, with the CQS maintaining the NBBO and each individual exchange
routings the orders. This further complicated the HFTs front-running plans, meaning they need to know delays between the exchanges and CQS in order to front-run effectively. Overall, its clear that RegNMS makes front-running more difficult to perform and a much riskier strategy for HFTs.

6.3 Further Work

The experiments conducted in this project are by no means a complete analysis of the impact of HFTs on the equity market. There is room for further agent based simulation to be used to delve deeper into the topic of front-running and investigate other aspects of RegNMS. Some project further project ideas are given below.

- Exploring what measures (legal or illegal) can be taken by HFTs to improve their success at front-running
- Exploring what impact the presence of multiple front-running HFTs has on the profit-making ability of each individual HFT
- Investigating whether RegNMS has led to an average increase or decrease in overall cost of trading (transaction and securities)
Appendix A

Appendix A. Supporting Material

This appendix contains supporting material that provides details regarding the working of the implementation of the simulator and experiments.

A.1 BYX: Messages Sent/Received

Messages Received

- **BATS Orders Messages**

  At each time-step, the BYX exchange agent checks for any order messages it has received. These order messages represent Limit or Market orders sent to the exchange by other agents. These orders are then processed and either accepted, rejected or routed to another exchange.

- **NBBO (National Best Bid Offer)**

  At each time-step, the BYX exchange receives a message quoting the NBBO for all the securities it trades from the CQS agent. This allows the exchange to keep track of the best prices for each security, allowing it to make a decision on whether to route orders or execute them locally.

Messages Sent

- **Order Acknowledgements/ Rejections**

  For every order it receives, the BYX exchange needs to reply with a rejection or an acknowledgement. The exchange will reject a limit order if the order would result in a crossed limit order book. A crossed order book is one where the best bid price is equal or higher than the best (lowest) ask price. If an order is accepted, BYX will reply with an order acknowledgement message. In both cases, the message will be sent directly to the agent who sent in the original order.

- **Order Cancellation**
An order is cancelled when it can no longer be executed. The most common cause of an order cancellation is when an order expires. Every order submitted to the BYX exchange has an instruction specifying its time in force. For example, all Market orders are by default marked as Immediate or Cancel (IOC) meaning that if an order isn’t immediately matched to an existing limit order, then it is cancelled. Other expiry specifications include Good Till Cancelled (GTC), Fill or Kill (FOK) and DAY orders.

- **Order/Trade execution confirmations**

  An order execution confirmation is a message sent to both counterparties when any trade occurs on the exchange. The message specifies the number of shares executed and the price.

- **Order execution confirmations**

  An order execution confirmation is a message sent to both counterparties when any trade occurs on the exchange. The message specifies the number of shares executed and the price.

- **Order execution broadcasts**

  Each time a trade is executed, the BYX exchange is also required to broadcast the anonymized trade details to all those that subscribe the information. This information is identical to the trade execution confirmation and specifies the number of shares executed and at what price.

- **BBO (Best Bid Offer) Broadcast**

  At each time-step, the BYX exchange broadcasts the best bid and offer prices for each security to all those that subscribe to it. The CQS agent subscribes to the BBO broadcast of all exchanges and uses this information to construct the NBBO.

- **LOB (Limit Order Book) Broadcast**

  Along with the BBO, the BYX exchange also broadcasts its entire limit order book for each security to all agents with a level III data subscription. In our experiment, we assume that both the HFT and the Broker are subscribed to the LOB broadcast from BYX1 and BYX2. This gives them an idea of the prices and liquidity at each exchange and allows them to make calculated orders.

### A.2 System Manual - Running Experiments

#### A.2.1 Running Experiments

Running a particular experiment involves running the simulator located in `market-sim/Main.hs`

To run a particular simulation, simply uncomment the name of the experiment (making sure to comment out the rest) and build the file.

It should be noted that the experiments are labelled differently in the code than in this report.
Appendix A. Supporting Material

A.2.2 Modifying Experiment Setup

The actual set ups of the experiment are defined in market-sim/SaagarExperiments.hs. This file defines the agents in the experiment and their delays. It also defines the behaviour of all agents by linking the agents to wrapper functions.

The inter agent delays for each experiment are parsed as an adjacency matrix as show in Figure A.2. The rows represent the 'from' agent and the columns represent the 'to' agent.

A.2.3 Modifying Agent Behaviour

Adjusting the agent behaviour is done by editing the wrapper functions that govern the messages sent by each agent. The code for the wrapper functions for all agents can be found in the following directory: market-sim/Agents.
• The code for the broker agents can be found in market-sim/Agents/Brokers
  
  – **BrokerV1.hs** is the code for the broker agent in Experiment 1 (frExp1)
  
  – **BrokerV1a.hs** is the code for the broker agent in Experiment 2 (frExp1a)
  
  – **BrokerV2.hs** is the code for the broker agent in Experiment 3 (frExp3)

• The code for the HFT agents can be found in market-sim/Agents/Misc
  
  – **HFTV1.hs** is the code for the HFT agent in Experiment 1 and 2 (frExp1 and frExp1a)
  
  – **HFTV2.hs** is the code for the HFT agent in Experiment 3 (frExp2)

• The code for the Probe agents can be found in market-sim/Agents/Traders
  
  – **ProbeV1.hs** is the code for the Probe agent in Experiment 1 and 2 (frExp1 and frExp1a)
  
  – **ProbeV2.hs** is the code for the Probe agent in Experiment 3 (frExp2)

Another key aspect of the agents is the composition of their state record. This is a class definition that determines what each agent stores in its internal state. The internal state records of each agent can be found in market-sim/Common/Comm.hs.

Figure A.3 shows a snippet of Comm.hs where the internal state of the agents are defined.

```haskell
-- | Internal state for Probe agent.
data ProbeV1StateRecord
  = ProbeV1StateRecord
    { _pClordIDs :: [Int] -- Client Order IDs, unique for each order.
    , _limOrds :: [OrderDetails] -- list of orders to be sent
    } deriving (Show, Eq)

-- | Internal state for Broker agent.
data BrokerV1StateRecord
  = BrokerV1StateRecord
    { _bClOrdIDs :: [Int] -- Client Order IDs, unique for each order.
    , _mrkOrds :: [OrderDetails] -- list of orders to be sent
    } deriving (Show, Eq)

-- | Internal state for HFT agent.
data HFTV1StateRecord
  = HFTV1StateRecord
    { _hClOrdIDs :: [Int] -- Client Order IDs, unique for each order.
    , _BYX1Bids :: Map.Map Int BYX1Orderlist -- Map of bids at each time BYX1
    , _BYX1Asks :: Map.Map Int BYX1Orderlist -- Map of asks at each time BYX1
    , _BYX2Bids :: Map.Map Int BYX2Orderlist -- Map of bids at each time BYX2
    , _BYX2Asks :: Map.Map Int BYX2Orderlist -- Map of asks at each time BYX2
    , _BrokrOrdSize :: Int -- Total market order size (we're assuming HFT knows)
    , _TradeExecutions :: Map.Map (Int, Shares, AgentSimID, AgentSimID) -- Map of trade executions at each time
    } deriving (Show, Eq)
```

**Figure A.3: Comm.hs**
Appendix B

Appendix B. Simulator Code

B.1 BATSMessages.hs

The following code is a function that was added to the BATSMessages file to allow for faster order creation within the simulator. The complete file can be found in market-sim/Common/FIXMessages/BATSMessages.hs

```
-- Order detail data constructor
-- stores useful values for creating orders
data OrderDetails = OrderDetails
    { sendingTime :: Int,
      myID :: Int,
      uniqueOrderID :: Int,
      quantity :: Int,
      exchLabel :: String,
      otype :: String,
      side :: String,
      symbol :: String,
      routable :: Bool,
      price :: Double,
      timeinF :: String,
      expireTime :: Int
    }
```

```
-- Saagar Additions
```

---

```
-- Create new order easily
-- Takes in order details, outputs bats message
```
Appendix B. Simulator Code

-- _ = exchange Label (assigned when creating BYX message)
-- trace ("HFT State at time " ++ show time ++ ": " ++ (showBYXOrderlist (fromJust (Map.lookup time (_BYX2Asks updatedState) )) )
createBATSOrder :: OrderDetails -> BATSMessage
createBATSOrder (OrderDetails sendingTime myID uniqueOrderID quantity exchLabel otype side symbol routable price timeinF expTime) = trace (exchLabel ++ " receives " ++ otype ++ " " ++ side ++ " order from Agent " ++ show myID ++ " at price " ++ show price)
batsNewOrderMessageSkeleton

{ _BNO_SendingTime_FT = checkAndSetTagValue BYX sendingTime_FT (SendingTime_FTVal sendingTime)
  , _BNO_ClOrdID_FT = checkAndSetTagValue BYX clOrdID_FT (ClOrdID_FTVal myID uniqueOrderID)
  , _BNO_OrderQty_FT = checkAndSetTagValue BYX orderQty_FT (OrderQty_FTVal quantity)
  , _BNO_OrdType_FT = checkAndSetTagValue BYX ordType_FT ordType
  , _BNO_Side_FT = checkAndSetTagValue BYX side_FT ordSide
  , _BNO_Symbol_FT = checkAndSetTagValue BYX symbol_FT (Symbol_FTVal symbol)
  , _BNO_RoutingInst_BT = checkAndSetTagValue BYX routingInst_BT routing
  , _BNO_Price_FT = checkAndSetTagValue BYX price_FT (Price_FTVal price)
  , _BNO_TimeInForce_FT = checkAndSetTagValue BYX timeInForce_FT timeInForce
  , _BNO_ExpireTime_FT = checkAndSetTagValue BYX expireTime_FT (ExpireTime_FTVal expireTime)
}
where

ordType
| otype == "lim" = OrdType_FTVal_Limit
| otype == "mrkt" = OrdType_FTVal_Market

ordSide
| side == "buy" = Side_FTVal_Buy
| side == "sell" = Side_FTVal_Sell

routing
| routable == True = RoutingInst_BTVal_Routable
| routable == False = RoutingInst_BTVal_BookOnly

timeInForce
| timeinF == "IOC" = TimeInForce_FTVal_IOC
| timeinF == "FOK" = TimeInForce_FTVal_FOK
| timeinF == "DAY" = TimeInForce_FTVal_Day
| timeinF == "GTC" = TimeInForce_FTVal_GTC

expireTime
| timeinF == "GTC" = expTime
| otherwise = 0

B.2 SaagarExperiments.hs

The following is a snippet of code to setup the experiments conducted in this project. The complete file can be found in market-sim/saagarExperiments.hs
Appendix B. Simulator Code

-- Front Running Ex 1: 2 Exchanges, FR colocated at each, Probe populating LOB, Broker making non-routable

frExp1 :: IO ()
frExp1
= do
  createDirectoryIfMissing True ("SimulationOutput/" ++ thefilename)
  sim 60 meinargs (RuntimeArgFunctions.transformForSim agents)
where
  meinargs = [(Arg (Str "Calm", 1)),
              (Arg (Str fthefilename, 9989793425))
              , (Arg (Str "Randomise", 1))
              , funArg1, maxDelayArg, funArg3 ] -- funArg2, funArg3 ]
  fthefilename = "SimulationOutput/" ++ thefilename ++ "/" ++ thefilename
  thefilename = "frExp1"

-- We define our agent list in the new-style i.e. with an agent label for each wrapper.
agents = [("BYX1", (byxWrapper, [globalNBBOsBroadcastChannel] ))
          , ("BYX2", (byxWrapper, [globalNBBOsBroadcastChannel] ))
          , ("CQS", (cqsWrapper, [globalBBOsBroadcastChannel] ))
          , ("Probe", (probeV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
          , ("Broker", (brokerV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
          , ("HFT1", (hFTV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
          , ("HFT2", (hFTV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
          ]

-- We encode our adjacency matrix representation of the experiment's graph.
delays :: [ [Maybe Int] ]
delays = [ [Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0]
          , [Just 0, Nothing, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0]
          , [Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Just 0, Just 0]
          , [Just 0, Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Just 0]
          , [Just 0, Just 0, Just 0, Nothing, Just 0, Just 0, Nothing, Just 0]
          , [Just 0, Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Nothing]
          , [Just 0, Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Nothing]]

-- Front Running Ex 1a: 2 Exchanges, FR colocated at each, Probe populating LOB, Broker making 2 routable

frExp1a :: IO ()
frExp1a
= do
createDirectoryIfMissing True ("SimulationOutput/"++thefilename)
sim 60 meinargs (RuntimeArgFunctions.transformForSim agents)
where
meinargs = [(Arg (Str "Calm", 1))
, (Arg (Str fthefilename, 9989793425))
, (Arg (Str "Randomise", 1))
, funArg1, maxDelayArg, funArg3 ] -- funArg2, funArg3 ]

fthefilename = "SimulationOutput/" ++ thefilename ++ "/" ++ thefilename

thefilename = "frExp1a"

-- We define our agent list in the new-style i.e. with an agent label for each wrapper.
agents = [ ("BYX1", (byxWrapper, [globalNBBOsBroadcastChannel] )),
  ("BYX2", (byxWrapper, [globalNBBOsBroadcastChannel] )),
  ("CQS", (cqsWrapper, [globalBBOsBroadcastChannel] )),
  ("Probe", (probeV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] )),
  ("Broker", (brokerV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] )),
  ("HFT1", (hFTV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
  ("HFT2", (hFTV1Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
]

-- Front Running Ex 2: 2 Exchanges, FR colocated at each, Probe populating LOB, 2 Brokers making 2 non-routable IOC market orders to both exchange

frExp2 :: IO ()
frExp2
= do
createDirectoryIfMissing True ("SimulationOutput/"++thefilename)
sim 60 meinargs (RuntimeArgFunctions.transformForSim agents)
where
meinargs = [(Arg (Str "Calm", 1))
, (Arg (Str fthefilename, 9989793425))
, (Arg (Str "Randomise", 1))
, funArg1, maxDelayArg, funArg3 ] -- funArg2, funArg3 ]

fthefilename = "SimulationOutput/" ++ thefilename ++ "/" ++ thefilename

thefilename = "frExp2"

-- We define our agent list in the new-style i.e. with an agent label for each wrapper.
agents = [ ("BYX1", (byxWrapper, [globalNBBOsBroadcastChannel] )),
  ("BYX2", (byxWrapper, [globalNBBOsBroadcastChannel] )),
  ("CQS", (cqsWrapper, [globalBBOsBroadcastChannel] )),
  ("Probe", (probeV2Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] )),
  ("Broker1", (brokerV2Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] ))
]
Appendix B. Simulator Code

("Broker2", (brokerV2Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] )),
("HFT1", (hFTV2Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] )), -- This would be given ID 7, HFT Collocated at Exchange1.
("HFT2", (hFTV2Wrapper, [globalLOBsBroadcastChannel, globalTradeExecutionsBroadcastChannel] )) -- This

---------------- FunArg1

-- We encode our adjacency matrix representation of the experiment’s graph.

delays :: [ [Maybe Int] ]
delays = [ -- {- 0 Sim -} {-1 BYX_Exch_1-} {-2 BYX_Exch_2-} {-3 CQS-} {-4 PR-} {-5 BRK1-} {-6 BRK2} {-7 HFT1-} {-8 HFT2-}
  {-0 Simulator -} [ Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0 ],
  {-1 BYX_Exch_1-} [ Just 0, Nothing, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0 ],
  {-2 BYX_Exch_2-} [ Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0 ],
  {-3 CQS-} [ Just 0, Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Just 0, Just 0 ],
  {-4 PR1-} [ Just 0, Just 0, Just 0, Just 0, Nothing, Just 0, Just 0, Just 0, Just 0 ],
  {-5 BRK1-} [ Just 0, Just 1, Just 7, Just 0, Just 0, Nothing, Just 0, Just 0, Just 0 ],
  {-6 BRK2-} [ Just 0, Just 7, Just 1, Just 0, Just 0, Just 0, Nothing, Just 0, Just 0 ],
  {-7 HFT1-} [ Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Just 0, Nothing, Just 0 ],
  {-8 HFT2-} [ Just 0, Just 0, Just 0, Just 1, Just 0, Just 0, Just 0, Just 0, Nothing ]

B.3 ProbeV1.hs

The following is a snippet of code for the probe agent in experiment 1.

The complete file can be found in market-sim/Agents/Traders/ProbeV1.hs

probeV1Wrapper :: Agent_t
-- Empty state case

probeV1Wrapper (Emptyagentstate) args ((time, messages, broadcasts) : rest) myID
  = ( ( [Hiaton] ) -- Output a hiaton on emptyagentstate
      : (probeV1Wrapper (ProbeV1State initialState) args rest myID) ) -- Update state to initialstate of type [int], client orders
  where
    initialState = ProbeV1StateRecord { _pClOrdIDs = [1..], _limOrds = initialProbeOrders time myID} -- Initial state

-- Non empty state to next state

probeV1Wrapper (ProbeV1State currentState) args
((time, messages, broadcasts) : rest) myID
  = ( output : ( probeV1Wrapper (ProbeV1State updatedState) args rest myID ) )
  where
    (updatedState, output) -- Updated state, output message determined by time
      | time == 4 = probeLogic currentState args time myID broadcasts
      | otherwise = (currentState, [])
-- Creates order message from order details and BATS message
createBYXMessage ord@(OrderDetails {myID = me, exchLabel = exch}) args
  = BYXMessage (me, byxID) batsMessage
where
  (AgentID byxID) = getCorrespondingAgentIdentifier (AgentLabel exch)
getCorrespondingAgentIdentifier = returnGetCorrespondingAgentIdentifierFunction args
batsMessage = createBATSOrder ord

-- Create initial orders. Orders that probe will execute
-- (unique ID’s are all set to 1, will be reassigned when order is executed)
initialProbeOrders :: Int -> Int -> [OrderDetails]
  -- sending time ID ORDID QT EXCHLBL TYPE SIDE SYM ROUT PRICE TINF EXP
initialProbeOrders time myID = [OrderDetails time myID 1 200 "BYX1" "lim" "buy" "APPL" False 23.95 "GTC" 60,
  OrderDetails time myID 1 400 "BYX1" "lim" "sell" "APPL" False 24.00 "GTC" 60,
  OrderDetails time myID 1 600 "BYX1" "lim" "sell" "APPL" False 24.10 "GTC" 60,
  OrderDetails time myID 1 200 "BYX2" "lim" "buy" "APPL" False 23.95 "GTC" 60,
  OrderDetails time myID 1 200 "BYX2" "lim" "sell" "APPL" False 24.05 "GTC" 60]

B.4 BrokerV1.hs

The following is a snippet of code for the broker agent in experiment 1.
The complete file can be found in market-sim/Agents/Brokers/BrokerV1.hs

brokerV1Wrapper :: Agent_t
  -- Empty state case
brokerV1Wrapper (Emptyagentstate) args ((time, messages, broadcasts) : rest) myID
  = ( ([Hiaton] ) -- Output a hiaton on emptyagentstate
    : (brokerV1Wrapper (BrokerV1State initialState) args rest myID) ) -- Update state to initialstate of type
where
  initialState = BrokerV1StateRecord { _bClOrdIDs = [1..], _mrktOrds = initialBrokerOrders time myID} -- In

  -- Non empty state to next state
brokerV1Wrapper (BrokerV1State currentstate) args
  ((time, messages, broadcasts) : rest) myID
  = ( output : ( brokerV1Wrapper (BrokerV1State updatedstate) args rest myID) )
where
  (updatedstate, output) -- Updated state, output message determined by time
  | time == 10 = brokerLogic currentstate args time myID broadcasts
  | otherwise = (currentstate, [])
Appendix B. Simulator Code

-- Defines broker actions sends orders
brokerLogic state args time myID broadcasts = (updatedState, outputMessages)
where
outputMessages = [executeOrder zipState args time | zipState <- zip (_bClOrdIDs state) (_mrktOrds state)]
updatedState = state { _bClOrdIDs = drop size currentClOrdIDList, _mrktOrds = []}
size = length currentMrktOrds

currentMrktOrds = _mrktOrds state -- returns the Market orders list of the state
currentClOrdIDList = _bClOrdIDs state -- returns the ClientOrderID list of the state

-- Update Unique ID to order, and create BYX message
executeOrder (clOrdID, ordDets) args time
= createBYXMessage adjustedOrdDets args
where
adjustedOrdDets = ordDets {sendingTime = time, uniqueOrderID = clOrdID}

-- Create initial orders. Orders that probe will execute
-- (unique ID’s are all set to 1, will be reassigned when order is executed)
initialBrokerOrders :: Int -> Int -> [OrderDetails]
initialBrokerOrders time myID = [OrderDetails time myID 1 400 "BYX1" "mrkt" "buy" "APPL" False 0.00 "IOC" 1
 , OrderDetails time myID 1 200 "BYX2" "mrkt" "buy" "APPL" False 0.00 "IOC" 1]}

B.5 HFTV1.hs

The following is a snippet of code for the HFT agent in experiment 1.
The complete file can be found in market-sim/Agents/Misc/HFTV1.hs

hFTV1Wrapper :: Agent_t
-- Empty state case
hFTV1Wrapper (Emptyagentstate) args ((time, messages, broadcasts) : rest) myID
= ( ( [Hiaton] ) -- Output a hiaton on emptyagentstate
 : (hFTV1Wrapper (HFTV1State initialState) args rest myID) ) -- Update state to initialstate of type [int]
where
initialState = HFTV1StateRecord { _hClOrdIDs = [1..] -- Initial state is set to [1,2,3...]
 , _BYX1Bids = Map.fromList [[0,([1], Bids))] -- Map of bids at each time BYX1
 , _BYX1Asks = Map.fromList [[0,([1], Asks))] -- Map of asks at each time BYX1
 , _BYX2Bids = Map.fromList [[0,([1], Bids))] -- Map of bids at each time BYX2
 , _BYX2Asks = Map.fromList [[0,([1], Asks))] -- Map of asks at each time BYX2
 , _BrokerOrdSize = 600 -- Total market order size
 , _TradeExecutions = Map.fromList [[0,(0,0,0,0)]]
}
-- Defines HFT Logic
-- For debugging

-- trace ("Trade execution at time :") ++ show time ++ "," ++ show (fromJust (Map.lookup time (_TradeExecutions updatedState)))
-- trace ("Shares remaining at time :") ++ show time ++ "," ++ show (_BrokerOrdSize updatedState)
-- trace (whatAsksBrokerSaw myID args time tradeExTime)
-- trace ("Broker Saw: " ++ (showBYXOrderlist brokerAsks1) ++ " when making trade " ++ show (tradeExTime))

hFTLogic state args time myID broadcasts messages = (updatedState, outputMessages)

where

outputMessages = tradeSeenMessage ++ orders
updatedState = state { 
    _BYX1Bids = byx1BidMap -- Map of bids at each time BYX1
    , _BYX1Asks = byx1AskMap -- Map of asks at each time BYX1
    , _BYX2Bids = byx2BidMap -- Map of bids at each time BYX2
    , _BYX2Asks = byx2AskMap -- Map of asks at each time BYX2
    , _TradeExecutions = tradeExMap -- Stores details of trade executed at this time
    , _BrokerOrdSize = finalRemainingShares
    , _hClOrdIDs = drop (length orders) currentClOrdIDList
}

currentClOrdIDList = _hClOrdIDs state -- returns the ClientOrderID list of the state

byx1BidMap = Map.insert time (getBYXOrderList broadcasts args "BYX1" Bids) (_BYX1Bids state) -- Update maps at each timestep
byx1AskMap = Map.insert time (getBYXOrderList broadcasts args "BYX1" Asks) (_BYX1Asks state)
byx2BidMap = Map.insert time (getBYXOrderList broadcasts args "BYX2" Bids) (_BYX2Bids state)
byx2AskMap = Map.insert time (getBYXOrderList broadcasts args "BYX2" Asks) (_BYX2Asks state)

orders = [executeOrder zipState args | zipState <- zip (_hClOrdIDs state) frOrders] -- Execute if trade execution message is received and broker still has shares left
frOrders = createFrOrders frOppSeen bestLocalAskPrice bestLocalAskLiqudity bestPriceOtherExch finalRemainingShares -- Depending on front running opportunity, create frontrunning orders

finalRemainingShares -- Updates the broker's shares remaining, Takes into account debug received from other exchanges

| debugSeen = newRemainingShares - sharesExecuted
| otherwise = newRemainingShares

(debugSeen, frOppSeen, sharesExecuted, bestPriceOtherExch) = checkDebugs messages

-- received debug, genuine front running opportunity, shares executed on other exchange, best ask price on other exchange

tradeSeenMessage -- Decide whether or not to send debug message and what to put in the message

| getShares tradeExTime > 0 = [genTradeSeenMessage myID args tradeExTime frOpp bestLocalAskPrice]
| otherwise = []

frOpp -- Check for a possible front running opportunity

| getShares tradeExTime > 0 && newRemainingShares > 0 && nextOrdTo /= imAt = trace ("Possible Front Running")
| otherwise = False

nextOrdTo = nextOrderIsGoingTo brokerAsks1 brokerAsks2 myID args -- based on ask books, where

(brokerAsks1, brokerAsks2) = whatBrokerSaw myID args time sendTime state -- returns the ask books from both exchange

-- we assume broker only sends buy order

sendTime = whenOrderWasSent myID args time tradeExTime -- If a trade was spotted on local exchange

newRemainingShares = currentRemainingShares - (getShares tradeExTime) -- Remaining shares after subtracting local exchange trades

tradeExMap = Map.insert time tradeExTime (_TradeExecutions state) -- Update trade execution map
tradeExTime = getTradeExecution broadcasts myID args -- Trade execution at current time

currentRemainingShares = _BrokerOrdSize state -- returns the ClientOrderID list of the state

bestLocalAskPrice = fromJust (olBYX_getLowestPrice (olBYX_clean (getBYXOrderList broadcasts args imAt Asks)))

bestLocalAskLiqudity = fromJust (olBYX_peekFirstOrderSize (olBYX_clean (getBYXOrderList broadcasts args imAt Asks)))

imAt = colcatedAt myID args -- Where current HFT is colocated

getBYXOrderList ((RoutedBroadcast (from,to) (Broadcastmessage _ (BYXOrderlistBroadcast byxordlst@(_,oLstSide)))): xs) args exchLabel side

getTradeExecution ((RoutedBroadcast (from,to) (Broadcastmessage _ (BYXTradeExecutionBroadcast tradeDets@(price, shares, cp1ID, cp2ID)))): xs) myID args

genTradeSeenMessage myID args tradeDets@(@price, shares, cp1ID, cp2ID) frOpp bestLocalAsk

getCorrespondingAgentIdentifier = returnGetCorrespondingAgentIdentifierFunction args

getBYXOrderList (_ : xs) args exchLabel side = getBYXOrderList xs args exchLabel side

getBYXOrderList [] _ _ side = (([], side))

getTradeExecution (_ : xs) myID args = getTradeExecution xs myID args

getTradeExecution [] _ _ = (0,0,0,0)
| otherwise = getCorrespondingAgentIdentifier (AgentLabel "HFT1")

getCorrespondingAgentIdentifier = returnGetCorrespondingAgentIdentifierFunction args

-- Creates front running orders if fronrunning opportunity exists
createFrOrders frOppSeen bestLocalAskPrice bestLocalAskLiqudity bestPriceOtherExch finalRemainingShares time myID imAt
  | frOppSeen == True && bestLocalAskLiqudity <= finalRemainingShares && finalRemainingShares > 0
  = trace ("\n-----------------------------------------------
" ++ imAt ++ " has verified FR opportunity.\n"
++
"Expecting " ++ show finalRemainingShares ++ " or less market buy order to arrive.\n"
++
"Current best ask on " ++ imAt ++ " is price/qty " ++ show bestLocalAskPrice ++ "/" ++ show bestLocalAskLiqudity
++
"Best price on other exchange is " ++ show bestPriceOtherExch ++ "/\n"
++
"I will be submitting market buy order, then a limit sell order to raise price to " ++ show frPrice ++
"
-----------------------------------------------
"
where
  frPrice = bestPriceOtherExch - 0.01

-- Returns delays between agents
delay :: String -> String -> Int
delay "Broker" "BYX1" = 1
delay "BYX1" "Broker" = 0
delay "Broker" "BYX2" = 0

-- Returns exchange which this agent is colocated at takes in ID and args
colcatedAt myID args | hftID == myID = "BYX1"
| otherwise = "BYX2"
where
  (AgentID hftID) = getCorrespondingAgentIdentifier (AgentLabel "HFT1")
  getCorrespondingAgentIdentifier = returnGetCorrespondingAgentIdentifierFunction args

B.6 BrokerV1a.hs

The following is a snippet of code for the Broker agent in experiment 1a.
The complete file can be found in market-sim/Agents/Brokers/BrokerV1a.hs

brokerV1aWrapper :: Agent_t
-- Empty state case
brokerV1aWrapper (Emptyagentstate) args ((time, messages, broadcasts) : rest) myID
  = ( ( [Hiaton] ) -- Output a hiaton on emptyagentstate
Appendix B. Simulator Code

: (brokerV1aWrapper (BrokerV1State initialState) args rest myID) -- Update state to initialState of type [Int], client orders
where
initialState = BrokerV1StateRecord { _bClOrdIDs = [1..], _mrktOrds = brokerOrders time myID} -- Initial state is set to [1,2,3...]
-- Non empty state to next state
brokerV1aWrapper (BrokerV1State currentState) args
((time, messages, broadcasts) : rest) myID
= ( output : ( brokerV1aWrapper (BrokerV1State updatedState) args rest myID) )
where
(updatedState, output) -- Updated state, output message determined by time
| time == 10 = brokerLogic currentState args time myID broadcasts
| otherwise = (currentState, [])

-- Create initial orders. Orders that probe will execute
-- (unique ID's are all set to 1, will be reassigned when order is executed)
brokerOrders :: Int -> Int -> [OrderDetails]
brokerOrders time myID = [OrderDetails time myID 1 400 "BYX1" "mrkt" "buy" "APPL" False 0.00 "IOC" 1, OrderDetails time myID 1 200 "BYX1" "mrkt" "buy" "APPL" True 0.00 "DAY" 1]

B.7 ProbeV2.hs

The following is a snippet of code for the probe agent in experiment 2.
The complete file can be found in market-sim/Agents/Traders/ProbeV2.hs

probeV2Wrapper :: Agent_t
-- Empty state case
probeV2Wrapper (Emptyagentstate) args ((time, messages, broadcasts) : rest) myID
= ( [Hiaton] ) -- Output a hiaton on emptyagentstate
: (probeV2Wrapper (ProbeV1State initialState) args rest myID) ) -- Update state to initialState of type [Int]
where
initialState = ProbeV1StateRecord { _pClOrdIDs = [1..], _limOrds = initialProbeOrders time myID} -- Initial state is set to [1,2,3...]
-- Non empty state to next state
probeV2Wrapper (ProbeV1State currentState) args
((time, messages, broadcasts) : rest) myID
= ( output : ( probeV2Wrapper (ProbeV1State updatedState) args rest myID) )
where
(updatedState, output) -- Updated state, output message determined by time
| time == 4 = probeLogic currentState args time myID broadcasts
| otherwise = (currentState, [])
Appendix B. Simulator Code

-- Defines probe actions sends orders
probeLogic state args time myID broadcasts = (updatedState, outputMessages)
where
outputMessages = [executeOrder zipState time | zipState <- zip (_pClOrdIDs state) (_limOrds state)]
updatedState = state { _pClOrdIDs = drop size currentClOrdIDList, _limOrds = []}
size = length currentLimOrds

currentLimOrds = _limOrds state -- returns the ClientOrderID list of the state

currentClOrdIDList = _pClOrdIDs state -- returns the ClientOrderID list of the state

-- Create initial orders. Orders that probe will execute
-- (unique ID's are all set to 1, will be reassigned when order is executed)
initialProbeOrders :: Int -> Int -> [OrderDetails]
-- sending time ID ORDID QT EXCHLBL TYPE SIDE SYM ROUT PRICE TINF EXP
initialProbeOrders time myID = [OrderDetails time myID 1 200 "BYX1" "lim" "buy" "APPL" False 23.95 "GTC" 60
, OrderDetails time myID 1 400 "BYX1" "lim" "sell" "APPL" False 24.00 "GTC" 60
, OrderDetails time myID 1 600 "BYX2" "lim" "sell" "APPL" False 24.10 "GTC" 60
, OrderDetails time myID 1 200 "BYX2" "lim" "buy" "APPL" False 23.95 "GTC" 60
, OrderDetails time myID 1 400 "BYX2" "lim" "sell" "APPL" False 24.00 "GTC" 60
, OrderDetails time myID 1 600 "BYX2" "lim" "sell" "APPL" False 24.10 "GTC" 60]

B.8 BrokerV2.hs

The following is a snippet of code for the broker agent in experiment 2.
The complete file can be found in market-sim/Agents/Brokers/BrokerV2.hs

brokerV2Wrapper :: Agent_t
-- Empty state case
brokerV2Wrapper (EmptyAgentstate) args ((time, messages, broadcasts) : rest) myID
= ( ( [Hiaton] ) -- Output a hiaton on emptyagentstate
  : (brokerV2Wrapper (BrokerV1State initialState) args rest myID) ) -- Update state to initialstate of type
where
initialState = BrokerV1StateRecord { _bClOrdIDs = [1..], _mrktOrds = initialBrokerOrders time myID args}

-- Non empty state to next state
brokerV2Wrapper (BrokerV1State currentState) args
((time, messages, broadcasts) : rest) myID
= ( output : ( brokerV2Wrapper (BrokerV1State updatedState) args rest myID) )
where
(updatedState, output) -- Updated state, output message determined by time
  | time == 10 = brokerLogic currentState args time myID broadcasts
  | otherwise = (currentState, [])
-- Create initial orders. Orders that probe will execute
-- (unique ID's are all set to 1, will be reassigned when order is executed)
initialBrokerOrders time myID args | whichBroker myID args == 1 = 
[OrderDetails time myID 1 400 "BYX1" "mrkt" "buy" "APPL" False 0.00 "IOC" 1,
OrderDetails time myID 1 200 "BYX2" "mrkt" "buy" "APPL" False 0.00 "IOC" 1]
| whichBroker myID args == 2 = 
[OrderDetails time myID 1 400 "BYX2" "mrkt" "buy" "APPL" False 0.00 "IOC" 1,
OrderDetails time myID 1 200 "BYX1" "mrkt" "buy" "APPL" False 0.00 "IOC" 1]

-- Returns which broker this agent
whichBroker myID args | brkID == myID = 1
| otherwise = 2
where
(AgentID brkID) = getCorrespondingAgentIdentifier (AgentLabel "Broker1")
getCorrespondingAgentIdentifier = returnGetCorrespondingAgentIdentifierFunction args

B.9 HFTV2.hs

The following is a snippet of code for the HFT agent in experiment 2.
The complete file can be found in market-sim/Agents/Misc/HFTV2.hs

hFTV2Wrapper :: Agent_t
-- Empty state case
hFTV2Wrapper (Emptyagentstate) args ((time, messages, broadcasts) : rest) myID
= (([Hiaton]) -- Output a hiaton on emptyagentstate
: (hFTV2Wrapper (HFTV1State initialState) args rest myID)) -- Update state to initialstate of type [int]
where
initialState = HFTV1StateRecord {_hClOrdIDs = [1..]} -- Initial state is set to [1,2,3...]
, _BYX1Bids = Map.fromList [(0,[]), Bids] -- Map of bids at each time BYX1
, _BYX1Asks = Map.fromList [(0,[]), Asks] -- Map of asks at each time BYX1
, _BYX2Bids = Map.fromList [(0,[]), Bids] -- Map of bids at each time BYX2
, _BYX2Asks = Map.fromList [(0,[]), Asks] -- Map of asks at each time BYX2
, _BrokerOrdSize = 1200 -- Total market order size
, _TradeExecutions = Map.fromList [(0,0,0,0))]

-- trace ("HFT State at time " ++ show time ++ ": " ++ (showBYXOrderlist (fromJust (Map.lookup time (_BYX1Bids updatedState) )) )
-- Non empty state to next state
hFTV2Wrapper (HFTV1State currentState) args
Appendix B. Simulator Code

```
((time, messages, broadcasts) : rest) myID
= ( output : ( hFTV2Wrapper (HFTV1State updatedState) args rest myID) )
where
(updatedState, output) -- Updated state, output message determined by time
| time >= 0 = hFTLogic currentState args time myID broadcasts messages
| otherwise = (currentState, [])

-- Defines HFT Logic
-- For debugging
--trace ("Trade execution at time : " ++ show time ++ " is " ++ show (fromJust (Map.lookup time (_TradeExecutions updatedState)) ))
-- trace ("Shares remaining at time : " ++ show time ++ " is " ++ show (_BrokerOrdSize updatedState) )
-- trace (whatAsksBrokerSaw myID args time tradeExTime)
-- trace ("Broker Saw: " ++ (showBYXOrderlist brokerAsks1) ++ " when making trade " ++ show (tradeExTime) ++ " Next order is going to " ++ nextOrdTo ++ " BYX1 best ask: " ++ show byx1best ++ " BYX2 best ask: " ++ show byx2best)

hFTLogic state args time myID broadcasts messages = (updatedState, outputMessages)
where
outputMessages = tradeSeenMessage ++ orders

updatedState = state { _BYX1Bids = byx1BidMap -- Map of bids at each time BYX1
  , _BYX1Asks = byx1AskMap -- Map of asks at each time BYX1
  , _BYX2Bids = byx2BidMap -- Map of bids at each time BYX2
  , _BYX2Asks = byx2AskMap -- Map of asks at each time BYX2
  , _TradeExecutions = tradeExMap -- Stores details of trade executed at this time
  , _BrokerOrdSize = finalRemainingShares
  , _hClOrdIDs = drop (length orders) currentClOrdIDList
}

currentClOrdIDList = _hClOrdIDs state -- returns the ClientOrderID list of the state
byx1BidMap = Map.insert time (getBYXOrderList broadcasts args "BYX1" Bids) (_BYX1Bids state) -- Update maps at each timestep
byx1AskMap = Map.insert time (getBYXOrderList broadcasts args "BYX1" Asks) (_BYX1Asks state)
byx2BidMap = Map.insert time (getBYXOrderList broadcasts args "BYX2" Bids) (_BYX2Bids state)
byx2AskMap = Map.insert time (getBYXOrderList broadcasts args "BYX2" Asks) (_BYX2Asks state)
orders = [executeOrder zipState args | zipState <- zip (_hClOrdIDs state) frOrders] --Execute if trade execution message is received and broker still has shares left
frOrders = createFrOrders frOppSeen bestLocalAskPrice bestLocalAskLiquidity bestPriceOtherExch finalRemainingShares
-- Depending on front running opportunity, create frontrunning orders
finalRemainingShares -- Updates the broker’s shares remaining, Takes into account debug received from other exchanges
| debugSeen = newRemainingShares - sharesExecuted
| otherwise = newRemainingShares
(debugSeen, frOppSeen, sharesExecuted, bestPriceOtherExch) = checkDebugs messages
-- received debug, genuine front running opportunity, shares executed on other exchange, best ask price on other exchange
tradeSeenMessage -- Decide whether or not to send debug message and what to put in the message
| getShares tradeExTime > 0 = [genTradeSeenMessage myID args tradeExTime frOpp bestLocalAskPrice] 
| otherwise = []
frOpp -- Check for a possible front running opportunity
```
getShares tradeExTime > 0 && newRemainingShares > 0 && nextOrdTo /= imAt = trace ("Possible Front Running opportunity at " ++ nextOrdTo ++ " spotted at time " ++ show time) True
otherwise = False

nextOrdTo = nextOrderIsGoingTo brokerAsks1 brokerAsks2 myID args -- based on ask books, where brokerAsks1, brokerAsks2 = whatBrokerSaw myID args time sendTime state -- returns the ask books from both brokers
we assume broker only sends buy order

sendTime = whenOrderWasSent myID args time tradeExTime -- If a trade was spotted on local exchange, get the time it was sent
newRemainingShares = currentRemainingShares - (getShares tradeExTime) -- Remaining shares after subtracting local exchange current trade

tradeExMap = Map.insert time tradeExTime (_TradeExecutions state) -- Update trade execution map
tradeExTime = getTradeExecution broadcasts myID args -- Trade execution at current time

currentRemainingShares = _BrokerOrdSize state -- returns the ClientOrderID list of the state
bestLocalAskPrice | isNothing (olBYX_getLowestPrice (olBYX_clean (getBYXOrderList broadcasts args imAt Asks))) = 0
| otherwise = fromJust (olBYX_getLowestPrice (olBYX_clean (getBYXOrderList broadcasts args imAt Asks)))
bestLocalAskLiquidity | isNothing (olBYX_peekFirstOrderSize (olBYX_clean (getBYXOrderList broadcasts args imAt Asks))) = 0
| otherwise = fromJust (olBYX_peekFirstOrderSize (olBYX_clean (getBYXOrderList broadcasts args imAt Asks)))
imAt = colocatedAt myID args -- Where current HFT is colocated

-- Scans through broadcasts and gets the Bid/Ask Order Lists from a given exchange
-- Takes in Broadcast list, args, exchange name, ask/bid
-- Outputs the broadcast message that represents the requested list of orders from the requested exchange
getBYXOrderList ((RoutedBroadcast (from,to) (Broadcastmessage _ (BYXOrderlistBroadcast byxordlst@(_,oLstSide)))): xs) args exchLabel side
| from == exchID && side == oLstSide = byxordlst
| otherwise = getBYXOrderList xs args exchLabel side
where
(AgentID exchID) = getCorrespondingAgentIdentifier (AgentLabel exchLabel)
getCorrespondingAgentIdentifier = returnGetCorrespondingAgentIdentifierFunction args
getBYXOrderList (_ : xs) args exchLabel side = getBYXOrderList xs args exchLabel side
getBYXOrderList [] _ _ side = ([], side)

-- Outputs ask books seen by the Broker given a sendTime of an order
-- Takes into account delays between HFT, the two exchanges and the broker
whatBrokerSaw myID args curtime sendTime state = (byx1Asks, byx2Asks)
where
byx1Asks = fromJust (Map.lookup receiveTime1 (_BYX1Asks state) )
byx2Asks = fromJust (Map.lookup receiveTime2 (_BYX2Asks state) )
receiveTime1
| colocatedAt myID args == "BYX1" = (sendTime - 1 - delay "BYX1" "Broker") + (1 + delay "BYX1" "HFT1")
| colocatedAt myID args == "BYX2" = (sendTime - 1 - delay "BYX1" "Broker") + (1 + delay "BYX1" "HFT2")
receiveTime2
| colocatedAt myID args == "BYX1" = (sendTime - 1 - delay "BYX2" "Broker") + (1 + delay "BYX2" "HFT1")
| colocatedAt myID args == "BYX2" = (sendTime - 1 - delay "BYX2" "Broker") + (1 + delay "BYX2" "HFT2")
Appendix B. Simulator Code

-- Outputs either BYX1 or BYX2 depending on where the next best price is
-- Gets rid of the best order on the local exchange and finds out where the next best ask is

nextOrderIsGoingTo brokerseen_byx1Asks brokerseen_byx2Asks myID args = goTo

where

goTo

| \( \text{olBYX\_getLowestPrice (olBYX\_clean byx1Asks\_postEx)} \leq \text{olBYX\_getLowestPrice (olBYX\_clean byx2Asks\_postEx)} \) = "BYX1"
| \( \text{olBYX\_getLowestPrice (olBYX\_clean byx2Asks\_postEx)} < \text{olBYX\_getLowestPrice (olBYX\_clean byx1Asks\_postEx)} \) = "BYX2"

byx1Asks\_postEx

| \( \text{colcatedAt myID args == "BYX1" = olBYX\_popBestOrderList brokerseen_byx1Asks} \)
| \( \text{colcatedAt myID args == "BYX2" = brokerseen_byx1Asks} \)

byx2Asks\_postEx

| \( \text{colcatedAt myID args == "BYX1" = brokerseen_byx2Asks} \)
| \( \text{colcatedAt myID args == "BYX2" = olBYX\_popBestOrderList brokerseen_byx2Ask} \)
Appendix C

Appendix C. Simulator Trace Files

This appendix contains trace files for each of the experiments. A trace file is a log that is produced each time the simulator is run. It details all messages sent and received amongst agents in the simulation.

C.1 frExp1-trace

The following is a portion of the trace file for a simulation of experiment 1 where the front-running attempt is successful.

The complete file can be found in market-sim/SimulationOutput/frExp1/frExp1-trace

START OF SIMULATION – seed is default
NB System Time 2 aligns with Exchange Time 1

System Time: 0
Messages to sim: ,

END OF STATE

 System Time: 1
 Messages to sim: ,

Broadcasts:
Routed Broadcast from/to agents (3,1):
Original broadcast: from/to group (3,2) [EmptyNBBO for (Symbol: "TheTradedInstrument") with (Timestamp: 0)]
Appendix B. Simulator Trace Files

Routed Broadcast from/to agents (3,2):
Original broadcast: from/to group (3,2) [EmptyNBBO for (Symbol: "TheTradedInstrument") with (Timestamp: 0)]

END OF STATE

-----------------------------

Broadcasts:
Routed Broadcast from/to agents (2,4):
Original broadcast: from/to group (2,3) [<EMPTY ORDER LIST (Asks)>]
Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,3) [<EMPTY ORDER LIST (Bids)>]
Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,3) [<EMPTY ORDER LIST (Asks)>]
Routed Broadcast from/to agents (2,4):

END OF STATE

-----------------------------

System Time: 3
Broadcasts:
Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,3) [<EMPTY ORDER LIST (Asks)>]
Routed Broadcast from/to agents (2,4):
Original broadcast: from/to group (2,3) [<EMPTY ORDER LIST (Bids)>]
Routed Broadcast from/to agents (1,4):
END OF STATE

-----------------------------

System Time: 5
Messages to sim:
Message from/to (4,1) BYXMessage (BATSNewOrderMessage) where:

---------------------- BATSNewOrderMessage Mandatory Fields: ----------------------
Sending Time: Just (SendingTime_FTVal 4) Symbol: Just (Symbol_FTVal "APPL")
Appendix B. Simulator Trace Files

Client Order ID: Just (ClOrdID_FTVal 4 2) Side: Just Side_FTVal_Sell
Order Quantity: Just (OrderQty_FTVal 400) Order Type: Just OrdType_FTVal_Limit
Routing Instruction: Just RoutingInst_BTVal_BookOnly

---------------------- BATSNewOrderMessage Optional Fields: -----------------------
Price: Just (Price_FTVal 24.0) Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_GTC Expiration Time: Just (ExpireTime_FTVal 60)

Message from/to (4,1) BYXMessage (BATSNewOrderMessage) where:

---------------------- BATSNewOrderMessage Mandatory Fields: ----------------------
Sending Time: Just (SendingTime_FTVal 4) Symbol: Just (Symbol_FTVal "APPL")
Client Order ID: Just (ClOrdID_FTVal 4 1) Side: Just Side_FTVal_Buy
Order Quantity: Just (OrderQty_FTVal 200) Order Type: Just OrdType_FTVal_Limit
Routing Instruction: Just RoutingInst_BTVal_BookOnly

---------------------- BATSNewOrderMessage Optional Fields: -----------------------
Price: Just (Price_FTVal 23.95) Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_GTC Expiration Time: Just (ExpireTime_FTVal 60)

Message from/to (4,1) BYXMessage (BATSNewOrderMessage) where:

---------------------- BATSNewOrderMessage Mandatory Fields: ----------------------
Sending Time: Just (SendingTime_FTVal 4) Symbol: Just (Symbol_FTVal "APPL")
Client Order ID: Just (ClOrdID_FTVal 4 3) Side: Just Side_FTVal_Sell
Order Quantity: Just (OrderQty_FTVal 600) Order Type: Just OrdType_FTVal_Limit
Routing Instruction: Just RoutingInst_BTVal_BookOnly

---------------------- BATSNewOrderMessage Optional Fields: -----------------------
Price: Just (Price_FTVal 24.1) Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_GTC Expiration Time: Just (ExpireTime_FTVal 60)

Message from/to (4,2) BYXMessage (BATSNewOrderMessage) where:
Appendix B. Simulator Trace Files

---------------------- BATSNewOrderMessage Mandatory Fields: ----------------------
Sending Time: Just (SendingTime_FTVal 4)  Symbol: Just (Symbol_FTVal "APPL")
Client Order ID: Just (ClOrdID_FTVal 4 4)  Side: Just Side_FTVal_Buy
Order Quantity: Just (OrderQty_FTVal 200)  Order Type: Just OrdType_FTVal_Limit
Routing Instruction: Just RoutingInst_BTVal_BookOnly

---------------------- BATSNewOrderMessage Optional Fields: ----------------------
Price: Just (Price_FTVal 23.95)  Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_GTC  Expiration Time: Just (ExpireTime_FTVal 60)

C.2 frExp1a-trace

The following is a portion of the trace file for a simulation of experiment 2 where the front-running attempt is successful.

The complete file can be found in market-sim/SimulationOutput/frExp1a/frExp1a-trace

Message from/to (4,2) BYXMessage (BATSNewOrderMessage) where:

---------------------- BATSNewOrderMessage Mandatory Fields: ----------------------
Sending Time: Just (SendingTime_FTVal 4)  Symbol: Just (Symbol_FTVal "APPL")
Client Order ID: Just (ClOrdID_FTVal 4 4)  Side: Just Side_FTVal_Buy
Order Quantity: Just (OrderQty_FTVal 200)  Order Type: Just OrdType_FTVal_Limit
Routing Instruction: Just RoutingInst_BTVal_BookOnly

---------------------- BATSNewOrderMessage Optional Fields: ----------------------
Price: Just (Price_FTVal 23.95)  Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_GTC  Expiration Time: Just (ExpireTime_FTVal 60)

Message from/to (4,2) BYXMessage (BATSNewOrderMessage) where:

---------------------- BATSNewOrderMessage Mandatory Fields: ----------------------
Sending Time: Just (SendingTime_FTVal 4)  Symbol: Just (Symbol_FTVal "APPL")
Client Order ID: Just (ClOrdID_FTVal 4 4)  Side: Just Side_FTVal_Buy
Order Quantity: Just (OrderQty_FTVal 200)  Order Type: Just OrdType_FTVal_Limit
Routing Instruction: Just RoutingInst_BTVal_BookOnly

Message from/to (4,2) BYXMessage (BATSNewOrderMessage) where:
Appendix B. Simulator Trace Files

BATSNewOrderMessage Optional Fields:  -------------------------------------
Price: Just (Price_FTVal 24.05) Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_GTC Expiration Time: Just (ExpireTime_FTVal 60)

Broadcasts:
Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,3) [<EMPTY ORDER LIST (Asks)>]

System Time: 6
Messages to sim:
Message from/to (1,3) CQS Short Quote Message (ExchangeBBO) where:
Quote Condition: EligibleQuote
Best Bid: Just 23.95 Best Bid Size: Just 200
Best Offer: Just 24.0 Best Offer Size: Just 400]

Message from/to (2,3) CQS Short Quote Message (ExchangeBBO) where:
Quote Condition: EligibleQuote
Best Bid: Just 23.95 Best Bid Size: Just 200
Best Offer: Just 24.05 Best Offer Size: Just 200]
Appendix B. Simulator Trace Files

--- (Start) BATSOrderAcknowledgmentMessage: ---

Acknowledgment for order with Client Order ID: Just (ClOrdID_FTVal 4 1).
BYX Order ID: Just (OrderID_FTVal 1 2) Order Quantity: Just (OrderQty_FTVal 200)
--- (End) BATSOrderAcknowledgmentMessage: ---

Message from/to (1,4) BYXMessage (BATSOrderAcknowledgmentMessage) where:

--- (Start) BATSOrderAcknowledgmentMessage: ---

Acknowledgment for order with Client Order ID: Just (ClOrdID_FTVal 4 2).
BYX Order ID: Just (OrderID_FTVal 1 1) Order Quantity: Just (OrderQty_FTVal 400)
--- (End) BATSOrderAcknowledgmentMessage: ---

Broadcasts:

Routed Broadcast from/to agents (1,3):
Original broadcast: from/to group (1,1) [
BBO for (Symbol: "Security on BYX Exchange with Simulator ID: 1") with (Exchange Timestamp: 5) and
BB Price: Just 23.95 BB Size: Just 200
BO Price: Just 24.0 BO Size: Just 400
Quote Condition: EligibleQuote
]

Routed Broadcast from/to agents (2,3):
Original broadcast: from/to group (2,1) [
BBO for (Symbol: "Security on BYX Exchange with Simulator ID: 2") with (Exchange Timestamp: 5) and
BB Price: Just 23.95 BB Size: Just 200
BO Price: Just 24.05 BO Size: Just 200
Quote Condition: EligibleQuote
]

Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,3) [<START ORDER LIST (Asks)>
At Price 24.0:
{ [OrderID: 1] [From simID: 4] [Received (time, fractime): (5, 1)] [Just (Symbol_FTVal "APPL")]
[Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 400)] [Just RoutingInst_BTVal_BookOnly]
[Just (Price_FTVal 24.0)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing]
}

At Price 24.1:
Appendix B. Simulator Trace Files

{ [OrderID: 3] [From simID: 4] [Received (time, fractime): (5, 3)] [Just (Symbol_FTVal "APPL")] 
  [Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 600)] [Just RoutingInst_BTVal_BookOnly] 
  [Just (Price_FTVal 24.1)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing] 
} <END ORDER LIST (Asks)>

Routed Broadcast from/to agents (2,4):

Original broadcast: from/to group (2,3) [<START ORDER LIST (Bids)>

At Price 23.95:

{ [OrderID: 1] [From simID: 4] [Received (time, fractime): (5, 1)] [Just (Symbol_FTVal "APPL")] 
  [Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 200)] [Just RoutingInst_BTVal_BookOnly] 
  [Just (Price_FTVal 23.95)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing] 
} <END ORDER LIST (Bids)>

System Time: 12

Messages to sim:
Message from/to (5,1) BYXMessage (BATSNewOrderMessage) where:

---------------------------------- BATSNewOrderMessage Mandatory Fields: ----------------------------------
Sending Time: Just (SendingTime_FTVal 10)  Symbol: Just (Symbol_FTVal "APPL")
Client Order ID: Just (ClOrdID_FTVal 5 1)  Side: Just Side_FTVal_Buy
Order Quantity: Just (OrderQty_FTVal 400)  Order Type: Just OrdType_FTVal_Market
Routing Instruction: Just RoutingInst_BTVal_BookOnly

---------------------------------- BATSNewOrderMessage Optional Fields: -------------------------------
Price: Just (Price_FTVal 0.0)  Execution Instructions: Empty
Time In Force: Just TimeInForce_FTVal_IOC  Expiration Time: Just (ExpireTime_FTVal 0)

C.3 frExp2-trace

The following is a portion of the trace file for a simulation of experiment 3.
The complete file can be found in market-sim/SimulationOutput/frExp2/frExp2-trace

Message from/to (1,7) BYXMessage (BATSOrderExecutionMessage) where:

---------------------------------- (Start) BATSOrderExecutionMessage: ----------------------------------
Appendix B. Simulator Trace Files

BYX Order ID: OrderID_FTVal 1 6 Client Order ID: ClOrdID_FTVal 7 1
Contra Agent ID: ContraBroker_FTVal 7 Execution Type: ExecType_FTVal_Fill
Execution Price: LastPx_FTVal 24.09 Execution Amount: LastShares_FTVal 200
Shares Remaining: LeavesQty_FTVal 0

------------------------ (End) BATSOrderExecutionMessage: -----------------------------

Message from/to (1,7) BYXMessage (BATSOrderAcknowledgmentMessage) where:

------------------------ (Start) BATSOrderAcknowledgmentMessage: -----------------------------
Acknowledgment for order with Client Order ID: Just (ClOrdID_FTVal 7 2).
BYX Order ID: Just (OrderID_FTVal 1 5) Order Quantity: Just (OrderQty_FTVal 200)
------------------------ (End) BATSOrderAcknowledgmentMessage: -----------------------------

Broadcasts:
Routed Broadcast from/to agents (2,4):
Original broadcast: from/to group (2,3) [<START ORDER LIST (Bids)>
At Price 23.95:
{ [OrderID: 3] [From simID: 4] [Received (time, fractime): (5, 3)] [Just (Symbol_FTVal "APPL")]
  [Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 200)] [Just Routing]
  [Just (Price_FTVal 23.95)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing]
}
<END ORDER LIST (Bids)>]
Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,4) [

------------------------ (Start) BYXTradeExecutionBroadcast: ------------------------
Agent IDs: (7, 7) Execution Price: 24.09 Execution Amount: 200
------------------------ (End) BYXTradeExecutionBroadcast: ------------------------

Routed Broadcast from/to agents (2,4):
Original broadcast: from/to group (2,3) [<START ORDER LIST (Asks)]
At Price 24.1:
{ [OrderID: 1] [From simID: 4] [Received (time, fractime): (5, 1)] [Just (Symbol_FTVal "APPL")]
  [Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 600)] [Just Routing]
  [Just (Price_FTVal 24.1)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing]
}
<END ORDER LIST (Asks)>]
Routed Broadcast from/to agents (1,4):
Original broadcast: from/to group (1,3) [<START ORDER LIST (Bids)]
Appendix B. Simulator Trace Files

At Price 23.95:

\{
  \[\text{OrderID: 3}\] \[\text{From simID: 4}\] \[\text{Received (time, fractime): (5, 3)}\] \[\text{Just (Symbol\_FTVal \"APPL\")}\]
  \[\text{Original Sending Time: Just (SendingTime\_FTVal 4)}\] \[\text{Just (OrderQty\_FTVal 200)}\] \[\text{Just RoutingInst\_BTVal BookOnly}\]
  \[\text{Just (Price\_FTVal 23.95)}\] \[\text{Just TimeInForce\_FTVal GTC}\] \[\text{Just (ExpireTime\_FTVal 60)}\] \[\text{Nothing}\]
\}

<END ORDER LIST (Bids)>]

Routed Broadcast from/to agents (1,4):

Original broadcast: from/to group (1,3) [<START ORDER LIST (Asks)>]

At Price 24.1:

\{
  \[\text{OrderID: 1}\] \[\text{From simID: 4}\] \[\text{Received (time, fractime): (5, 1)}\] \[\text{Just (Symbol\_FTVal \"APPL\")}\]
  \[\text{Original Sending Time: Just (SendingTime\_FTVal 4)}\] \[\text{Just (OrderQty\_FTVal 600)}\] \[\text{Just RoutingInst\_BTVal BookOnly}\]
  \[\text{Just (Price\_FTVal 24.1)}\] \[\text{Just TimeInForce\_FTVal GTC}\] \[\text{Just (ExpireTime\_FTVal 60)}\] \[\text{Nothing}\]
\}

<END ORDER LIST (Asks)>]

Routed Broadcast from/to agents (2,5):

Original broadcast: from/to group (2,3) [<START ORDER LIST (Bids)>]

At Price 23.95:

\{
  \[\text{OrderID: 3}\] \[\text{From simID: 4}\] \[\text{Received (time, fractime): (5, 3)}\] \[\text{Just (Symbol\_FTVal \"APPL\")}\]
  \[\text{Original Sending Time: Just (SendingTime\_FTVal 4)}\] \[\text{Just (OrderQty\_FTVal 200)}\] \[\text{Just RoutingInst\_BTVal BookOnly}\]
  \[\text{Just (Price\_FTVal 23.95)}\] \[\text{Just TimeInForce\_FTVal GTC}\] \[\text{Just (ExpireTime\_FTVal 60)}\] \[\text{Nothing}\]
\}

<END ORDER LIST (Bids)>]

Routed Broadcast from/to agents (1,5):

Original broadcast: from/to group (1,4) [

--------------------------- BYXTradeExecutionBroadcast: ---------------------------

Agent IDs: (7, 7) Execution Price: 24.09 Execution Amount: 200

---------------------------

Routed Broadcast from/to agents (2,5):

Original broadcast: from/to group (2,3) [<START ORDER LIST (Asks)>]

At Price 24.1:

\{
  \[\text{OrderID: 1}\] \[\text{From simID: 4}\] \[\text{Received (time, fractime): (5, 1)}\] \[\text{Just (Symbol\_FTVal \"APPL\")}\]
  \[\text{Original Sending Time: Just (SendingTime\_FTVal 4)}\] \[\text{Just (OrderQty\_FTVal 600)}\] \[\text{Just RoutingInst\_BTVal BookOnly}\]
  \[\text{Just (Price\_FTVal 24.1)}\] \[\text{Just TimeInForce\_FTVal GTC}\] \[\text{Just (ExpireTime\_FTVal 60)}\] \[\text{Nothing}\]
\}

<END ORDER LIST (Asks)>]

Routed Broadcast from/to agents (1,5):

Original broadcast: from/to group (1,3) [<START ORDER LIST (Bids)>]
Appendix B. Simulator Trace Files

At Price 23.95:

\[
\{ \text{OrderID: 3} \text{ From simID: 4} \text{ Received (time, fractime): (5, 3)} \text{ Just (Symbol_FTVal "APPL")}\}
\]
\[
\text{Original Sending Time: Just (SendingTime_FTVal 4)} \text{ Just (OrderQty_FTVal 200)} \text{ Just RoutingInst_BTVal_BookOnly}
\]
\[
\text{Just (Price_FTVal 23.95)} \text{ Just TimeInForce_FTVal_GTC} \text{ Just (ExpireTime_FTVal 60)} \text{ Nothing}
\]

<END ORDER LIST (Bids)>]

Routed Broadcast from/to agents (1,5):

Original broadcast: from/to group (1,3) [<START ORDER LIST (Asks)]

At Price 24.1:

\[
\{ \text{OrderID: 1} \text{ From simID: 4} \text{ Received (time, fractime): (5, 1)} \text{ Just (Symbol_FTVal "APPL")}\}
\]
\[
\text{Original Sending Time: Just (SendingTime_FTVal 4)} \text{ Just (OrderQty_FTVal 600)} \text{ Just RoutingInst_BTVal_BookOnly}
\]
\[
\text{Just (Price_FTVal 24.1)} \text{ Just TimeInForce_FTVal_GTC} \text{ Just (ExpireTime_FTVal 60)} \text{ Nothing}
\]

<END ORDER LIST (Asks)>]

Routed Broadcast from/to agents (2,6):

Original broadcast: from/to group (2,3) [<START ORDER LIST (Bids)]

At Price 23.95:

\[
\{ \text{OrderID: 3} \text{ From simID: 4} \text{ Received (time, fractime): (5, 3)} \text{ Just (Symbol_FTVal "APPL")}\}
\]
\[
\text{Original Sending Time: Just (SendingTime_FTVal 4)} \text{ Just (OrderQty_FTVal 200)} \text{ Just RoutingInst_BTVal_BookOnly}
\]
\[
\text{Just (Price_FTVal 23.95)} \text{ Just TimeInForce_FTVal_GTC} \text{ Just (ExpireTime_FTVal 60)} \text{ Nothing}
\]

<END ORDER LIST (Bids)>]

Routed Broadcast from/to agents (1,6):

Original broadcast: from/to group (1,4) [------------------------------ BYXTradeExecutionBroadcast: -------------------------------

\text{Agent IDs: (7, 7) Execution Price: 24.09 Execution Amount: 200}

------------------------------ BYXTradeExecutionBroadcast: -------------------------------]

Routed Broadcast from/to agents (1,7):

Original broadcast: from/to group (1,4) [------------------------------ BYXTradeExecutionBroadcast: -------------------------------

\text{Agent IDs: (7, 7) Execution Price: 24.09 Execution Amount: 200}

------------------------------ BYXTradeExecutionBroadcast: -------------------------------]

<END ORDER LIST (Bids)>]

Routed Broadcast from/to agents (1,8):

Original broadcast: from/to group (1,4) [------------------------------ BYXTradeExecutionBroadcast: -------------------------------

\text{Agent IDs: (7, 7) Execution Price: 24.09 Execution Amount: 200}

------------------------------ BYXTradeExecutionBroadcast: -------------------------------]
Appendix B. Simulator Trace Files

Routed Broadcast from/to agents (2,8):

Original broadcast: from/to group (2,3) [<START ORDER LIST (Asks)>

At Price 24.1:

{ [OrderID: 1] [From simID: 4] [Received (time, fractime): (5, 1)] [Just (Symbol_FTVal "APPL")]

[Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 600)] [Just Routing]

[Just (Price_FTVal 24.1)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing]

}<END ORDER LIST (Asks)>]

Routed Broadcast from/to agents (1,8):

Original broadcast: from/to group (1,3) [<START ORDER LIST (Bids)>

At Price 23.95:

{ [OrderID: 3] [From simID: 4] [Received (time, fractime): (5, 3)] [Just (Symbol_FTVal "APPL")]

[Original Sending Time: Just (SendingTime_FTVal 4)] [Just (OrderQty_FTVal 200)] [Just Routing]

[Just (Price_FTVal 23.95)] [Just TimeInForce_FTVal_GTC] [Just (ExpireTime_FTVal 60)] [Nothing]}

}
Bibliography


