Latency and Asset Prices[*]

Andrei Kirilenko—MIT Sloan School of Management
Guilherme Lamacie—BM&FBOVESPA

January 7, 2015

ABSTRACT

We measure message processing time or latency inside an automated trading platform. We show that latency is a random variable that has a strong predictive power over both volatility and the volatility of volatility of a highly liquid asset over and above changes in message traffic. We argue that in automated markets, processing time contains valuable nontrade information about the price formation process. We recommend that automated trading platforms improve pre–trade price transparency by reporting characteristics of latency to market participants on an ongoing basis along with order book events, transaction prices, and trading volume.

[*] We thank Thalita Franklin and Yurie Yassunaga Suzuki for outstanding research assistance. We also thank the IT team for excellent technological support. The views presented in this paper are our own and do not represent an official position of any entity, its management or staff.

Latency is the delay between a signal and a response. It is measured in units of time: seconds, milliseconds, microseconds and so forth. Latency is an essential feature of any nontrivial mechanical or electronic system. The system of interest to us is an automated trading platform. In this paper, we measure latency inside an automated trading platform and show that latency is a random variable that has a strong predictive power over both volatility and the volatility of volatility of a highly liquid asset.

In automated financial markets, there are three main types of latency that affect the trading process: communication latency, market feed latency, and trading platform latency. Trading platform latency is the time it takes for a message--a standardized packet of data that traders use to communicate with automated trading platforms (e.g., stock exchanges, derivatives exchanges or dark pools)—to travel inside the platform. Unlike communication latency and market feed latency[1], there is little a trader can do about this type of latency. From a point of view of a trader, trading platform latency is exogenous. Typically, it cannot be influenced by an individual trader and must be taken as given.[2]

The focus of this paper is on trading platform latency and its relation to asset prices. Intuitively, trading platform latency should matter for asset prices at least at the extreme. If trading platform latency becomes extremely large or extremely volatile, alarms will go off inside the traders' systems. Subsequent automated or human interventions can result in changes in trading behavior, including exiting existing positions, trading much less or waiting out the perceived market turbulence. Thus, very large delays in the processing of messages should have an effect on changes in asset prices. But, what about the relation between the level and variability of latency and changes in asset prices on an ongoing basis? This is an empirical question that can be answered if trading platform latency is measured and then statistically interacted with the relevant characteristics of asset prices.

We present measurements of different components of trading platform latency within a modern automated trading platform, the BM&FBOVESPA Exchange. Latency measurements for the front-month Mini Ibovespa Futures, the most liquid index futures contract traded exclusively on the the BM&FBOVESPA, were collected on March 13, 2014. Latency measurements were collected by hardware devices that listen to all message traffic flowing through different parts of the platform. The devices themselves do

[1] Communication latency is the time it takes for a message to travel between a traders computer and an automated trading platform. In order to reduce communication latency, traders can, for a fee, locate their trading servers as close as possible to the "matching engine" of a trading venue and use the fastest data processing technology inside their co-located cage. Market feed latency is the time it takes for an automated trading platform to disseminate market data bids and offers, as well as executed transactions to all market participants. To reduce market feed latency, market participants subscribe, for a fee, to data services provided by exchanges or third-parties.

[2] While individual traders cannot influence trading platform latency, operators of a trading platform can reduce this type of latency by investing in technology that speeds up the internal transmission and processing of data packets. For example, a trading platform can improve the quality of its cables and switches; optimize software to process messages faster; or build or acquire a faster matching engine.

not add additional latency; they just listen to messages as they fly by. Importantly, the identities of message–submitters or any other specific characteristics of messages are never revealed; only the metadata on message traffic.

We find that trading platform latency is not constant. It is a variable drawn from a distribution. In fact, trading platform latency is significantly more volatile than one may surmise; its (log) variations are not well described by a bell-shaped distribution. This means that when an exchange announces that its average trading platform latency is 3 milliseconds, it is not very meaningful. In practice, an automated trading platform can take as little as 800 microseconds to process a traders message or as much as 80 milliseconds (or 100 times higher than 800 microseconds).[3] Trading platform latency is just that variable.

We then show that intraday dynamics of median trading platform latency and the dispersion of latency add significant explanatory power to changes in volatility and the volatility of volatility. Notably, the effects of median latency and latency dispersion arise over and above changes in the number of messages (which we use as a control variable). Thus, what seems to matter is not how many messages a trading platform needs to process, but how long a trading platform takes to process certain types of messages. A possible explanation for this is that messages that result in changes of the best bid and offer take more time for a platform to process (the matching engine needs to adjust the "locations" of many queues) than messages that change the lengths of existing queues. At the same time, the very messages that result in the changes of the best bid and offer are certainly much more valuable in terms of the price formation process than those that add a bit of depth here and there, but don't move the midpoint price.

These empirical regularities can be present in the data for a number of reasons. One possibility is that some technologically–advanced traders monitor trading platform latency on an ongoing basis as a proxy for "the collective wisdom about the state of the market" and actively adjust their trading behavior to changes in latency and in the variability of latency which they view as predictive about future imbalances or orders. Or, as Ait–Sahalia and Saglam (2013) argue, technologically–advanced traders are "exploiting fleeting trading opportunities arising from the trading process itself."

Another possibility is that intraday dynamics of both trading platform latency and asset prices are driven by changes in one or several latent variables (e.g., "information", "liquidity", "sentiment") that first manifest themselves in the characteristics of latency and then in asset prices simply due to trading technology. For example, a change in "information" would give rise to a flurry of incoming messages that change the best bid and offer, which would give rise to an increase in latency because the matching engine has to readjust multiple queues, which, if executions took place, would in turn give rise to a small spike in recorded volatility.

However, we believe that irrespective of whether latency is a risk factor in its own right or an early proxy for market adjustments due to trading technology, if greater

[3] A microsecond is one–millionth of a second. A millisecond is one–thousandth of a second. There are 1,000 microseconds in one millisecond.

median delays and higher volatility of processing time inside trading platforms matter for asset prices, then characteristics of latency should be reported by the trading platforms to the market participants on an ongoing basis. Among other things, such reporting would address the criticism levied by Coval and Shumway (2001), who find that a rise in the sound level inside a trading pit conveys valuable information (i.e., they precede periods of elevated price volatility) and argue that "electronic exchanges will continue to be imperfect substitutes to open outcry trading as long as they cannot fully replicate exchange sound levels and the variables for which they are likely to proxy."

We do not attempt to fully replicate the analysis of Coval and Shumway (2001). For one thing, unlike face to face trading, which could in principle be conducted entirely with stylized hand signals, electronic trading can only be done by exchanging messages with trading platforms. In an automated environment, there is no alternative to messaging; messaging is the only way to participate in the price–setting process that takes place inside an automated trading platform. For another, the processing of messages inside a trading platform, which we describe further in detail, is completely technologically different from the initiation and processing of verbal or visual commands in a trading pit. Thus, characteristics of noise level and message traffic are related in spirit as they both represent valuable nonprice information that affects the price–setting process, but are not really directly comparable.

To support our argument, we choose to focus on the volatility and the volatility of volatility as relevant metrics for asset price dynamics. We do not analyze the statistical relations between latency and other indicators of market activity such as trading volume and depth, because these statistical relations arise due to purely technical reasons: it simply takes a lot more time for a trading platform to process an execution (several reference databases must be accessed) than any other message-related event; it also takes more time to process more executions that add up to higher trading volume.

Regulators and policymakers around the world have intuitively noticed that latency matters and that inability to operate in a low latency trading environment has a disproportionate impact on the least technologically savvy market participants. In response, regulators proposed a number of *ad hoc* measures that aim to affect both absolute latency (i.e., to "slow everyone down") and relative latencies (i.e., to "slow some traders down") faced by different market participants. For example, a measure called "minimum quote life" proposes to add latency to the time any resting order remains available for trading before it could be canceled. Another measure proposes to scramble the time priority of orders submitted by market participants by adding random latencies to their original order submission times. Yet another measure proposes to give latency priority to certain more desirable market participants ahead of others.

We believe that however well–intentioned these measures might be, if applied without good understanding of where latency comes from inside a trading platform and what determines its empirical properties, such measures could result in extra cost to the very market participants that they are designed to protect, as well as reduced informativeness and additional complexity of automated trading platforms. As a result, ad hoc latency measures may do exactly the opposite of what they are intended for–leveling the playing field for the less technologically–advanced market participants.

Instead, we recommend that automated trading platforms report characteristics of latency to market participants on an ongoing basis along with order book events, transaction prices and trading volume. That way, any valuable nontrade information contained in latency can be discovered directly along asset prices. As we show, the latency of every single submission, cancellation, modification or execution can be measured down to a microsecond. Together these measurements add up to something very informative about the dynamics of the market as a whole. Why not allow market participants to use this valuable information about the price formation process? It could further improve the pre–trade transparency of automated trading platforms. Moreover, if needed, financial instruments can be introduced to transfer latency risk from those who do not wish to hold it to those who wish to take it on. If volatility can be traded, so can latency.

The paper is organized as follows. Trading platform latency is described in Section I. Descriptive statistics of the different components of latency are presented in Section II. The analysis of the statistical relation between the intraday dynamics of trading platform latency and the characteristics of asset prices are in Section III. Our concluding remarks are in Section IV.

I. Trading Platform Latency

Trading platform latency arises due to the time it takes to process and route messages inside an automated trading platform. In order to explain where and how latency arises and where and how we measure it, we offer a stylized graphical representation of an automated trading platform in Figure 1.

<Insert Figure 1>

In the top left corner of Figure 1 are the entry ports for the clients, who can connect either via external network connections or via a co-located server. From the point of view of the clients, an entry port is where the trading platform physically starts. Cables, switches, hardware and software on the outer side of the entry ports belong to clients, their brokers an technology vendors. Cables, switches, and all other hardware and software components on the inner side of the entry ports belong to the trading platform. The main hardware and software components of a trading platform that we need to be cognizant about in this study from the point of view of latency measurement are the gateways and the matching engine. These components are graphically aligned along the top of Figure 1.

In addition to entry ports, other relevant components of an automated trading platform for the clients, brokers and technology vendors are drop-copy gateways and market data subsystems (they are stylistically presented in the bottom left corner of Figure 1). For the trading platform itself, other relevant components include audit–trail, surveillance, and regulatory reporting subsystems (labelled Audit Trail and other Sub–Systems). A multicast bus visually represented as running vertically through the automated trading platform allows packets of data to be dispatched to multiple sub–systems.

The key packet of data that travels through an automated trading platform is called a message. A message is a standardized packet of data that enables a trader and a trading platform to communicate with each other. It is a primitive unit of valuable information that enables all pre-trading, trading, and post-trading activity. For example, a message can be a directive from a trader to a trading platform to enter a new executable order or to cancel or modify an existing order. Similarly, a message can be a confirmation from a trading platform informing a trader that her directive has been received. Each message is time–stamped when it passes through a specific subsystem of an automated trading platform. This makes it possible to follow messages inside an automated trading platform to measure latency of its different components.

Messages arrive at the entry points and are directed to the gateways. Inside the gateways the messages are checked for completeness and then ran through pre–trade risk safeguards. Pre–trade risk safeguards are designed to identify and reject incomplete or erroneous messages before they become trading orders. Examples of pre–trade risk safeguards include asset–specific price and quantity bands to prevent a fat–finger error, firm–specific trading limits to guard against a firm's inability to pay, and trading operator–specific throttles to prevent an algo–gone–wild scenario. In other words, pre–trade risk safeguards are designed to cover against a variety of adverse contingencies: they are asset–specific (i.e., applicable to all who trade a given asset on a platform), firm or trader–specific or algorithm–specific. Pre–trade throttles that are specific to a particular firm, session, operator or algorithm require accessing one or several reference databases, checking against multiple contingencies, and executing various calculations and logical comparisons. This takes processing time. As a result, gateways processing comprises a substantial part of overall trading platform latency.

After the gateways, messages are transmitted to the matching engine. On their way to the matching engine, messages pass through the multicast bus, which enables other subsystems to record and process them. When a message reaches the matching engine, it is assigned a timestamp and further processed. The message can, for instance, represent a new executable limit buy order for which possible matching sell orders must be searched, thus possibly triggering further trading processing routines within the matching engine. Matching engine processing time varies depending on the complexity of an order type, number of other matching engine events that are being processed, and trading volume at a point in time. After a message is processed, the matching engine generates a confirmation that goes back to the gateway and ultimately to the original message submitter. This confirmation is called "execution report." This does not mean that a trade execution has taken place; it means that the trading platform has executed the instructions contained in the message that it has originally received. Execution report is a message back from the trading platform to a client.

In order to measure latency components, we had listening devices installed within the BM&FBOVESPA trading platform. In Figure 1, the listening devices are labelled TAP Inputs 1 and TAP Inputs 2. TAPs, which stands for test access ports, are hardware devices designed to listen to all message data flowing through different points of the network. TAPs themselves do not add additional latency; they do not process any messages; they just listen to and record the instances of messages passing by. The collected message traffic, which flows through multiple parallel routes within the platform, is appropriately compiled by monitoring devices, and then sent to a database where it is stored for subsequent analysis. The database contains message traffic metadata corresponding to all client sessions flowing through all of the entry ports, gateways and matching engines on a given trading day.

As can be seen in Figure 1, TAP Inputs 1 devices are installed after the entry ports into the trading platform and before the gateways. TAP Inputs 1 listen to the incoming message traffic passing from the clients to the gateways (forward) and to the outgoing message traffic from the gateways back to the clients (back). TAP Inputs 2 are installed after the gateways and before the matching engines. TAP Inputs 2 devices listen to the incoming message traffic data before it enters the matching engines and the outgoing confirmation messages generated by the matching engines. After appropriate synchronization to account for parallel processing, we are able to measure the latency of all message traffic that passes between two sets of TAP Inputs.

Trading platform latency consists of three parts: forward (FWD), matching engine (ENG), and back (BWD). The first part is the time it takes for a message to travel from an entry port through the gateway (incoming messages between TAP Inputs 1 and TAP Inputs 2). Recall that gateway latency arises because each message needs to be processed by the pre-trade risk safeguards for completeness and evaluated for the application of message throttling and trading limits.

The second part of trading platform latency is the time it takes for a message to be processed by the matching engine. It can be measured by the time it takes for a message to crosses TAP Inputs 2 the first time and the time when it returns to TAP Inputs 2 (in the form of "execution report") after being processed by the matching engine. Matching engine latency arises because the engine needs to process each message in relation to the state it is in and adjust its state, which could in some cases be quite calculation (and, thus, time) intensive.

Lastly, the third part of trading platform latency is the time it takes for a message to travel from the matching engine back through the gateway and to an entry port (outgoing messages between TAP Inputs 2 and TAP Inputs 1). Outgoing latency arises because it takes the gateway a bit of time to update the risk limits for the cancellation and some modification messages (among others) before an outgoing message is able to proceed to an entry port (which on the way back functions as an exit port).

II. Descriptive Statistics of Latency

In this section, we present latency characteristics for the front-month Ibovespa index Mini-Futures, the most liquid index futures contract traded exclusively on the the BM&FBOVESPA. Latency measurements for all 818,359 messages for the front-month contract maturing in April 2014 (symbol WINJ14) were collected on March 13, 2014, which was an ordinary trading day.

A. Message Data

From every message that passes by the TAPs, we record and save key source data fields: Security, Timestamp (HH:MM:SS.000.000), OrderID, Action (Submit, Modify, Cancel), Latency 12, Latency ME, Latency 21. We further cross the data with high frequency information from the order book, including time-stamped mid-quotes.

B. Measuring Latency and Its Components

We compute statistics for three separate message types—submission, modification, and cancellation—and all message types together. For each message type and for all message types together, we analyze Round Trip Time (RTT) latency and its components, the Forward latency (FWD), the Matching Engine latency (ENG) and the Outgoing latency (BWD). Thus, for each separate message type and all messages combined, we have four sets of measurements (FWD, ENG, BWD, and their sum, RTT), which gives us 16 separate latency datasets to analyze. Each latency component is measured in microseconds for 370,342 submissions, 248,203 modifications and 199,814 cancellations, with a total sum of 818,359 messages.

Table I presents summary statistics.

<Insert Table I>

According to Table I, there is some difference between the medians and means of the RTT latencies for the three order types. Median RTT for cancellation messages is 1,086 microseconds; for new submission messages, it is 1,205 microseconds; and for modification messages, it is 1,488 microseconds. This makes intuitive sense as it should take less time for a trading platform to cancel an existing order (since it already knows what to cancel), then to process a new submission (it needs to run the pre–trade safeguards), then to modify (it needs to both cancel an existing order and generate a modified order). This is reflected in the medians of three components of the RTT latency.

At the same time, the variability of latency is ordered in the opposite direction. Standard deviation of the RTT for cancellation messages is 18,575 microseconds; for a new submission message, it is 11,899 microseconds; and for a modification message, it is 1,597 microseconds. Taking a closer look, reveals that the main contributor to the standard deviation of the RTT latency for all message types is FWD latency. FWD latency is the highest for cancellations (at 18,545 microseconds). It is closely followed by new submissions (at 11,427 microseconds). It is an order of magnitude lower for modifications at 1,190 microseconds.

Summary statistics for the skewness and kurtosis of different message types, as well as their minimum and maximum recorded latencies strongly suggest that RTT latency and its components are not centered around a mean a median, and do not follow bell–shaped distributions. This is quite evident by observing the histograms for the RTT and different latency components of different message types, as well as all types together.

C. Histograms of Latency and Its Components

Figure 2 presents full and truncated histograms of RTT latency for all message types.

<Insert Figure 2>

The left panel of Figure 2 presents a full histogram of RTT latency for all message types. The right panel of the Figure presents a histogram of RTT latency for all message types truncated at 1,500 microseconds, the number between the mean (1,239 microseconds) and the median (1,792 microseconds) RTT latency for all message types. From the the histograms, it seems plausible that the RTT latency for all message types is a mixture of two distributions. One distribution is centered on a number somewhat to the right of the median and is almost bell–shaped. The other distribution is a power law distribution that describes the right tail (by definition latency is a nonnegative number).

In Figure 2, we fit a lognormal distribution to the RTT latency for all message types truncated at 1,500 microseconds and a power law distribution to the RTT latency that exceeds 1,500 microseconds. The threshold for the cutoff point was optimized by applying the Kolmogorov–Smirnov goodness–of–fit test.

<Insert Figure 3>

The two distributions capture the mixed nature of the data–generating process of the RTT latency for all messages types. At lower levels, RTT latency has a strong tendency to cluster around a central value, but after a certain cutoff point, RTT latency could become extremely large. Figure 4 illustrates that this empirical regularity is also present for RTT latency for different message types.

<Insert Figure 4>

Figure 5 presents the FWD component of latency for the three different order types.

<Insert Figure 5>

According to Figure 5, FWD latency for different order types below a cutoff point exhibits a strong tendency to cluster around two separate data points. It does lend itself well to being described by a smooth unimodal lognormal distribution. We conjecture that the shape of the distribution reflects the nature of throttling mechanisms that trigger during bursts, as well as by the queries and updates of risk limits by the pre–trade risk system.

Figure 6 presents the ENG component of latency for the three different order types.

<Insert Figure 6>

According to Figure 6, ENG latency for different order types below a cutoff point could be very well parametrized by a lognormal distribution, but also has a very large power law tail. We believe that the power law tail of the ENG latency arises due to the complexity of reflecting updates to different order types in the central limit order book.

Figure 7 presents the BWD component of latency for the three different order types.

<Insert Figure 7>

According to Figure 7, BWD latency for different order types below a cutoff point has a very strong tendency to cluster around 450 microseconds and a not very significant power law tail. We believe that the main reason for the strong clustering of BWD latency is the similarity in the amount of time it takes to update pre–trade risk limits prior to releasing the outgoing message.

Looking across all histograms, we could highlight the following empirical regularities. First, both RTT latency and latency components are described by distributions, not point estimates. Second, cancellations have the shortest median RTT latency, as well as the shortest median FWD and ENG latencies. Third, new submissions presented longer median FWD latency, while cancellations presented longer median BWD latency.

These empirical regularities are likely rooted in the message processing protocol inside the trading platform. For example, we would expect a trading platform to process cancellations faster on the way in because there no need for the application of the pre-trade safeguards (they have already been applied to whatever is being cancelled). We would also expect the matching engine to take the least amount of time to remove something form the central limit order book, because it just needs to remove an order rather than to figure out where to fit it in. And, we would expect the cancellation message to take longer on the way back, because the trading platform needs to update pre–trade limits upward before an execution report could go out.

The key question though is do any of these measurements have anything to do with the dynamics of asset prices.

III. Latency and Asset Prices

In this section we examine the statistical relation between the intraday dynamics of trading platform latency and the characteristics of asset prices: returns, volatility, and the volatility of volatility. As we eluded to earlier, the intraday dynamics of both trading platform latency and asset prices could be driven by changes in one or several latent factors that first manifest themselves in the increase of processing time and then in, say, elevated volatility.

A. Time Series Plots

Before we proceed with the formal econometric analysis, it might be helpful to visually examine the time series of RTT latency.

Figure 8 presents the times series of RTT latency for all message types (top panel) and RTT latency for cancellation messages separately (bottom panel).

<Insert Figure 8>

Each panel of the Figure contains three lines: the middle line is for the median RTT latency, the bottom line is for the 10th percentile, and the top line is for the 90th percentile of RTT latency measured for each non–overlapping 3–second interval starting at 9:00 am and ending at 5:55 pm local time.[4] Between 9:00 am and 5:55 pm, there are 10,700 non–overlapping 3–second intervals. Given the total number of messages of different types, a simple back of the envelope calculation suggests that in the course of each 3 second interval, there are on average 76 messages of all types, of which 35 are new submission messages, 23 are modification messages, and 18 are cancellation messages. For each 3–second interval, we sort the messages by their latency (from the smallest to the largest) and find the median, 10th smallest and 90th largest latency within the interval. The top panel of Figure 8 plots the time series of these three numbers for all messages, the bottom panel – for cancellation messages only.

The time series plots suggest that the RTT latency for all message types and, to some extent, cancellation messages mostly fluctuate within fairly narrow bands around the median value of about 1,100–1,200 microseconds, but also exhibit a significant number of spikes of different magnitude. The spikes in latency, some of which are an order of magnitude higher than the median, seem to have a memory – they increase, reach a peak, and then decrease.

[4] The actual trading day begins with an opening auction that starts at 8:55 am and lasts for five minutes and ends with a closing auction that starts at 5:55 pm and also lasts for five minutes. We exclude the 5 minutes of the opening auction and the 5 minutes of the closing auction from the data because the matching engine operates according to a different algorithm than the rest of the trading day.

There are also a lot more spikes in the RTT for all messages than in the RTT for cancellations only. RTT latency for cancellation messages only serves as a useful baseline case for the trading platform latency since cancellations are not expected not get stuck in the pre-trade gateways or trigger lots of processing in the matching engine. Cancellation messages do take a bit longer to process on the way back than new submission messages, but less time than modification messages.

To further investigate where the spikes in the RTT latency for all messages might be coming from, we decompose RTT latency for all messages into the forward, engine and backward components. Figure 9 presents the times series for the median, 10th smallest and 90th largest latency for each 3–second interval for the forward, engine and backward components of RTT latency for all messages.

<Insert Figure 9>

The time series plots in Figure 9 suggest that the spikes in the RTT latency for all message types originate almost entirely in the engine component of the RTT latency with the forward latency contributing just a few additional large spikes. Intuitively, significant increases in engine latency come from the readjustments of multiple queues due rapid movements of the best bid and offer prices. If in addition to processing these readjustments, a matching engine also has to account for a series of executions (e.g., an order walking the book), which also take time to process, then we would expect that significant changes in the state of the order book are also associated with the changes in the dynamics of latency.

However, significant changes in the state of the order book are, almost by definition, also associated with the significant changes in message traffic, not just the delay in processing. High messaging can naturally increase latency in different parts of the trading platform, especially in the pre-trade risk systems, in the matching engine and, in some cases, in generating execution reports.

Figure 10 presents the times series for the number of messages in each 3–second interval.

<Insert Figure 10>

The time series plot of the number of messages also exhibits a significant number of spikes of different magnitude, as well a possibility of memory. Thus, in order to establish that what matters for asset prices is the processing time for certain types of messages rather than the sheer increase in the number messages needed to be processed, we would need to control for the number of messages in conducting our formal econometric analysis.

B. The Econometric Approach

We now proceed to formally examine the statistical relation between the intraday dynamics of trading platform latency and the characteristics of asset prices. We focus on the following characteristics of the asset price process: returns (differences of log prices of mid-quotes), absolute returns (absolute differences of log prices of mid–quotes), volatility (range calculated as log difference between the highest and lowest mid–quotes), and volatility of volatility (a semi–parametric measure proposed by Wang, Kirby and Clark (2013)).

We compute the time series of these statistics for the 10,700 non–overlapping 3–second intervals—the plots of which are presented in Figure 11, Figure 12, and Figure 13—and statistically interact them with the median latency and the dispersion (log-difference between the 90th and 10th percentiles) of latency taken within the same intervals.

<Insert Figure 11>

<Insert Figure 12>

<Insert Figure 13>

Our econometric approach is as follows. First, we run linear regressions of price–based variables on (i) their lagged values and (ii) contemporaneous values of other price–based variables. We use the ARMAX model with the lags selected according to either Akaike or Bayesian information criteria (we check both). Second, we re–run the above regressions with the log number of messages added as a control variable. These two sets of regressions serve as baseline models: the first set of regressions creates a baseline for the amount of information that could be obtained from price–based variables alone; and the second set of regressions serves as a modified baseline model which also controls for changes in the message traffic. Adding the (log) number of messages also serves as a control for the delays due to trading intensity as it has a 0.81 correlation with the number of trades.

Then, we add to the baseline and modified baseline models sequentially four latency variables: (1) log median RTT for all messages, (2) log median RTT for cancellation messages only, (3) RTT dispersion for all messages (log difference between the 90th and 10th percentiles of RTT for all messages), and (4) RTT dispersion for cancellation messages only (log difference between the 90th and 10th percentiles of RTT for cancellation messages only). We re–run the regressions with the latency variables added in and test for the significance of regression coefficients on the added latency variables, as well as for the improvement in the goodness–of–fit (appropriately adjusted).

For concreteness, we illustrate actual regression specifications for the case of the Volatility of Volatility as a dependent variable and Range as an independent variable.

Baseline model: The Volatility of Volatility as a dependent variable, lagged Volatility of Volatility and contemporaneous Range as independent variables.

$$VolVol_t = \sum_{i=1}^{m} \phi_i VolVol_{t-i} + \sum_{j=1}^{n} \theta_j \epsilon_{j-i} + Range_t + \epsilon_t$$

Modified baseline model: Log number of messages added to the baseline model.

$$VolVol_t = \sum_{i=1}^{m} \phi_i VolVol_{t-i} + \sum_{j=1}^{n} \theta_j \epsilon_{j-i} + Range_t + \log(NMsg) + \epsilon_t$$

Four RTT latency variables added to the baseline model.

$$\begin{aligned} VolVol_t = &\sum_{i=1}^{m} \phi_i VolVol_{t-i} + \sum_{j=1}^{n} \theta_j \epsilon_{j-i} + Range_t + \\ &+ Latency_t^{(k)} + \epsilon_t \qquad\qquad k = 1, 2, 3, 4 \end{aligned}$$

Four RTT latency variables added to the modified baseline model.

$$\begin{aligned} VolVol_t = &\sum_{i=1}^{m} \phi_i VolVol_{t-i} + \sum_{j=1}^{n} \theta_j \epsilon_{j-i} + Range_t + \\ &+ \log(NMsg) + Latency_t^{(k)} + \epsilon_t \qquad\qquad k = 1, 2, 3, 4 \end{aligned}$$

where

$Latency_t^{(1)} = log[Median(RTTAllMessages_t)]$
$Latency_t^{(2)} = log[Median(RTTCancelMessages_t)]$
$Latency_t^{(3)} = dispersion(RTTAllMessages_t)$
$Latency_t^{(4)} = dispersion(RTTCancelMessages_t)$.

We run similar regression specifications for the Range–Volatility of Volatility pair, as well as the Volatility of Volatility–Absolute Returns and the Absolute Returns–Volatility of Volatility pairs.[5] We also perform this analysis for log Returns by using lagged log returns for the baseline model and adding the log number of messages for the modified baseline model.

C. Results

Our results for the statistical relation between latency and returns are presented in Table II.

<Insert Table II>

As evidenced by the t-statistics, none of the latency variables have significant statistical explanatory power over and above lagged returns in terms of improving the goodness–of–fit.

In contrast, for the Range–Volatility of Volatility specification of the baseline and modified models presented in Table III, latency variables come in highly statistically significant and improve the goodness–of–fit.

<Insert Table III>

Similarly, for the Volatility of Volatility–Range specification presented in Table IV, latency variables come in highly statistically significant and improve the goodness–of–fit.

<Insert Table IV>

This specification is particularly worth noticing since our semi–parametric measure of the volatility of volatility proxies for the latent unobserved variable of the same name while range serves as a statistical proxy for the (also latent) volatility. Thus, these results can be interpreted as follows: latency indicators improve estimates of the volatility of volatility over and above whatever could be extracted from the observed prices, as well as observed prices and the message traffic combined. Furthermore, latency dispersion makes a stronger statistical contribution than median latency.

[5] Absolute Returns serve as alternative–less efficient than Range–estimator of volatility. The results for Absolute Returns as an estimator of volatility are qualitatively very similar to the results for the Range. They were conducted for robustness and are available upon request.

IV. Concluding Remarks

We present measurements of different components of intraday trading platform latency for the most liquid index futures contract traded exclusively on the the BM&FBOVESPA. We find that trading platform latency is not a constant. It is a random variable best described by a mixture of a bell–shaped distribution and power law right tail. At lower levels, latency has a strong tendency to cluster around a central value, but after a certain cutoff point, trading platform latency could become extremely large.

We then show that intraday dynamics of median trading platform latency and the dispersion of latency add significant explanatory power to changes in volatility and the volatility of volatility. One way to interpret these empirical regularities is that latency is a risk factor that arises due to automated trading technology. Another is that intraday dynamics of both trading platform latency and asset prices are driven by changes in one or several endogenous latent factors that first manifest themselves in the characteristics of latency and then in asset prices.

Irrespective of what latency proxies for, if latency and its dispersion matter for the price–setting process matter for asset prices, they are should be reported by the trading platforms to the market participants on an ongoing basis. Trading platforms should include latency indicators into their market feed. That way, latency can be discovered directly along with asset prices. This would improve the pre–trade transparency of automated trading platforms.

References

Ait–Sahalia, Yacine, and Mehmet Saglam, 2013, High Frequency Traders: Taking Advantage of Speed, mimeo.

Bollerslev, Tim, Tauchen, George, and Hao Zhou, 2009, Expected Stock Returns and Variance Risk Premia, *Review of Financial Studies* 22, 11, 4463–4492.

Coval, Joshua D., and Shumway, Tyler, 2001, Is Sound Just Noise?, *Journal of Finance* 56, 5, 1887–1910.

Wang, Ruoyang, Kirby, Chris, and Steven Clark, 2013, Volatility of Volatility, Expected Stock Return and Variance Risk Premium, mimeo.

Table I: Descriptive Statistics of Latency

Type	Variable	Median	Mean	Std Dev	Skewness	Kurtosis	Minimum	Maximum	N
Submission	**RTT us**	1205	1799	11899	53	3245	797	999986	370342
	FWD us	405	700	11427	57	3694	201	998811	370342
	BWD us	315	374	3099	144	23479	188	689425	370342
	ENG us	459	725	1095	12	330	305	67654	370342
Modification	**RTT us**	1488	1784	1597	98	17315	814	306634	248203
	FWD us	479	542	1190	231	56377	219	305849	248203
	BWD us	415	484	269	10	301	195	15583	248203
	ENG us	542	758	874	13	400	315	56381	248203
Cancellation	**RTT us**	1086	1788	18575	43	1956	790	1001260	199814
	FWD us	286	788	18545	44	1968	205	1000453	199814
	BWD us	342	375	284	29	1206	195	15669	199814
	ENG us	444	625	879	12	265	303	42964	199814
All Types	**RTT us**	1239	1792	12210	60	3886	790	1001260	818359
	FWD us	395	674	11979	62	4149	201	1000453	818359
	BWD us	346	408	2095	211	50834	188	689425	818359
	ENG us	466	711	983	12	348	303	67654	818359

Latency descriptive statistics are shown for the most liquid future traded in Brazil, the Ibovespa index Mini-Future, for maturity “April 2014” (symbol WINJ14) on March 13, 2014.

We compute statistics per message types (submission, modification, cancellation) and all types aggregated. We analyze Round-Trip (RTT) latency and its components: the Forward latency (FWD), the Matching Engine latency (ENG) and the Outgoing latency (BWD).

Table II: Regressions - Log Returns

Y = Model	**Criteria**	**Variable X**	**t-stat X**	**AIC**	**BIC**	**LBox**	**Adj R2**	**log-likd**
Ret = ARMAX(3,5,X)	Opt AIC	-	-	-64,321	-64,248	0	52.1%	32,170
		Median (All)	(2.88)	-64,327	-64,247	0	52.1%	32,175
		Median (Cancel)	(0.46)	-64,303	-64,223	0	52.0%	32,163
		Disp (All)	(5.13)	-64,327	-64,247	0	52.1%	32,175
		Disp (Cancel)	(4.28)	-64,322	-64,242	0	52.1%	32,172
Ret = ARMAX(2,3,X)	Opt BIC	-	-	-64,318	-64,267	0	52.0%	32,166
		Median (All)	0.55	-64,317	-64,259	0	52.0%	32,166
		Median (Cancel)	0.89	-64,318	-64,260	0	52.0%	32,167
		Disp (All)	(1.33)	-64,318	-64,259	0	52.0%	32,167
		Disp (Cancel)	(1.78)	-64,318	-64,260	0	52.0%	32,167
Ret = ARMAX(3,6,NMsg,X)	Opt AIC	-	-	-64,331	-64,244	0	52.1%	32,177
		Median (All)	(2.07)	-64,326	-64,231	0	52.1%	32,176
		Median (Cancel)	1.86	-64,330	-64,236	0	52.1%	32,178
		Disp (All)	2.67	-64,331	-64,237	0	52.1%	32,179
		Disp (Cancel)	1.50	-64,326	-64,231	0	52.1%	32,176
Ret = ARMAX(1,2,NMsg,X)	Opt BIC	-	-	-64,323	-64,279	0	52.1%	32,167
		Median (All)	0.11	-64,273	-64,222	1	51.8%	32,144
		Median (Cancel)	0.26	-64,321	-64,270	0	52.1%	32,167
		Disp (All)	2.61	-64,324	-64,273	0	52.1%	32,169
		Disp (Cancel)	(0.02)	-64,321	-64,270	0	52.0%	32,167

Respectively, (a) the best ARMAX models of log-returns and (b) the best ARMAX models of log-returns with log-number of messages are tested without and with the addition of different latency variables. The best ARMAX models are defined as the two models that minimize AIC and the BIC criteria. The latency variables are: the median latency per interval of all messages (Median(All)) and of cancel messages only (Median(Cancel)); and the latency dispersion per interval of all messages (Disp(All)) and of cancel messages only (Disp(Cancel)). The columns show the t-statistic of the added exogenous variable, the AIC and BIC criteria, the Ljung-Box test (with rejection of the null of uncorrelated residuals with 5% confidence level), the adjusted R2, and the log-likelihood. Latency variables have neither shown substantial significance nor improved regressions of log-returns, especially after controlling for (log) number of messages.

Table III: Regressions - Volatility (Range)

Y = Model	**Criteria**	**Variable X**	**t-stat X**	**AIC**	**BIC**	**LBox**	**Adj R2**	**log-likd**
Range = ARMAX(1,4,VolVol,X)	Opt AIC	-	-	-36,731	-36,673	0	53.2%	18,374
		Median (All)	14.60	-36,786	-36,720	0	53.5%	18,402
		Median (Cancel)	14.44	-36,786	-36,720	0	53.5%	18,402
		Disp (All)	30.97	-36,951	-36,886	0	54.2%	18,485
		Disp (Cancel)	34.03	-36,949	-36,883	0	54.2%	18,483
Range = ARMAX(3,1,VolVol,X)	Opt BIC	-	-	-36,731	-36,680	0	53.2%	18,372
		Median (All)	14.46	-36,785	-36,727	0	53.5%	18,401
		Median (Cancel)	14.62	-36,787	-36,729	0	53.5%	18,401
		Disp (All)	30.70	-36,949	-36,891	0	54.2%	18,482
		Disp (Cancel)	33.99	-36,948	-36,890	0	54.2%	18,482
Range = ARMAX(4,5,VolVol,NMsg,X)	Opt AIC	-	-	-36,992	-36,898	0	54.4%	18,509
		Median (All)	7.65	-37,008	-36,906	0	54.4%	18,518
		Median (Cancel)	16.01	-37,065	-36,963	0	54.7%	18,547
		Disp (All)	16.51	-37,065	-36,963	0	54.7%	18,546
		Disp (Cancel)	21.59	-37,108	-37,006	0	54.9%	18,568
Range = ARMAX(0,4,VolVol,NMsg,X)	Opt BIC	-	-	-36,984	-36,926	0	54.3%	18,500
		Median (All)	7.90	-37,000	-36,934	0	54.4%	18,509
		Median (Cancel)	16.55	-37,060	-36,995	0	54.6%	18,539
		Disp (All)	16.39	-37,055	-36,989	0	54.6%	18,536
		Disp (Cancel)	21.63	-37,098	-37,033	0	54.8%	18,558

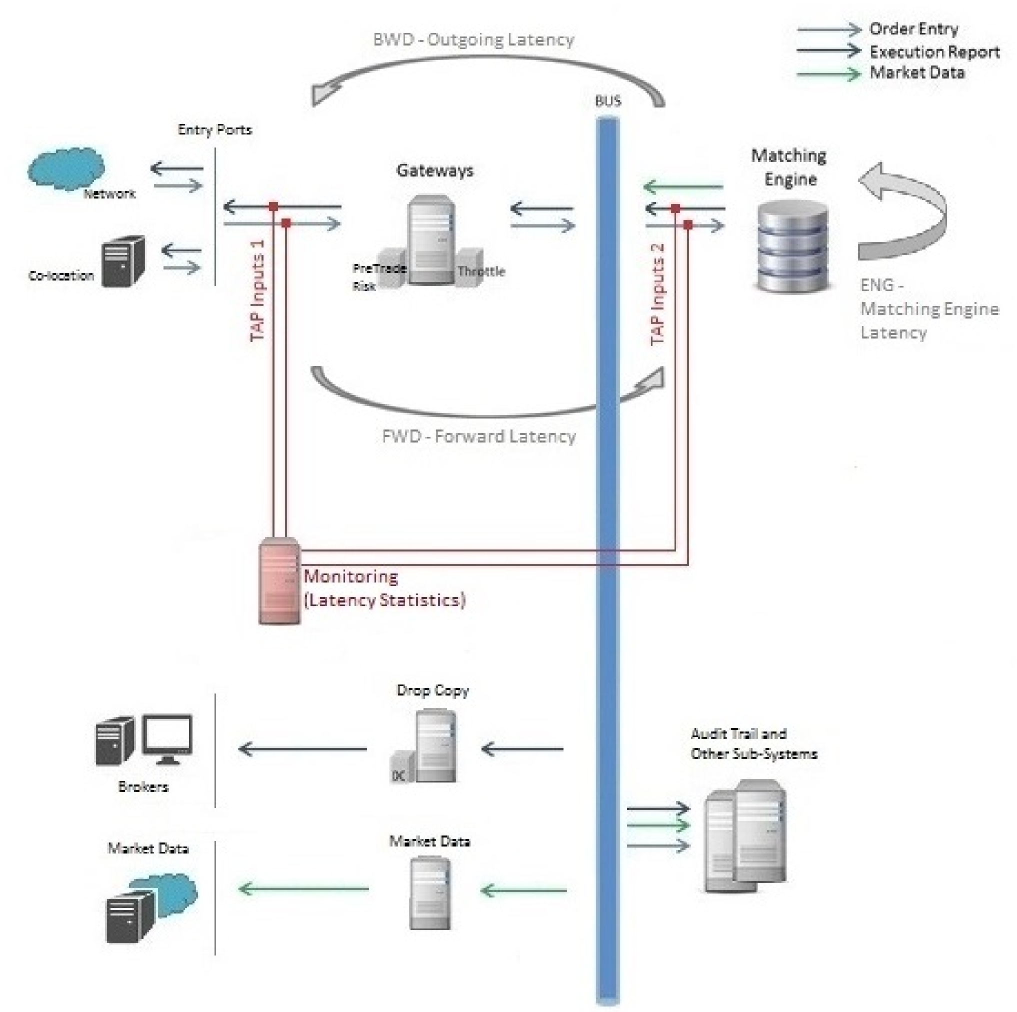
Respectively, (a) the best ARMAX models of range with volatility-of-volatility and (b) the best ARMAX models of range with volatility-of-volatility and log-number of messages are tested without and with the addition of different latency variables. The best ARMAX models are defined as the two models that minimize AIC and the BIC criteria. The latency variables are: the median latency per interval of all messages (Median(All)) and of cancel messages only (Median(Cancel)); and the latency dispersion per interval of all messages (Disp(All)) and of cancel messages only (Disp(Cancel)). The columns show the t-statistic of the added exogenous variable, the AIC and BIC criteria, the Ljung-Box test (with rejection of the null of uncorrelated residuals with 5% confidence level), the adjusted R2, and the log-likelihood. Latency variables have both shown substantial significance and improved regressions of ranges, including those controlled by (log) number of messages.

Table IV: Regressions - Volatility of Volatility

Y = Model	**Criteria**	**Variable X**	**t-stat X**	**AIC**	**BIC**	**LBox**	**Adj R2**	**log-likd**
VolVol = ARMAX(10,3,Range,X)	Opt AIC	-	-	-101,327	-101,211	0	53.4%	50,680
		Median (All)	75.34	-103,075	-102,951	0	60.4%	51,554
		Median (Cancel)	32.80	-101,580	-101,457	1	54.5%	50,807
		Disp (All)	218.10	-103,635	-103,511	0	62.5%	51,834
		Disp (Cancel)	157.19	-102,676	-102,552	0	58.9%	51,355
VolVol = ARMAX(5,1,Range,X)	Opt BIC	-	-	-101,304	-101,238	0	53.2%	50,661
		Median (All)	77.92	-103,065	-102,992	0	60.4%	51,543
		Median (Cancel)	33.47	-101,630	-101,558	0	54.6%	50,825
		Disp (All)	223.92	-103,625	-103,553	0	62.4%	51,823
		Disp (Cancel)	164.84	-102,721	-102,648	0	59.1%	51,370
VolVol = ARMAX(1,2,Range,NMsg,X)	Opt AIC	-	-	-104,931	-104,881	0	66.8%	52,473
		Median (All)	39.48	-105,553	-105,495	0	68.7%	52,785
		Median (Cancel)	33.95	-105,291	-105,233	0	67.9%	52,654
		Disp (All)	64.30	-105,328	-105,270	0	68.0%	52,672
		Disp (Cancel)	55.81	-105,286	-105,228	0	67.9%	52,651
VolVol = ARMAX(1,2,Range,NMsg,X)	Opt BIC	-	-	-104,931	-104,881	0	66.8%	52,473
		Median (All)	39.48	-105,553	-105,495	0	68.7%	52,785
		Median (Cancel)	33.95	-105,291	-105,233	0	67.9%	52,654
		Disp (All)	64.30	-105,328	-105,270	0	68.0%	52,672
		Disp (Cancel)	55.81	-105,286	-105,228	0	67.9%	52,651

Respectively, (a) the best ARMAX models of volatility-of-volatility with range and (b) the best ARMAX models of volatility-of-volatility with range and log-number of messages are tested without and with the addition of different latency variables. The best ARMAX models are defined as the two models that minimize AIC and the BIC criteria. The latency variables are: the median latency per interval of all messages (Median(All)) and of cancel messages only (Median(Cancel)); and the latency dispersion per interval of all messages (Disp(All)) and of cancel messages only (Disp(Cancel)). The columns show the t-statistic of the added exogenous variable, the AIC and BIC criteria, the Ljung-Box test (with rejection of the null of uncorrelated residuals with 5% confidence level), the adjusted R2, and the log-likelihood. Latency variables have both shown substantial significance and improved regressions of volatility-of-volatility, including those controlled by (log) number of messages.

Figure 1: An Automated Trading Platform



Messages arrive at the Entry Ports, sent from either co-located servers or external networks. They enter the Gateways and are submitted to pre-trade risk checks. Messages arriving from different Gateways towards a certain market will be directed to the same Matching Engine, via the (multicast) Bus. The Matching Engine takes a while to process the message and sends a confirmation back to the Gateways, which needs further processing, too. The Monitoring System is in red. The time it takes for a message to travel from TAP Inputs 1 until TAP Inputs 2 measures the Forward Latency (FWD), the bulk of which is the Gateway processing time. The Matching Engine Latency (ENG) is measured by the delay between when a message passes through TAP Inputs 2 and its confirmation is observed back at the same point. The Outgoing Latency (BWD) is measured by the time it takes for the confirmation to travel back from TAP Inputs 2 until TAP Inputs 1, the bulk of which is, again, comprised by Gateway processing overhead. The Bus dispatches message information to other systems as well, like the drop-copy, market data and audit trail.

Figure 2: RTT All Types

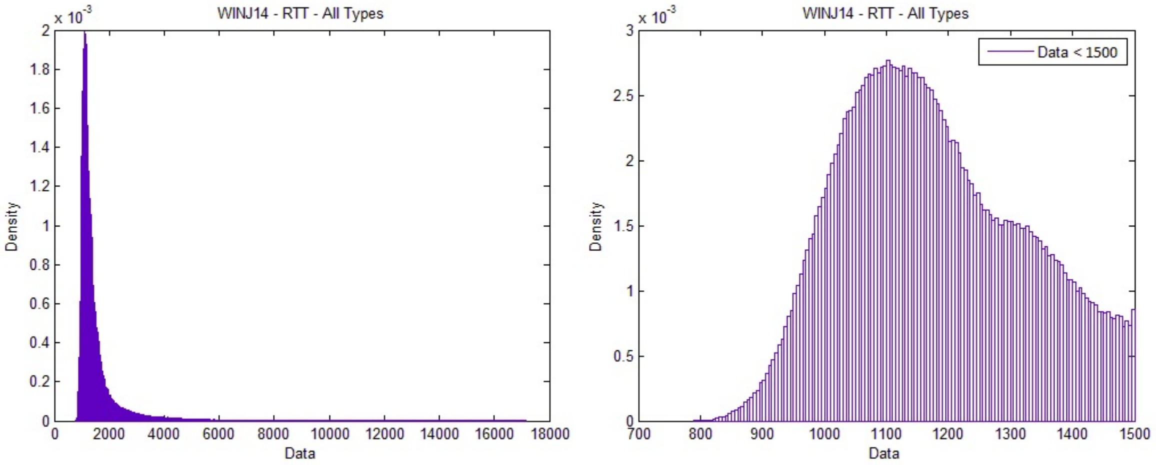


Figure 3: RTT All Types with Log-Normal/Power-Law Fit

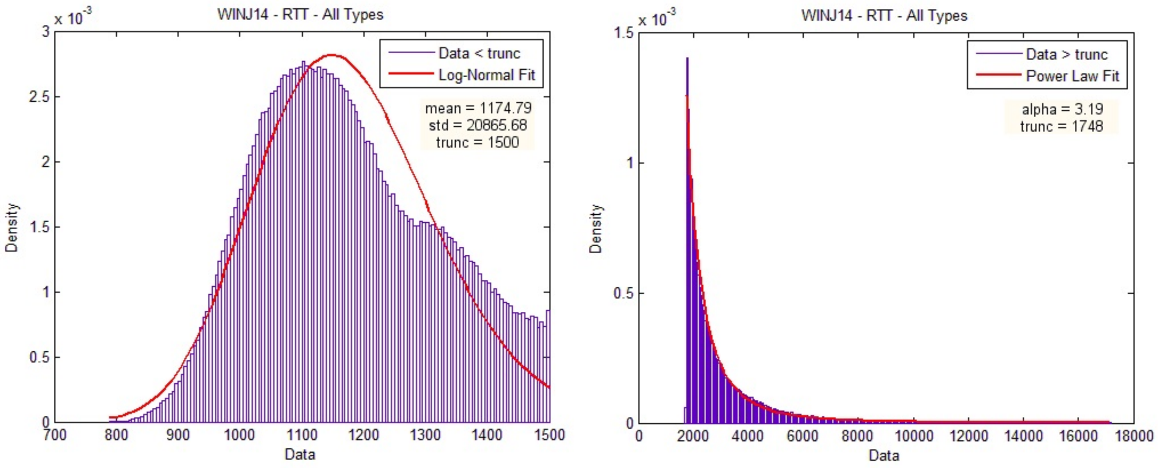


Figure 4: RTT for Submit, Modify and Cancel Messages

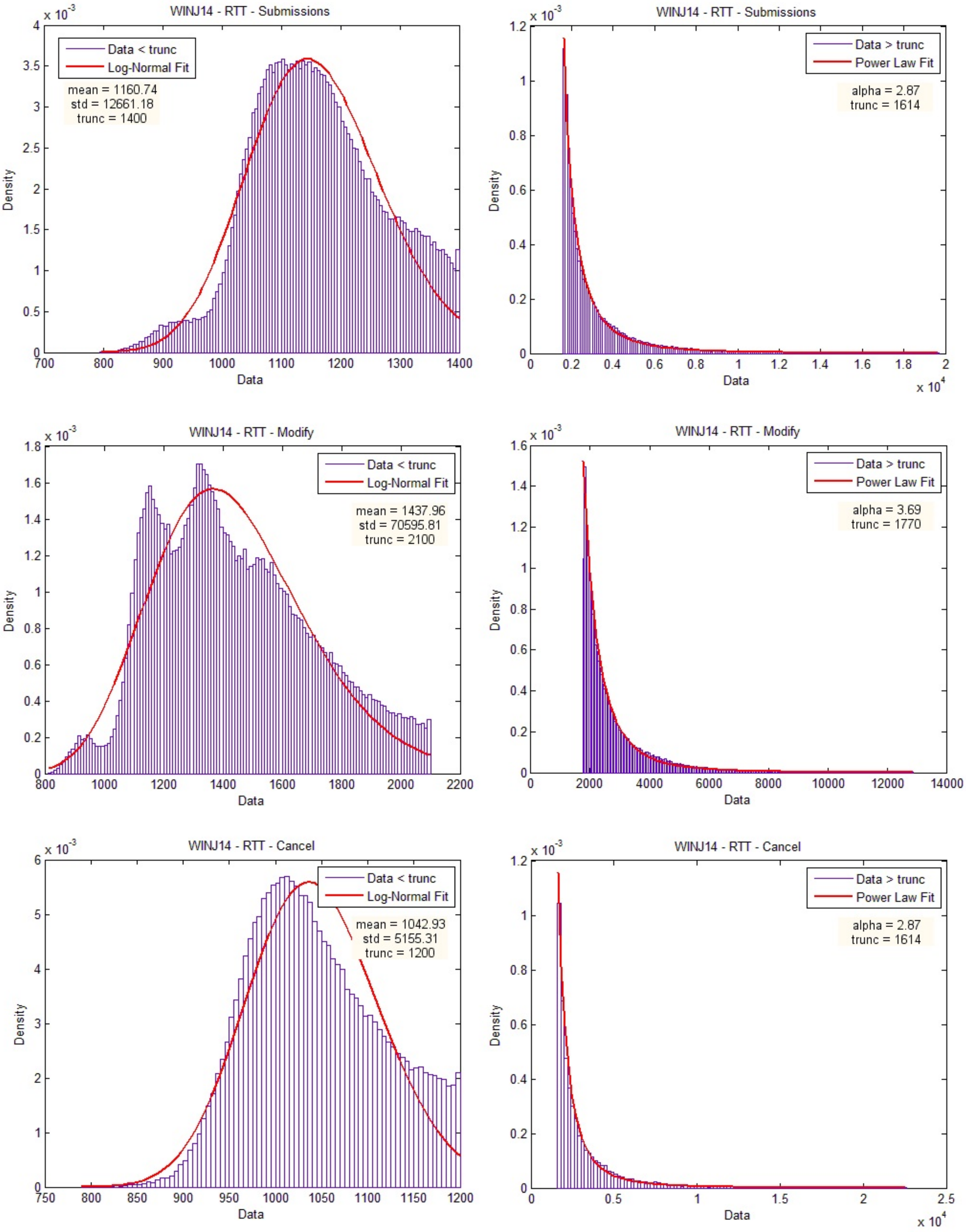


Figure 5: FWD for Submit, Modify and Cancel Messages

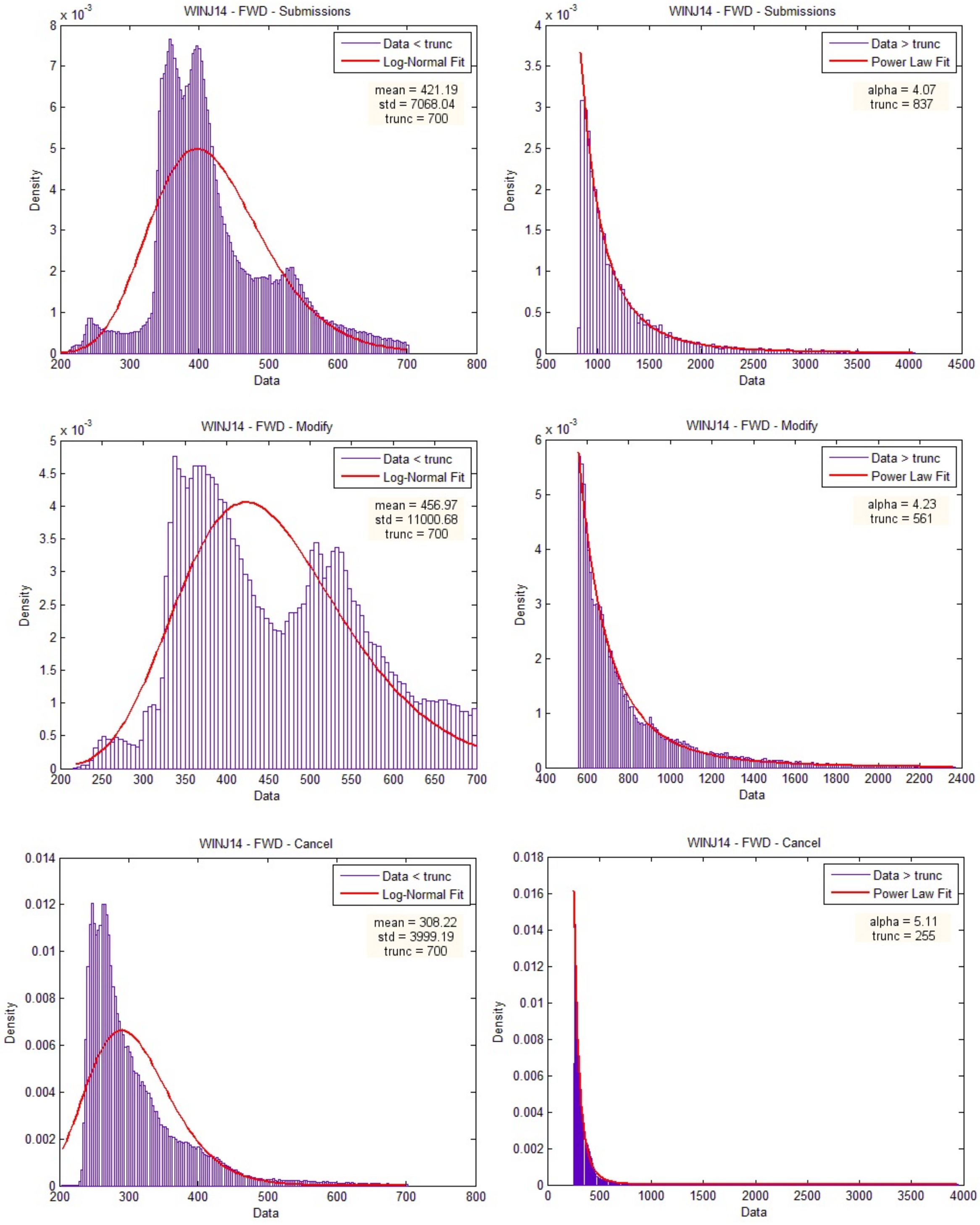


Figure 6: ENG for Submit, Modify and Cancel Messages

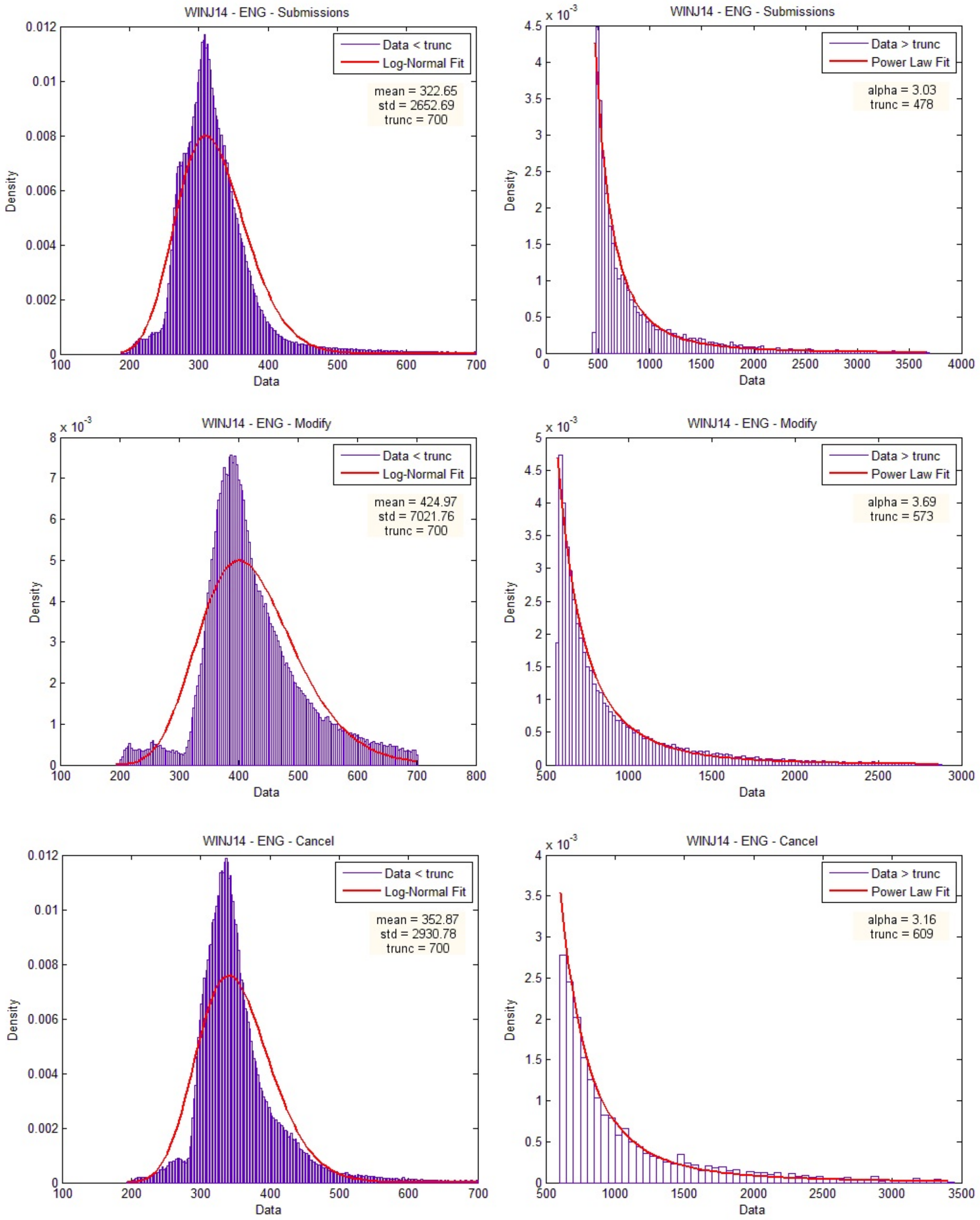


Figure 7: BWD for Submit, Modify and Cancel Messages

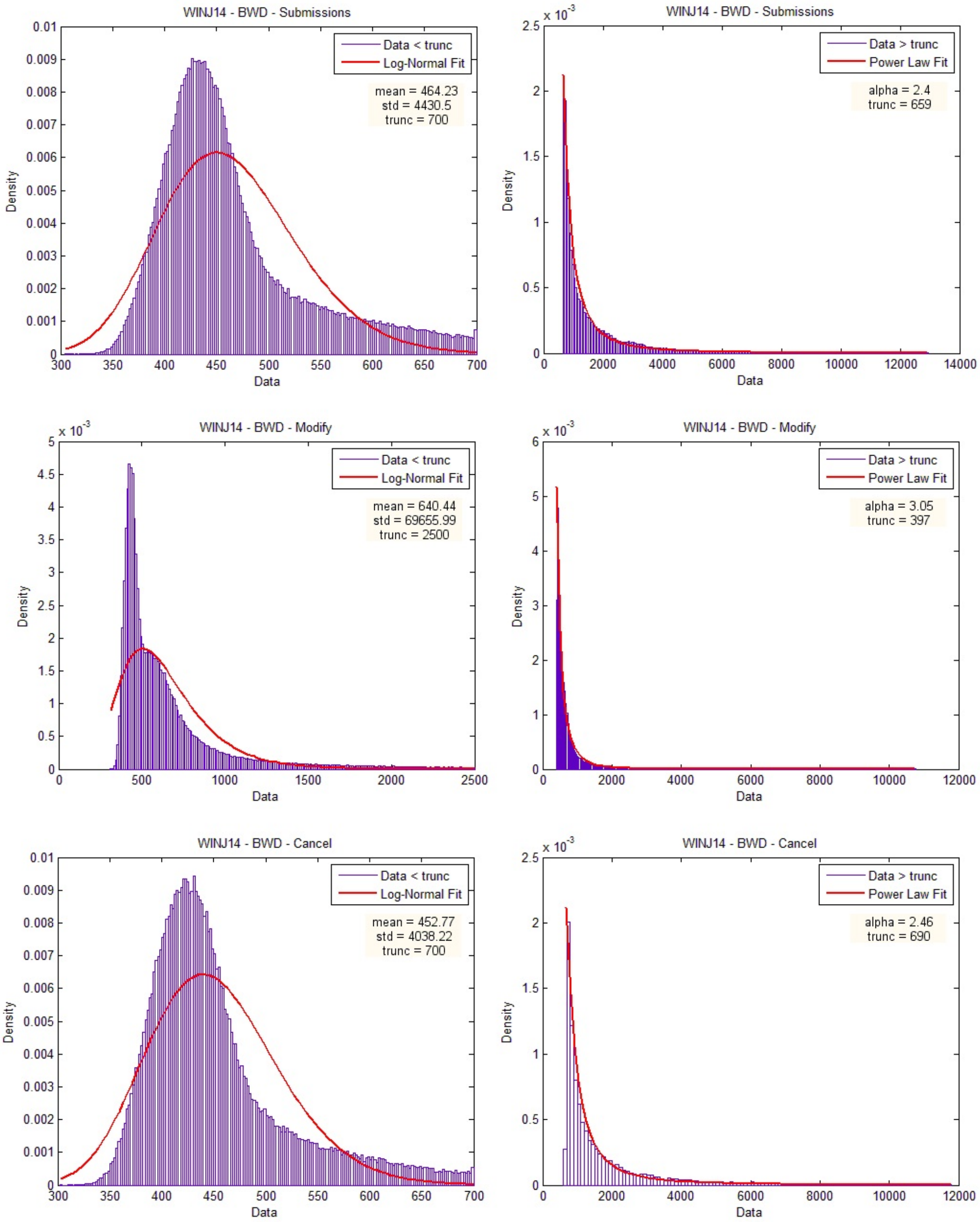


Figure 8: RTT Time Series - All Messages and Cancellations

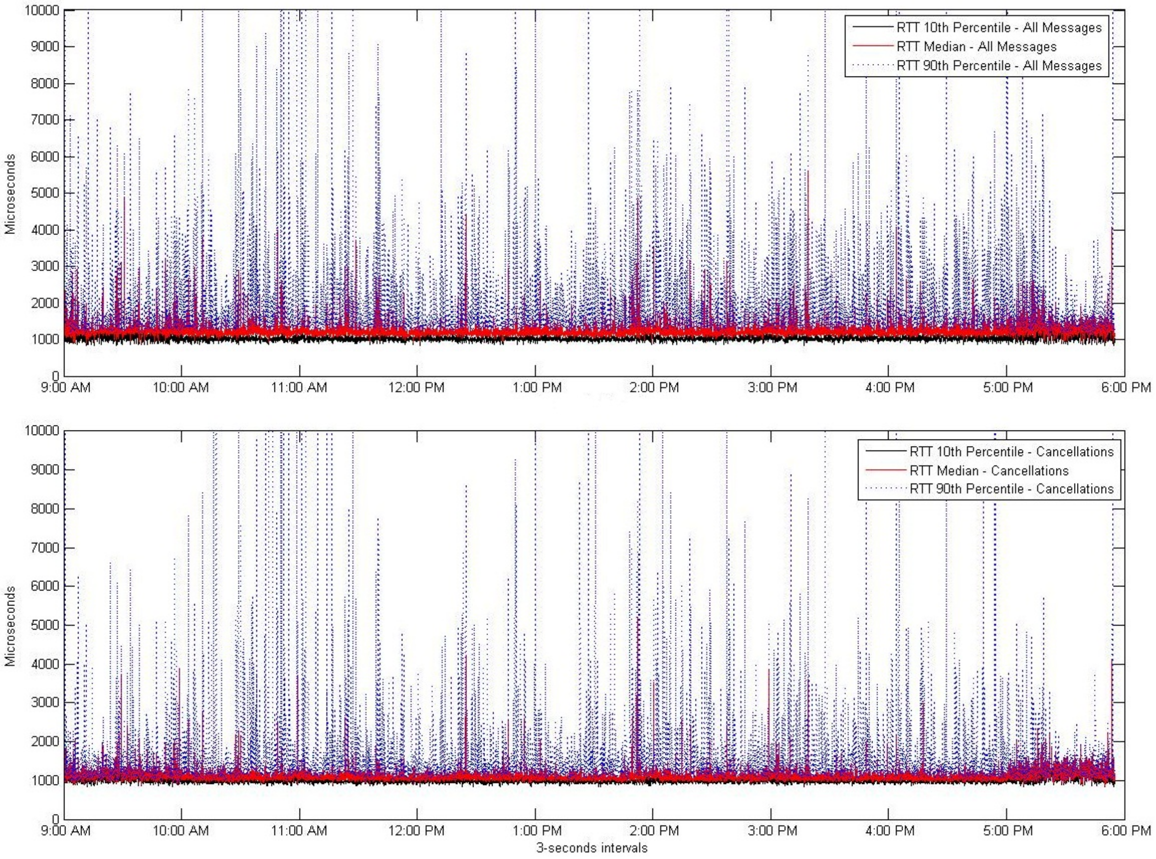


Figure 9: FWD/ENG/BWD Time Series - All Messages

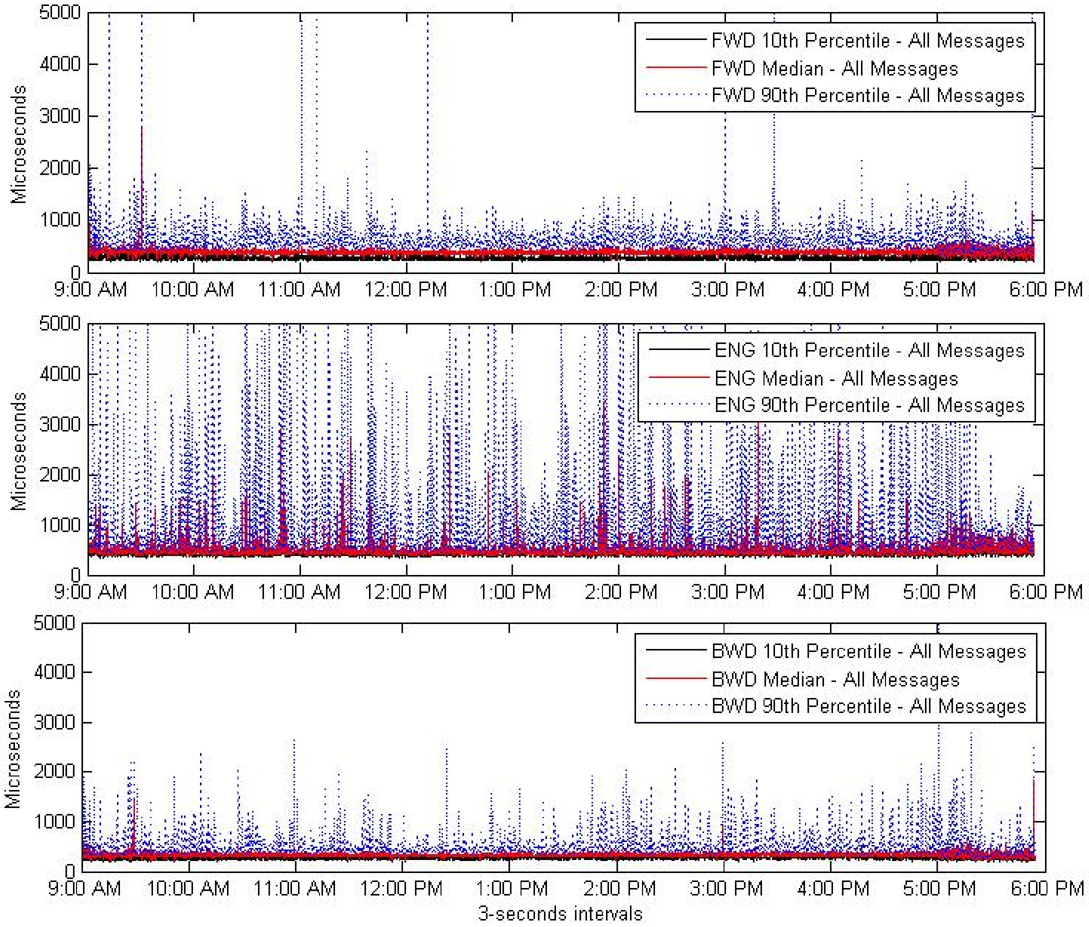


Figure 10: Number of Messages Time Series

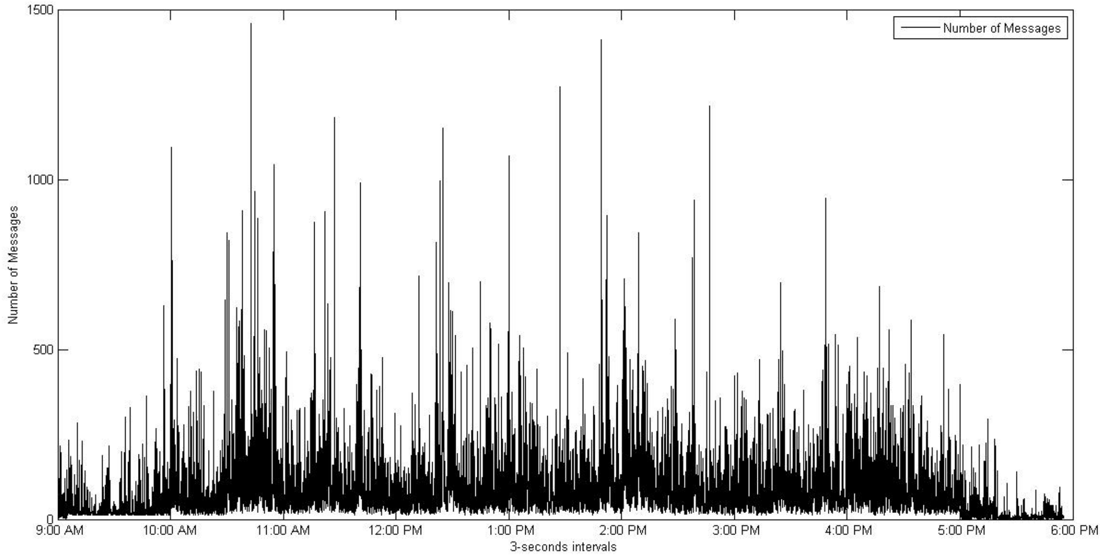


Figure 11: Log-Returns Time Series

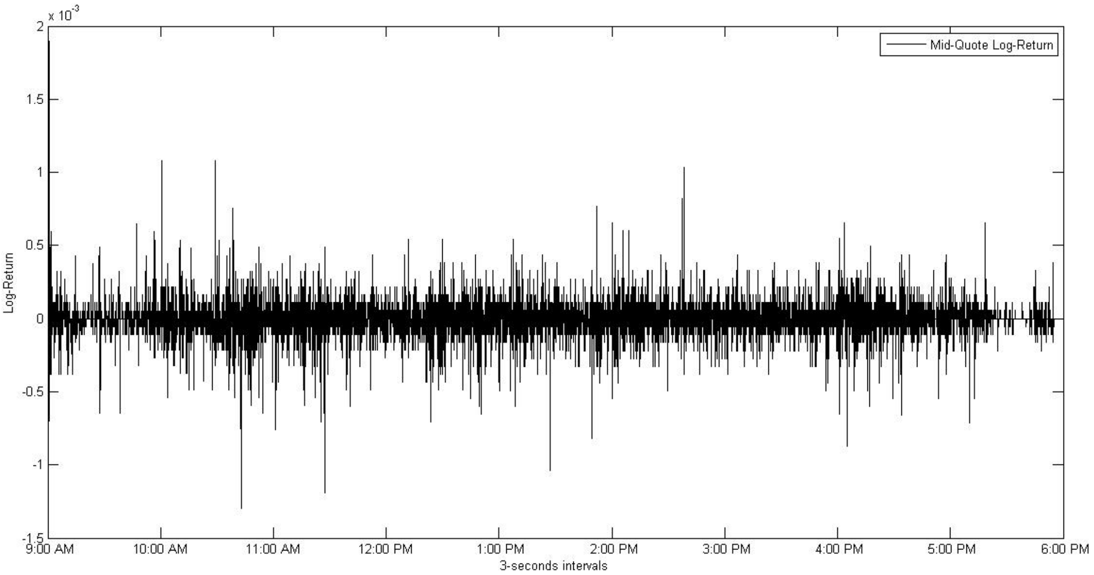


Figure 12: Range Time Series

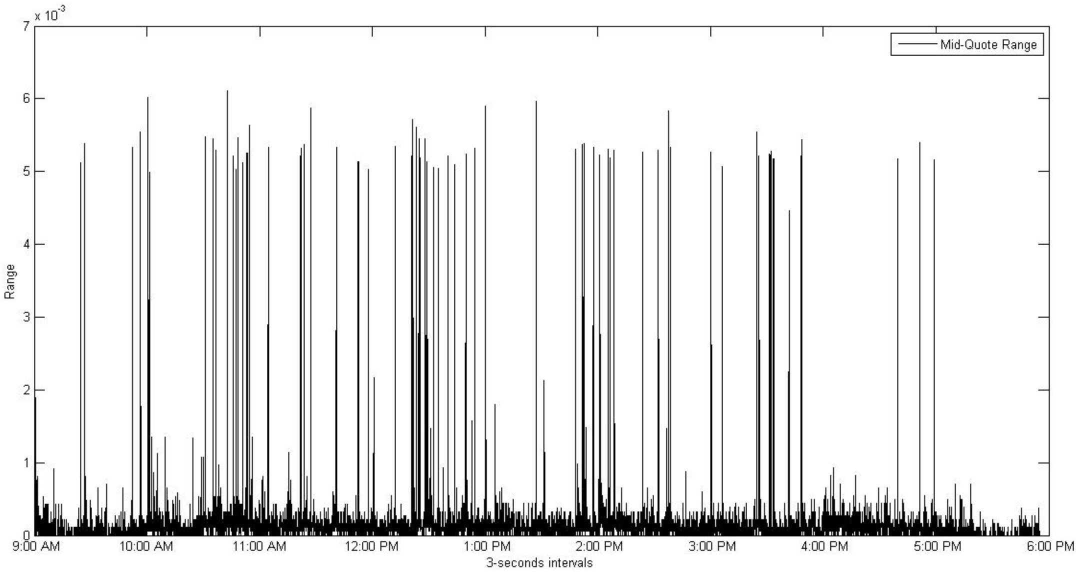
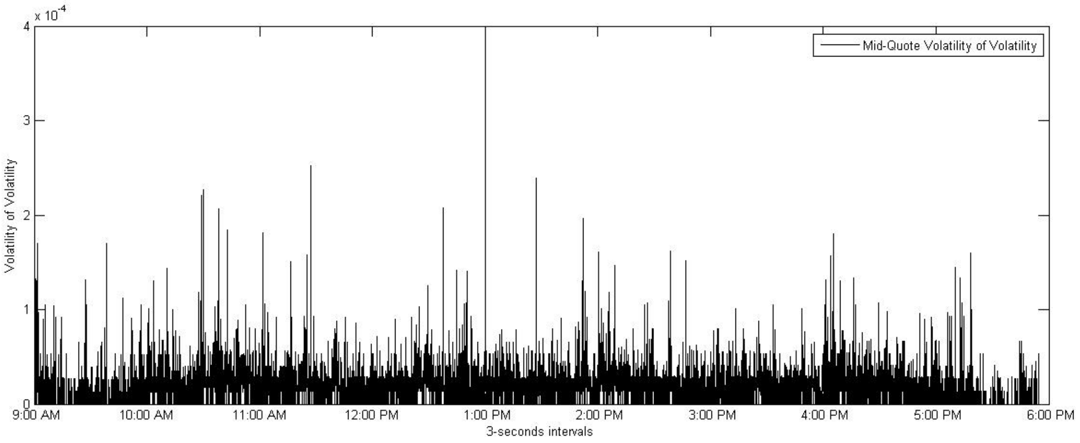


Figure 13: Volatility of Volatility Time Series


x 10^-4
Volatility of Volatility
Mid-Quote Volatility of Volatility
9:00 AM
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
4:00 PM
5:00 PM
6:00 PM
3-seconds intervals