Low Latency Trading in FIX Environments
Introduction

Financial Information eXchange (FIX) is a session-layer messaging standard designed for and by the financial securities industry. FIX was primarily conceived as a means to standardize securities transactions and the exchange of real-time market information and status. Today, FIX is viewed as the de facto international language for the automated trading of financial instruments, used extensively by buy- and sell-side firms, trading platforms, and regulators. FIX is a fundamental component of many of the electronic trading trends that have emerged over the past decade, improving efficiencies and enabling new forms of highly automated trading. With trillions of dollars traded annually on exchanges, financial institutions continue to invest heavily in optimizing electronic trading systems, employing direct market access (DMA) to increase their speed to market.

Managing the performance and service delivery of trading applications increasingly requires an intimate understanding of FIX, together with solutions that properly support this highly demanding environment. FIX trading requires a high-performance, low latency, fully secured network infrastructure, with zero downtime. FIX trading also requires fine-grained inspection and control of transactions – to enable the best client experience and provide competitive advantage.

A10 Solutions for Securities Trading

In this paper, we review the history of FIX, related standards and key securities applications, and we show how A10’s Advanced Core Operating System (ACOS®) is ideally suited to cope with the low latency/low jitter demands of securities trading. We discuss how A10 Networks high-performance Application Services Gateway (ASG) products are ideally suited for optimizing, accelerating and securing FIX messaging, as well as higher level Web traffic, across a range of applications that help scale server pools and improve user experience. We discuss how A10 solutions deliver fast, secure and highly resilient services; features that improve the customer experience and ultimately increase customer loyalty.

For readers with a nonfinancial background, a small glossary is provided together with useful references to resources at the end of this document.

FIX Overview

The FIX Protocol specification was co-authored in 1992 by Robert Lamoureux and Chris Morstatt, specifically to support electronic communication of equity trading data between Fidelity Investments and Salomon Brothers, for pre-trade and trade activities. Since its initial release, the scope of FIX has expanded significantly, through to the post-trade space, supporting straight-through processing (STP) from indications of interest (IOI) to allocations and confirmations. FIX is now the de facto messaging standard for pre-trade and trade messaging in global equity markets, and has been adopted more broadly in post-trade activities, foreign exchange (FX), fixed income and derivatives.

Who Uses FIX and Why?

There are a number of key reasons why the securities community continues to use, develop and invest in FIX infrastructure, including:

- FIX is a free and open standard.
- FIX is not generic. It was developed by the securities community for securities applications, and is extensively used, tested and peer reviewed to ensure it remains fit for purpose.
- The FIX protocol is relatively simple and has a fairly narrow focus; key benefits being very little overhead and improved protocol efficiency and performance over a generic messaging scheme.
- FIX compresses the time needed for price discovery and time to transact, by enabling the electronic exchange of trade-related information (it does not require human intervention).
- Since FIX enables the electronic exchange of trade-related information, it is a key component in automation for high-frequency trading, algorithmic trading and arbitrage (we discuss these shortly).

Today, FIX is the language for the global trading community, used by hundreds of FIX-enabled firms, together with virtually all of the primary dealers and alternate trading systems and many of the largest fund complexes and money management firms.
Applications Supported by FIX

FIX (and its XML variant FIXML) is applicable to functions in a number of securities business units within investment banking, including:

- **Equities**
- **Futures and Options**
- **Foreign Exchange**
- **Exchanges and Markets**
- **Fixed Income**

Note that full support for all features required by these applications depends on the specific version of FIX deployed, although the majority of the required functionality has been implemented since release 4.4. For further information on specific version mappings and capabilities, refer to the “Functionality Matrix” page at the FIX Trading Community portal [4].

**FIX Standardization**

The FIX messaging standard is developed and maintained through the collaborative efforts of the member firms of the FIX Trading Community™, which include many of the world’s leading financial institutions. These firms work together to ensure that FIX continues to meet emerging trading requirements, and to promote increased adoption. FIX Protocol Ltd (FPL) is a company which has been set up to own and maintain FIX specifications [1], and to ensure that FIX remains in the public domain. Many firms now post details of their FIX communications support capability on the FIX Trading Community website in order to promote interoperability.

**Release History**

Since its inception, FIX has been continuously developed through the collaboration of banks, broker-dealers, exchanges/execution venues, industry utilities/associations, institutional investors and information technology providers. FIX now comprises a number of related standards; the main FIX Protocol Application Layer messaging specification is currently at Version 5.0, issued in October 2006. The previous major release (version 4.0) was issued in January 1996, with four major patch revisions in-between.

As FIX has been continually improved, it has addressed broader requirements across a range of securities applications; for example, version 4.4 (issued April 2003) included much broader support for core banking operations, as described later in this paper. When a new FIX version is issued, organizations have a responsibility to rigorously test and characterize the new version in their own environment.

**Transport Methods**

FIX is a session-layer protocol (ISO OSI Model Layer 5), and was initially designed to run over TCP. Since FIX operates as a high-level messaging protocol, it will, in theory, work across existing legacy networks and newer transport methods (although efficiency can vary). The latest versions of FIX implement Transport Independence; this allows multiple versions of application messages to be carried over a single version of Transport Independent FIX Session. This enables alternate transport methods such as Message Queues (or even Web Services) to carry FIX messages.

**Additional Capabilities**

FIX also now supports Algorithmic Trading by use of FIX Algorithmic Trading Definition Language (FIXatdl). FIX Protocol Limited has also released the FIX Adapted for Streaming (FAST) protocol, used for sending multicast market data.

**Related Standards**

There are a number of related standards and vendor initiatives for securities applications. For example, W3C, ISO (15022) and others have worked on related standards initiatives promoting interoperability for FIXML, SWIFTML and FpML.
**FIX Message Format**

FIX is essentially a *self-describing protocol*, not dissimilar to more recent standards such as XML. FIX messages are represented in text form (using a structured sequence of delimited numeric fields called “TAGS,” followed by associated values), or by using XML (FIXML). To date, FIXML is not as widely deployed in low latency applications, primarily because of the additional complexity and messaging overhead (XML has much less compact tag encoding, and this can impact performance).

FIX message fields are delimited using the ASCII 01 <start of header> character. They comprise three main blocks: a *header*, a *body* and a *trailer*. Within a FIX message, there will be a number of mandatory and optional fields, with each field comprising a numeric “TAG” followed by an array of bytes indicating the associated value, and with each tag value pair separated by an “SOH” character (0x01). There are two main categories of message, *admin* and *application* (with *admin* responsible for session maintenance, and *application* dealing with trade-related information such as an order request or the current state of an order). For further information, refer to the specifications held at [1].

**An Example**

An example of a FIX message (version 4.2, body length 178 bytes) is given below. This is an Execution Report, and note that the pipe character “|” is being used to represent an SOH character (normally not visible).

```
8=FIX.4.2 | 9=178 | 35=8 | 49=PHLX | 56=PERS | 52=20071123-05:30:00.000 | 11=ATOMNOCCC9990900 | 20=3 | 150=E |
39=E | 55=MSFT | 167=CS | 54=1 | 38=15 | 40=2 | 44=15 | 58=PHLX EQUITY TESTING | 59=0 | 47=C | 32=0 | 31=0 | 151=15 |
14=0 | 6=0 | 10=128 |
```

In the above example, the FIX Message Body length is indicated by TAG 9, and the checksum by TAG 10. The body contains trade details and the trailer contains a Security Check through Signature and the CheckSum (TAG 10). Note that SenderCompID (TAG 49) is a key field used to identify a client and the FIX session, as we discuss later.

**FIX Application Requirements**

Applications that use FIX have a range of requirements, primarily centered on speed, fairness, service quality, reliability and scalability. Underlying these needs is the requirement for *application fluency*; that is, the ability for a device to understand the messaging internals. As we discuss here, A10’s solution for FIX has the performance characteristics required, together with the ability to inspect and apply real-time policy based on FIX message structure and content using A10’s powerful aFleX® scripting language.

**High-Frequency Trading, Algorithmic Trading, Arbitrage**

There are a number of trading activities where transaction time is critical, including *arbitrage*, *high-frequency* and *algorithmic trading*.

Arbitrage essentially means buying in one market and selling in another, profiting on a temporary differential price between markets. In foreign exchange (forex), for example, currency exchange rates on different exchanges can vary, and these price differences can be both extremely small and short-lived. In order to capitalize on these differences, a trader may need to execute a very large volume of orders before the pricing anomaly is closed. Given this volume of transactions, and the short time window before the market normalizes, execution time is clearly a critical factor.

Large investment banks, hedge funds and institutional investors now use high-performance computing and state-of-the-art algorithms to implement trading strategies and transact a large number of orders at extremely high speeds. *Algorithmic trading* is used to identify emerging trends, sometimes within a fraction of a second, and then employ buy and sell strategies for optimizing returns. These high-frequency trading systems typically send hundreds of baskets of stocks out into the marketplace, at bid-ask spreads that are advantageous to the traders. By anticipating and outperforming market trends, institutions that implement high-frequency trading may gain a significant competitive advantage in the open market, and can realize favorable returns on trades through their bid-ask spread, resulting in significant profits. Well-known examples of trading strategies include: *Stealth* (developed by Deutsche Bank), *Sniper* and *Guerilla* (developed by Credit Suisse), *arbitrage*, *statistical arbitrage*, *trend following* and *mean reversion*.

---

1 The SenderCompID, OnBehalfOfCompID, TargetCompID and DeliverToCompID fields are used for routing between a FIX message initiator and receiver.

2 FIX users may identify a FIX Session using just the SenderCompID – TargetCompID pair, or they may choose to include SubId and LocID on some messages (this is largely a matter of counter-party agreement).
High-frequency trading became commonplace following the introduction of incentives offered by exchanges for institutions (i.e., market makers) to add liquidity to the markets. With high-frequency trading, exchanges gain increased liquidity, while institutions see improved profits on every trade through more favorable spreads. Note: Although we are referring to fractions of a cent per transaction, this can result in sizable profits for high-frequency traders over a large number of trades, so there is a clear incentive to trading at high volume and speed.

**Fairness**
A more subtle concern in these environments is the issue of "fairness." In a global market, institutions rightly want to capitalize on their R&D investment in algorithmic trading, and their ability to execute at high frequency and high volume based on advanced analytics. This might be viewed as unfair to smaller traders; however, this is the nature of an open market. Conversely, clients of a trading platform do wish to see relative fairness in the service they are offered on that platform. If two clients have contracted for the same service level on a trading system, they do not expect to see the other party gain any competitive advantage in terms of execution and reporting time. This is where the bounds and characteristics of latency really matter, which we discuss shortly.

**Summary**
These new forms of electronic trading are driving the demand for low latency proximity hosting and global exchange connectivity. It is therefore imperative that solutions for this environment have implicit understanding of FIX transactions in order to provide optimal low latency support for electronic trading strategies.

**Understanding Latency and Jitter**
As discussed earlier, in securities it is important to achieve low and consistent transaction times to support key applications and to provide fairness across clients. In this section, we briefly define more fully what we mean by latency and jitter.

![Figure 2: An illustration of the relationship between mean latency and jitter with a round-trip transaction scenario.](image)

*Average latency is the sum of all delays averaged over the number of messages (approximately 125 microseconds here).*

*Jitter is the variability around the mean latency – often presented as the Standard Deviation (SD) around the mean.*

Worst-case latency in this example is around 240 microseconds and the best-case latency is around 75 microseconds, a spread of approximately 165 microseconds. For some applications, such as video streaming, a wide spread in variability is hugely detrimental, since it represents inconsistency in the way content is delivered (i.e., a non-smooth or jittery flow).
Latency refers to the delay in the movement of information between two specified points. Typically this will be from source to destination, either one-way or as a measurement of round-trip delay (and therefore including delays introduced along communications paths). Context here is clearly important; in this paper, we are referring to the delays introduced through an A10 device, one-way for a typical FIX transaction.

Jitter refers to the variability in delay. This is an equally important but a more subtle characteristic of transaction handling. Jitter refers to the distribution of delay; that is, the predictability and bounds of delay when compared over multiple transactions. While our mean delay may be acceptable, we may still experience variable delays (i.e., jitter) that are completely unacceptable for certain applications.

Given that trading systems service multiple clients and normally compete on the open market, users of such a service do not expect significant time variability 1) between transactions; and 2) in the performance observed by other clients, since this could be perceived to offer unfair advantage (both externally and within a trading system). Ideally, we want to see low latency, together with tightly bounded jitter (i.e., the worst- and best-case delays should be fairly tightly constrained around the average).

Just How Low Can Latency Be?
In securities trading, more so arguably than any other activity, “time means money.” The faster and more reliably a transaction can be performed, the higher the probability of more significant gains. From a trader’s perspective, the ideal time for a completed financial transaction is zero seconds. While we are continually improving latency, zero latency is clearly not achievable, for the following reasons:

- IT budgets are finite; there is a limit on just how much resource and development effort can be employed to achieve ultralow latency (in any case, we need to be clear whether we are referring to latency within a switch, between ports, or the whole transaction path for a specific FIX message).
- Clients, trading platforms and exchanges are not ordinarily colocated. Some clients may be coming in over the Internet, others over high-speed dedicated links. Some platforms may be directly connected to the exchange, others not. The end-to-end transmission paths can vary widely.
- Between the client and the trading platform, there are typically intermediate systems (routers, firewalls, switches, application delivery controllers (ADCs) etc.), each potentially buffering and queuing packets to ensure reliable transmission. There may be similar infrastructure variation between the trading platform and the exchange(s).
- FIX messages are not just sent on the wire, they are wrapped in layers of protocol to ensure reliable routing and message delivery over multiple hops. They are also processed and moved between application components and data stores. Significant time is lost packing/unpacking, decoding and processing these packets.
- System architecture techniques, processing and memory technologies are evolving. Unless custom hardware is developed specifically for the financial services space, there will always be compromises in the overall design, based on the need for general-purpose support, integration and management.
- Finally, speed is constrained by the laws of physics, since there is a lower bound at the speed of light corresponding to approximately 3.3 milliseconds per 1,000 kilometers of optical fiber. Even with a single direct connection, we cannot go faster than this.

FIX latency is therefore something we can optimize, but performance ultimately depends on several factors, some of which can be influenced by design. Financial institutions will therefore go to great lengths to remove some of the more obvious “big-hitters” in terms of delay – ensuring that end-to-end transaction paths are as short as possible, reducing the number of intermediate device hops, optimizing application and storage components, using dynamic routing of client traffic based on load and availability, and finally ensuring that all devices in the transaction path are completely characterized under worst-case conditions.

Application Fluency
One of the key features of designing a FIX-based trading and reporting system is the ability to steer FIX messages based on availability and service policy. For example, if there are multiple clients concurrently using a FIX-based platform, it may be important to consider features such as:

- Different service levels offered to clients (e.g., platinum, gold, silver, bronze etc.)
- Different client profiles based on trading volumes (e.g., volume, frequent, occasional user etc.)
- Difference service-level expectations based on proximity
To enable this level of control, the ADC/ASG needs to incorporate advanced deep-packet inspection (DPI) technology, the ability to assess server health and availability in real time, as well as the ability to hold policy tables (used to determine how best to steer sessions). A10 provides a powerful scripting language called aFleX to support DPI. aFleX is based on the Tool Command Language (TCL) and supports DPI and optional manipulation of transactions in real time, together with support for dynamic lookup tables to enable on-the-fly policy decisions. In the context of FIX applications, this means:

- The ability to assess FIX server status and availability
- The ability to examine incoming FIX messages and parse FIX-specific header information
- The ability to forward FIX sessions to specific servers based on policy, client identity and current server status

All of this can be achieved dynamically in real time and within a low latency context.

Application Security

It goes without saying that financial service solutions need to be both highly robust and highly secured. A10 provides a range of mature security solutions that complement the ADC and ASG range of appliances and virtual appliances, including:

- SSL offload (high performance, high scalability SSL using 1, 2 or 4K keys)
- SSL intercept (decryption for subsequent inspection of outgoing sessions)
- Authentication offload/pre-authentication (AAM)
- Web Application Firewall (WAF) and DNS Application Firewall (DAF)
- Access control lists (ACLs)

ACOS – Ideally Suited for High-Performance FIX Environments

One of the key differentiators between A10 and legacy “blade-bus” architectures is our innovation in multicore processing architecture – specifically the use of high-performance shared memory.

Traditional bus architectures suffer the inevitable latency associated with bus access contention and arbitration, and the consequent delay variability associated with crossing multiple blades back and forth across the bus.

By taking a radical look at systems architecture, A10 has extended traditional supercomputer techniques to scale concurrent performance linearly over multiple processors and multiple cores. A10’s advanced core operating system (ACOS) offers high performance primarily by communicating through shared memory and eliminating legacy bus arbitration and polling issues.

A10’s ACOS architecture is based on true 64-bit Symmetric Scalable Multicore Processing (SSMP) over multiple multicore processors and shared memory. This architecture has several characteristics that are especially attractive to low latency trading applications:

- Separate management and forwarding planes. Dedicated message forwarding cores are isolated from management, providing highly optimized and highly efficient packet forwarding and scalability.
- Shared memory architecture. This eliminates legacy interprocess communication (IPC), providing high-speed access to shared data within a full 64-bit address space.
- Internal fair load distribution, which includes deterministic message paths across multiple CPU cores (using patented techniques).
- Since ACOS does not use a conventional bus architecture (with associated bus arbitration/polling schemes), ACOS provides more predictable, more accurate packet forwarding and status characteristics – without the delays associated with a contended bus model.

ACOS’s architectural advantages manifest themselves as low latency and low jitter for FIX transactions, together with internal deterministic flow consistency (which greatly assists debugging). By separating the management plane, A10 appliances are also available for management and reporting functions even under extremely heavy loads.
A10 FIX Solutions: A Typical Use Case

A10’s FIX support offers low latency, low jitter characteristics that are ideal for a number of securities applications. In this section, we discuss how A10’s FIX support has been used recently in a foreign exchange application by a leading investment bank, specifically because of its low latency and consistently low jitter. As discussed earlier, forex returns are especially sensitive to small mispricing anomalies, therefore characterizing overall time-to-transact and the time required to view market data is extremely important.

Note that the results presented here were independently tested and reported by the bank’s internal operations staff, with identical tests run against several competing solutions. Based on an initial analysis of ADC feature maturity and DPI flexibility, the selection was finally narrowed down to A10 and one major competitor. We therefore discuss A10’s performance verses the competing solution to illustrate A10’s key strengths. These measurements represent transaction times through the appliance; from ingress to egress ports and small cable runs, and therefore characterize the key aspects of internal FIX processing behavior within an ADC appliance.

Requirements

There were several major requirements in this case, summarized as follows:

1. Keep overall FIX transaction latency low, ideally within 30 microseconds (on average) to be considered “a contender”
2. Ensure that variability between client transactions is minimized and relative fairness is maintained between clients (i.e., a preference for low bounded jitter)
3. Provide advanced health and status checking on the pool of FIX servers to ensure that backend system loading does not adversely impact higher volume forex clients in a contended situation
4. Include flexible and highly configurable DPI, to support dynamic policy-based FIX session-forwarding, based on client profile and server status

A10’s Key Advantages

A10 demonstrated a number of key strengths:

- **Low latency** – achieving substantially lower latency on average than the requirement
- **Low bounded, consistent jitter** – achieving a much lower delay variability than competitive solutions, with significantly smaller worst-case delays
- **Application fluency** – DPI and policy-based forwarding specifically for FIX sessions – based on our aFleX Scripting languages and advanced server health monitoring capabilities
- **High quality support** – A10’s ability to assist the client through the process, from proof of concept through to deployment, with consistently high caliber support and systems engineering staff who understand the financial space intimately

We discuss the technical advantages of A10’s solutions in more detail below.

Low Latency and Jitter

In the tests run independently by the customer for forex transactions, A10 was able to achieve an average latency of approximately 15 microseconds (well below the worst case entry level requirement of 30 microseconds).

More importantly, A10 demonstrated tightly bounded jitter characteristics, significantly better when compared to competitive solutions. A10 was consistently able to keep the maximum (i.e., worst case) latency below 50 microseconds under extensive testing, whereas the leading competitor demonstrated worst-case delays running up to 330 microseconds (see Figure 3).
Figure 3: Latency and jitter distribution around the mean for A10 against a leading competitor. Note that while the average delay (latency) is broadly similar, the worst-case delays are significantly different, with A10 peaking at around 50 microseconds and the competitor peaking at approximately 335 microseconds.

While these delays may seem very small, it is important to note that it is highly undesirable to exhibit such wide variability in delay, especially in applications such as forex. Variable delay can lead to unfair trading scenarios, both between internal clients and between other competitors. Traders need guaranteed market data response times, with the ability to execute within very short execution windows. A potential 1/30 second delay in receiving market data (or executing a buy order) could prove extremely costly.

Application Fluency

A10 Thunder™ ADC appliances were configured in front of a pool of FIX servers, using aFleX advanced DPI scripting, together with A10’s rich server load balancing and health check features. Information retrieved from health checks was used to support dynamic distribution of FIX traffic based on client transaction volume. In practical terms, a key requirement here is to manage server load given that there are a range of client usage profiles, avoiding the concentration of high volume users on the same servers where possible. The main characteristics of the configuration are:

- A10 incorporates advanced server health checks based on A10’s significant experience in server load balancing (SLB) and content caching.
- FIX clients can be identified based on the FIX “xxxxCompiD” fields [5].
- By using A10’s advanced DPI scripting language aFleX, it is possible to dynamically distribute FIX sessions based on client identity and dynamic policy lookup tables (configured and maintained by customer operations staff).
- Heuristics can be encoded using aFleX to determine how servers are to be weighted when allocating new FIX session flows from clients.

The result of this configuration, together with A10’s low latency solution, is a more evenly balanced session allocation across the FIX server pool based on knowledge of FIX client behavior, thereby improving transaction times for high value clients (by avoiding resource “hogging”), and improving scalability by more even load allocation.
Total Cost of Ownership (TCO)

Finally, another key benefit of using A10 solutions is that FIX support is available at no additional cost. A10 does not charge additional license fees for enabling FIX (unlike competitors), or any other ADC feature. When scaled up, this represents a significant saving in both CAPEX and OPEX, and provides customers with a transparent OPEX over the contract period (since the support cost stays flat, with no additional licenses added later). By simplifying licensing, this also reduces the chances of human error in recovery situations (such as licensing configuration errors, missing licenses, incorrect licenses etc.).

Conclusions

A10 solutions help financial institutions stay ahead, with technology innovation that both attracts new customers and increases existing customer loyalty. A10 provides solutions that scale, perform and offer 24x7x365 carrier class reliability.

FIX applications require consistent low latency message handling, and A10’s innovative SSMP architecture is designed specifically for high-performance, scalable and accurate message flow distribution. ACOS demonstrates more deterministic message handling than competitive solutions, by first maintaining low latency, but more importantly by bounding jitter more tightly. The direct benefit of this in a FIX environment is faster, fairer and more reliable service to clients.

ACOS demonstrates considerable flexibility in handling and understanding FIX messaging, enabling customers to construct and maintain tiered service levels based on client trade volume and server load. This level of application fluency directly benefits clients through improved and more deterministic transaction times, and it benefits FIX service providers by enabling more predictable scaling of server resource pools.

Using A10’s no-license model together with our minimal use of rack space, power and cooling, financial institutions can realize dramatic savings in CAPEX and OPEX, and they can consolidate core services with greatly reduced running and maintenance costs. A10’s no-license model also makes OPEX transparent for the total lifecycle of the solution, with no additional impacts on cost over the lifetime of a project.

Finally, A10 is absolutely focused on setting the industry benchmark for customer service and support. Today we support some of the largest and most demanding networks in the world, including Tier 1 carrier and major hosting centers. A10’s attention to detail in product quality, component selection and testing, coupled with our commitment to first-rate support, are highly compatible with the requirements of today’s financial institutions.

References

[2] FIX Trading Community. A large source of useful information, specifications, FIX history, application mapping etc. can be found at: http://www.fixtradingcommunity.org
Glossary

**Arbitrage** – buying in one market and selling in another, profiting on the temporary difference in pricing between markets. In markets such as foreign exchange, these price differentials can be extremely small and short-lived; therefore, a trader may typically execute a large volume of transactions over a very short period. Considered a “riskless profit” for the investor.

**Buy side firms** – investing institutions such as mutual funds, pension funds and insurance firms that typically buy large volume securities for money management purposes. May act as market makers in a security, using the trading infrastructure of sell side firms, but with more control over how the trade is executed.

**Direct market access (DMA)** – electronic trading facilities, typically supplied by independent firms, that allow buy side firms to access liquidity for securities they wish to buy or sell, and perform transactions at lower cost and lower risk of error (through the use of faster electronic trading).

**Indications of interest (IOI)** – a nonbinding underwriting expression showing a conditional interest in buying a security that is currently in registration (awaiting effectiveness by the SEC).

**Latency** – the delay between the transmission of information from a source and the reception of that information at a destination. Round-trip latency refers to the delay from source to destination, with acknowledgement back to the source.

**Jitter** – the distribution or variability in delay; a measure of the predictability and bounds of delay when compared over multiple transactions. Delay bounds do not have to be symmetrical, for example, while the average delay may be 100 ms, the worst case delay may be 300 ms and the best case delay might be 90 ms (i.e., the frequency of these anomalies determines the overall average, the standard deviation and the bounds of delay).

**Foreign Exchange** – a global market where currencies are traded virtually around-the-clock. Often abbreviated as “forex” and sometimes FX.

**High-frequency trading** – typically used by large investment banks, hedge funds and institutional investors, using a trading platform to automate large numbers of transactions at high speed, enabling traders to execute millions of orders and analyse multiple markets and exchanges in seconds, with the potential for institutions gaining significant competitive advantage.

**Securities** – financial instruments, typically divided into 1) debt securities; and 2) equities. Debt securities represent “money borrowed” that must be repaid (e.g., government and corporate bonds, certificates of deposit (CDs), preferred stock and collateralized securities such as CDOs and CMOs). Equities represent ownership interest held by shareholders in a corporation (such as a company stock).

**Sell side firms** – financial institutions involved in the selling, analysis or promotion of securities, typically providing public recommendations on upgrades, downgrades, target prices and investment opinion (e.g., brokerages, investment banks).

**Straight-through processing (STP)** – a process used by financial institutions to optimize transaction processing speed by allowing information entered electronically to be used during multiple stages of the settlement process, without requiring manual re-entry of that data.

For Further Information

If you would like further information or our pricing for support, please contact your local A10 Sales Team and view the support literature on our website at [http://www.a10networks.com](http://www.a10networks.com).

To email our sales team, use: sales@a10networks.com
About A10 Networks

A10 Networks is a leader in application networking, providing a range of high-performance application networking solutions that help organizations ensure that their data center applications and networks remain highly available, accelerated and secure. Founded in 2004, A10 Networks is based in San Jose, California, and serves customers globally with offices worldwide. For more information, visit: www.a10networks.com