

Five ways to optimise exchange connectivity latency

Every electronic trading algorithm has its own unique attributes impacting its operation. The general model is that the electronic trading algorithm takes in inputs, often composed of data disseminated by an exchange, and having processed them, may send out buy, sell or cancel orders to an exchange.

Trading is time sensitive as the prices change over time and become stale. Some trading algorithms base their decisions on infrequent events such as a company releasing its quarterly earnings or a central bank releasing economic figures. Other algorithms base their decisions on frequently changing events such as micro-changes in the market. The latter is often referred to as high-frequency trading (HFT).

The following guide suggests five methods that may be used to minimise the latency in the bi-directional communication between an electronic trading algorithm and an exchange and hence increase the potential for making successful trading decisions.

1. Minimise the latency between your network and the exchange gateway

As a general rule, two factors are key in minimising the network latency between two interconnected network endpoints:

1. The physical length of the connecting network media (copper cable/fibre)
2. The number and type of network devices (switches/routers) between the endpoints

Shorter network media reduce the propagation delay of the network traffic whilst passing through each network device adds latency. It must be noted that propagation delay varies across types of media.

There are two commonly used ways to connect to an exchange's network in order to receive data and place orders - either directly via a (typically 10GbE) dedicated exchange connection or indirectly via a shared exchange connection. A dedicated exchange connection, terminating within a colocation facility attached to the exchange, is designed to provide lowest possible latency – it is usually just a length of fibre patched through to an exchange edge-switch. On the other hand, sharing such a connection will, at the very least, require a switch or router to aggregate multiple individual connections together. Typical latency-optimised, off-the-shelf switches and routers add at least ~240 ns to a network round trip.

There is also the very real possibility that, as the connection to the exchange is shared, there is contention between orders from client-facing ports causing oversubscription and hence queueing within the network device which further adds latency (i.e. if two messages need to be sent over the media at the same time, one has to wait for the other to be transmitted before the other can begin).

There is also a trade-off between a dedicated and shared exchange connection, with a dedicated connection not only being more costly and requiring that the client terminate the connection on a network device supporting industry-standard routing protocols such as the border gateway protocol (BGP) and protocol-independent multicast (PIM). It is entirely possible, if the economics justify it and exchange rules allow it, to connect a single trading server (or FPGA) directly to a dedicated exchange connection.

2. Speed up your multicast market data ingress

By its very nature, multicast is a point-to-multipoint paradigm designed to get identical copies of the same packet to multiple consumers. Switches and routers either handle the delivery of multicast by flooding it out on ports in a particular virtual LAN (VLAN) or selectively forwarding it to a subset of ports. In both cases, doing so takes a minimum of 50 ns in the fastest off-the-shelf switches and routers.

Though this is impressively fast, it is possible to do this with an order of magnitude lower latency in two ways:

1. Pass the receive fibre of the incoming exchange connection through one or more passive fibre taps which split the optical signal across multiple trading server-connected ports, each containing an attenuated optical copy of the incoming signal, while introducing only a few nanoseconds of latency
2. Connect the exchange connection via a Layer 1 switch which can fan out the stream in as little as 4 ns

Though the passive fibre tap option may yield marginally lower latency than a L1 switch, it has a number of potential issues. Every time the Ethernet optical signal is split, it is attenuated. Splitting it passively more than a few times can easily result in the optical signal being marginal from the receiving transceivers' perspective causing bit errors and even loss of link. Furthermore, for longer lengths of exchange connection fibre from the exchange switch, there may be insufficient signal margin required for any kind of passive optical splitting at all. This solution is therefore really only practical when the exchange fibre connection arrives with low attenuation and for replicating the incoming stream to around four or fewer machines.

A Layer 1 switch on the other hand can offer a far more reliable solution as some models can perform signal regeneration and clock data recovery on the incoming stream. This ensures that the stream can be replicated as many times as required without any loss of signal integrity. Most Layer 1 switches also allow media conversion from fibre to direct-attach copper cable which reduces latency on the connections to trading server as well as cost. Another key benefit of using a Layer 1 switch is that some L1 devices offer Ethernet data and error counters allowing the port terminating the exchange connection to be monitored.

3. Speed up your exchange order egress

Unless each trading server has its own dedicated exchange connection, multiple servers sharing a connection will have to go through a switch or router. Latency-optimised switches and routers that support the routing protocols mandated by the exchanges can multiplex incoming packets out to an exchange connection in as little as ~190 ns.

There is, however, a low-latency alternative in the form of a specialised field-programmable gate array (FPGA) application known as a multiplexer or "Mux". For this specific use-case, these are designed to allow as many as 48 incoming 10GbE ports to share a single 10GbE exchange connection and are optimised for latency. Depending upon the device, latencies drop to ~45 ns.

Some vendor solutions even offer multiple Mux instances per device e.g. 4 × 12:1, 6 × 8:1 etc., allowing multiple exchange connections to share the same device. However, just offering ultra-low-latency muxing is not enough, as many exchanges require that devices terminating their exchange connections, support industry-standard protocols such as BGP and PIM. Hence the device running the Mux application needs to provide support for those protocols as well. It is also possible for a device to combine a Layer 1 switch and an FPGA-based Mux application to offer bi-directional ultra-low latency exchange connectivity in a single device that effectively replaces a traditional switch or router while reducing latency by at least 180 ns. (an example is the Arista 7130 E-Series)

4. Select the right low-latency network adapter for your trading application

There are many kinds of servers running electronic trading algorithms however they generally follow the same architecture when it comes to connecting to a 10GbE network. The central processing unit (CPU), which is often made by Intel, connects via a PCI Express bus to a 10GbE network adapter.

Network adapters are most commonly implemented using an application-specific integrated circuit (ASIC) but may also use an FPGA instead. There are a number of 10GbE network adapters available from vendors such as Broadcom, Intel, Mellanox and Solarflare. Each network adapter on the market essentially performs the same function; getting Ethernet packets to and from an application on the trading server.

There are two key factors that impact the latency with which they do it:

1. How quickly the network adapter can shuttle packets back-and-forth from the network to the server's memory across the PCI Express bus
2. Whether the network adapter offers a lower-latency alternative to the operating system's kernel adapter device driver and even the TCP/IP networking stack which communicates with the actual trading applications

These two factors are usually intertwined, with some vendors offering custom device drivers allowing the kernel to be bypassed with direct communication between the network adapter and user space - the trading application. In the case when the kernel is bypassed, as the majority of exchanges mandate that connections to their order gateways be via a TCP socket, either the adapter vendor needs to provide a lower-latency alternative to the kernel TCP/IP stack or the trading application needs to contain its own custom TCP implementation.

The choice of adapter will be a trade-off between the lowest possible latency and how much TCP/IP functionality is available to the trading application. The network adapter with the lowest latency to user-space does not necessarily offer the full TCP/IP functionality required by the trading application. There are however offerings in the market that combine low-latency network adapters coupled with an RFC-compliant TCP/IP stack, allowing the trading application to communicate with it as it would the kernel network stack.

5. Implement your trading application directly on an FPGA

The key advantage of implementing trading applications on FPGAs is primarily that the path from the 10GbE network to the FPGA fabric, where the application is implemented, is at least an order of magnitude lower-latency than the path from the network across the PCI Express bus to a server CPU.

Equivalent logic to that in a server network adapter that “speaks” 10GbE is available, when coupled with current FPGA transceivers, that allows an FPGA application to communicate with the network in ~50 ns (round-trip). There are other advantages that FPGA applications offer such as the ability to implement a trading application with consistent, deterministic latency characteristics.

The current generation of FPGAs is getting increasingly powerful. FPGA applications can be clocked faster, the quantity of resources within the FPGA, such as RAM, have increased by over an order of magnitude from previous generations without consuming more power. It has now become possible for multiple electronic trading applications to coexist simultaneously on the same FPGA.

FPGAs are also not just available on PCI Express boards designed to be inserted into a server. Some of the same vendors that offer Layer 1 switches also offer complete networked platforms integrating FPGAs within data centre-ready devices with integrated management processors and software development kits (SDK) for the FPGA platform. Some even offer multiple FPGAs within the same networked chassis e.g. Arista offers FPGA platforms which take up as little as a single rack unit (RU).

Writing an FPGA-based application is rather different to writing software in a high-level language such as C++ or Java. Hardware description languages (HDL) such as Verilog or VHDL are far lower-level and are written for an FPGA without the support of an operating system offering a plethora of standard libraries that can be leveraged by applications. This has not stopped an increasing number of trading firms writing trading applications, that natively run on FPGAs mainly to take advantage of the significantly lower latency they enjoy over server-based applications.

There are also vendors such as AlgoLogic, Enyx, Arrayware and Netcope that offer FPGA libraries implementing blocks of logic specific to electronic trading such as exchange feed handlers, order gateways and TCP/IP stacks allowing FPGA application developers to integrate these and only have to implement their electronic trading algorithms.

In summary

Not all ways of connecting to financial exchanges are the same - even when collocated with the exchange. Where minimising latency is a priority, the ideal solution is rather different to that where latency is of lesser importance.

For the lowest possible latency, a device containing an electronic trading algorithm implemented on an FPGA, plugged directly into a dedicated exchange connection is hard to beat. At the other extreme, a shared exchange connection via a financial service provider with an electronic trading application running on an off-the-shelf server would provide a far less deterministic, higher-latency path to the exchange.

Every electronic trading participant makes their own trade-offs, the topics discussed above cover most of the key ones.

Santa Clara—Corporate Headquarters

5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500

Fax: +1-408-538-8920

Email: info@arista.com

Ireland—International Headquarters

3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office

9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390

Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office

Global Tech Park, Tower A & B, 11th Floor

Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office

9 Temasek Boulevard

#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office

10 Tara Boulevard
Nashua, NH 03062

