The Advantages of Using Hardware-based TCP Fanout for High-Performance Messaging

When it comes to messaging, applications want to receive exactly and only the information they want, at a rate they can manage, without being affected by messages intended for other applications, or the misbehavior of other applications. Meanwhile, messaging administrators want simpler infrastructure, better isolation of misbehaving applications to prevent cascading failures and visibility and control over their entire deployment.

The best way to achieve all these goals is by sending each message only to interested recipients. However, the lack of network bandwidth and the inability to perform high volume message routing and fanout in software have prevented such per-client, customized delivery. As a result, multicast messaging systems today bundle messages into multicast packets and send them into the network where the network performs coarse-grained packet replication to each host that has joined the multicast group. Fine-grained message filtering on a per-topic basis is then performed on each subscriber machine for all messages on the multicast groups it has joined.

By embedding support for point-to-point 'unicast' distribution and message routing into purpose-built hardware, Solace has overcome these limitations. This means Solace enables point-to-point distribution over TCP with the performance, scalability, robustness and manageability required to give senders, subscribers and administrators exactly what they want:

- High fanout messaging with loose coupling to a heterogeneous layer 2 network
- Tolerant support for machines and applications of differing performance capabilities
- Client isolation from misbehaving applications
- More efficient use of client CPU resources
- Support for wide area networking
- Bullet proof security
- Centralized management and performance monitoring.

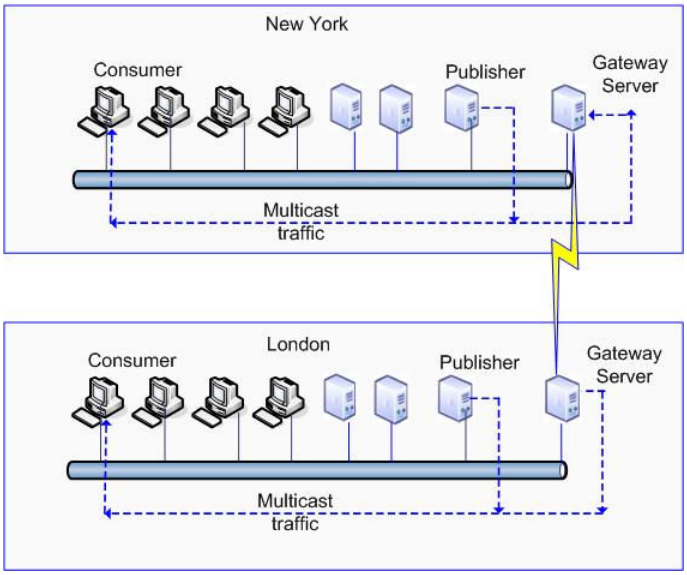
Solace Systems®

Copyright © Solace Systems, Inc.
http://www.solacesystems.com

Overview

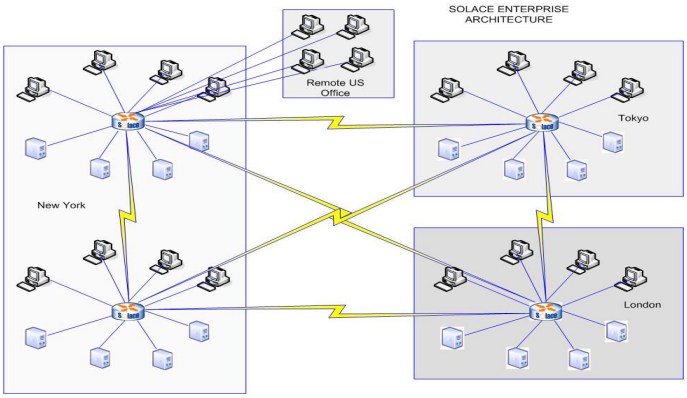
Messaging architectures come in two flavors: peer-to-peer and hub and spoke. In a peer-to-peer architecture there is no central message broker responsible for routing messages. Every client in the messaging infrastructure can publish and subscribe via UDP broadcast or multicast, and they often communicate directly with each other via unicast or TCP.

PEER TO PEER ARCHITECTURE



The picture to the right depicts this architecture and shows how different regions can be connected. The publisher to the right is sending messages on a topic via multicast. The consumer who wants to receive those messages needs to join the specific multicast group for that particular topic. Note that the publishers and consumers are not connected directly or via a central server. They are 'connected' to the same network and are clients within the same multicast domain. Different multicast domains can be bridged together usually via software gateways that bridge two or more networks or multicast domains to each other.

The server gateways to the right import and export traffic to and from the two regions that they are servicing (New York and London). The connection between these two servers is TCP over a WAN link at a substantially lower throughput capacity than the local network. Peer-to-peer architectures can provide guaranteed messaging services, and there are product-specific architectures for doing so.



In a hub-and-spoke architecture, clients (publishers and subscribers) all connect to a central broker. It is the broker's responsibility to manage all client connections, route messages from publishers to subscribers, and offload messaging functions from the clients. Brokers can be neighbored with and route messages to each other, so a publisher connected to Broker A can deliver a message to a subscriber connected to Broker B. Solace employs such a hub-and-spoke model, but implements the broker in hardware rather than software. This diagram shows how Solace is typically deployed in a global environment.

The major benefit of the peer-to-peer architecture is the elimination of machine hops and the high fanout. This is of most value in a small multicast environment where there is high volume, a need for low latency and high fanout (a high degree of overlap in client subscription lists). Traditional software-based hub-and-spoke solutions can't cope with the volume and fanout requirements since the broker needs to make a new copy of the same message for each subscriber. For example, if 10 subscribers are interested in IBM, a unique IBM quote will be sent separately to each subscriber. This represents a fanout of 10. Multicast allows the publisher to publish a single time and use the network capabilities of multicast to have that message replicated and delivered to all clients that have joined that multicast group – whether they want this message or not.

In this scenario, a central broker must transmit the same data repeatedly, once to each subscribing consumer. Software can't perform this fanout with very low latency or high

message rates, but Solace overcomes the issues of message routing and TCP fanout by embedding these functions into purpose-built hardware as described below.

Where high fanout is not an issue, a peer-to-peer model has little business value (and in fact has many drawbacks due to its complexity and lack of management) so most firms select at least two messaging solutions: one for high volume high fanout and one for all others. Often the same distinction can be found between reliable and guaranteed messaging products—firms will select one product for their reliable needs and another for guaranteed.

Solace supports both reliable and guaranteed messaging with both high throughput and low latency. Not only is it an enterprise solution, it also addresses the low latency high fanout requirements that have been traditionally satisfied via a multicast software-based solution.

Drawbacks of Multicast

Solace has architected its messaging system around TCP and Gigabit Ethernet because they are the best suited for both ultra-low latency messaging applications and enterprise-wide messaging deployments. Before one can understand the advantages of doing so, it's important to understand the weaknesses of—and problems introduced by—multicast.

Simply put, multicast-based messaging systems are limited in use to point applications under very controlled conditions due to their tight coupling to the Layer 2 network, lack of management and control, and lack of robustness. This section explains the key disadvantages of multicast.

Multicast-based messaging systems are limited in use to applications operating in tightly controlled conditions due to their tight coupling to the Layer 2 network, lack of management and control, and lack of robustness.

Tight coupling to Layer 2 network

Layer 2 networks are typically not uniform in that they include a mixture of 10GigE, 1GigE, 100Mbps and even 10Mbps links. The multicast groups joined by client devices determine the aggregate bandwidth consumption over the Layer 2 network to the client. Congestion points occur where there are speed mismatches, which then causes packet loss for all downstream clients, leading to client NACKs, retransmissions, network inefficiency and unpredictability. In the extreme, it can cause network collapse due to NACK and multicast storms. In short, tight coupling to the Layer 2 network makes the system fragile, and adding software routing gateways between layer 2 islands makes the system very complex and reduces performance considerably.

Difficulty dealing with "slow consumers"

Well-behaved applications can be negatively impacted by slow clients or poor network performance because there is no client isolation built into the architecture. When there are differences in the processing capabilities of devices or applications on a multicast bus, packets are lost at the client host and need to be negatively acknowledged or 'NACKed' to the publisher and retransmitted. This NACK processing adds burden to the publishing client application that is magnified based on the number of slow consumers, and therefore ends up affecting all consumers of information from this source.

Now the publishers, in addition to their need to publish out at already high traffic rates, become backlogged with honoring these re-requests. This backlog creates latency spikes. Often, the extra re-requested packets may be the direct cause of the storm, as consumers need to filter out packets already received and thus become slow consumers themselves that need to re-request lost packets.

Difficulty managing multicast groups

Setting up well-behaved multicast groups in any reasonable size layer 2 network is almost impossible. A group that behaves properly one day can become overloaded the next because of the dynamic nature of the market. On any given day, a news story may spur unusual spikes of market activity. Even if this spike is related to symbols that a given application isn't interested in, they must process (and discard) them because it is on a multicast group it is subscribed to. Rebalancing topics from one multicast group to another requires that all subscribers know what topics have been moved to what multicast transports. In some products, this coordination is left to applications and becomes another activity to coordinate across the client base.

Difficulty dealing with varying messages sizes

Multicast can only hope to deliver consistent latency and throughput when all messages are more or less the same size, and when there is a predictable pattern of message rates and sizes. This predictability can be found in market data environments to an extent, but in the real world there is other messaging activity that utilizes those same networks in parallel – traders are performing real-time analysis and entire portfolios are being published on the wire. These message sizes are much larger and wreak havoc with a multicast environment. Simply put, as soon as a publisher or a consumer does not follow 'best practices' or there is a small network misconfiguration, SLAs cannot be met and all bets are off.

Multicast can only hope to deliver consistent latency and throughput when all messages are more or less the same size, and when there is a predictable pattern of message rates and sizes.

Wasted CPU on client machines

Multicast requires that all receivers of a multicast stream receive the entire stream even though they may only want a small subset of the messages in that stream. Applications needing to subscribe to several topics must often join multiple multicast groups to receive these topics. Each client host, then, must expend significant CPU to receive all this packet traffic, process it in the operating system, transfer it to user space to process the messages in software to decide which ones the application wants and which to discard.

Lack of support for guaranteed messaging

There are several product-specific architectures to support guaranteed messaging – where message persistence can be performed centrally in message stores or fully distributed on the client machine. In either case, this introduces performance bottlenecks, management complexities and reliability issues and become impractical when scaling to an enterprise-wide solution.

Inability to cross the wide area

Using IP multicast across a WAN or MAN is expensive due to the need for IP routing products that must be purchased and managed, and because there is no message filtering performed – the entire contents of the multicast group(s) must be propagated over the WAN whether there are any consumers of the data or not, which leads to ongoing high bandwidth costs. Alternatively, TCP gateways can be used to filter messages by topic and use unicast IP between sites, but when implemented in software, these gateways typically yield poor performance. Most firms will follow the latter approach, but then these TCP gateways become the major bottleneck in the infrastructure and cannot keep up with high volume multicast traffic. Furthermore, some of these gateways require several levels of TCP fanout, adding extra machine hops, latency and complexity that causes more of the same problems they were hoping to solve.

Limited visibility and manageability

Because all applications communicate directly with each other over multicast without a broker, there is no single point of traffic monitoring and management control. It is often impossible to detect why a latency spike occurs and certainly not in enough time to avert disaster. In some cases, clients then deploy separate applications on each multicast segment to snoop and monitor traffic on those segments.

The drawbacks of this approach are numerous:

- Requires more servers and more software licenses
- Does not allow monitoring of any point-to-point traffic that also often occurs between the applications
- Can't report on packets that are lost in the network that it cannot see
- Only allows monitoring of transmitted traffic and does not allow monitoring of how receivers of information are coping with the traffic offered to them
- Does not allow any policy enforcement – only distributed monitoring

In some messaging products, the API exports the ability to gather performance information on each client application, but then it is up to the application developer to write code to gather and reliably report these metrics – which is added development cost and is information that often cannot be counted upon in the face of application misbehaviour. Since there is no central server, the environment itself is quite complex, and support people often don't know where to start chasing down a performance problem.

Security

With multicast, any application can publish on any topic and any application can receive on any topic. This introduces the risk of malicious or inadvertent insertion of incorrect data into topic streams or uncontrolled use of published data, which is problematic in cases where access controls must be provided to market data. For example, it is critical to create entitlements and access control lists to secure the environment, ensure its integrity and preserve the existing SLA. Consider a new publisher that just starts publishing 150,000 messages a second without being configured in advance. Regulatory requirements such as those of the Sarbanes-Oxley Act mandate that businesses prevent publishers who have not been permissioned from participating in a message flow. This is nearly impossible to prevent in a peer-to-peer architecture –there must be some central server than maintains permissions and controls and can prevent someone from publishing at will.

With multicast, any application can publish or receive on any topic. This introduces the risk of malicious or inadvertent insertion of incorrect data into topic streams, and the unauthorized use of information.

Benefits of Solace's Appliance

Different customers and use cases value the benefits of the Solace TCP distribution system differently. Some of its benefits relate directly to lower datacenter capital and operating expenses, others relate to higher performance, better stability and better extensibility than multicast-based systems. The main benefits a Solace distribution system delivers are:

1. Reduced client footprint
2. Low latency, tight distribution
3. More robust, predictable performance
4. Centralized management and monitoring
5. Reduced day-to-day management
6. Improved scalability
7. Robust security
8. Increased architectural flexibility

Reduced Client Footprint

As described earlier, each client in a multicast-based system must receive each packet on all the multicast groups it has joined and process all the data on those groups. With Solace, message filtering is performed centrally, in hardware by the Solace Router rather than on each client device. This frees up considerable CPU for application use – be it algorithmic processing or other. Also, there is no need for additional servers and applications to monitor the multicast bus at various locations. Per-client filtering alone can result in considerable savings in server upgrade costs due to (a) the reduced packet load initially and (b) the new paradigm where each server need not keep up with the entire rate of increase of market data rates, but rather only the message rate it requires.

Low Latency, Tight Distribution

In real world multicast deployments, as opposed to lab tests, multiple uncoordinated publishers produce messages into many multicast groups. This typically results in a large volume of multicast traffic, and therefore interrupt coalescing must be turned on to some level on all client machines to reduce the rate at which interrupts are generated to avoid interrupt storms. With Solace message filtering and TCP, the only packets received by the client machine are those containing messages desired by the client application, thereby considerably reducing the packet arrival rate and allowing interrupt coalescing to be turned off completely in low latency deployments. This typically saves 40usec on average and closer to 90usec at the 99.9th percentile compared to default Linux configurations. Combine this with the fact that message filtering is done faster in hardware by Solace appliances than in software and the net result is a lower latency system with a tighter latency distribution.

With Solace, the only packets received by the client machine are those containing messages desired by the client application, thereby considerably reducing the packet arrival rate and allowing interrupt coalescing to be turned off in low latency deployments.

More Robust, Predictable Performance

As a messaging system scales to serve a large number of clients, the robustness and predictability of the system becomes even more critical. It cannot be tolerated that the misbehaviour of a single client impact the performance of other clients. Solace achieves this in several ways:

- **Slow consumer**. Messages are queued per-consumer and delivered via unicast by the Solace router via TCP. When a consumer is slow, the per-consumer queue on the router grows in a bounded manner and the TCP window may close. However, this is significantly different to continued packet transmissions, discards, NACKs, retransmissions of multicast systems – with NACK storms being the ultimate instability potential. Solace handles slow consumers in a much more graceful, orderly manner.
- **Isolation from unwanted traffic spikes**. Since a consuming application receives only the messages it wants, it doesn't get hit by a high packet rate because there is a spike in messages on a multicast group it joined due to activity on topics it doesn't care about. The Solace router is able to absorb and route messages at millions per second and provides this isolation to the application. Otherwise, an algo working on financial stocks could be impacted by a barrage of packets caused by a spike in oil & gas that it doesn't care about.
- **Reduced Publisher Load**. Messages from publishers are sent to the Solace router for redistribution, and thus any retransmits due to packet loss are handled by the router without impacting the performance of the publishing application. Similarly, publishers need not keep large amounts of historical messages around in buffers for retransmission purposes.
- **Stability under control plane churn**. Solace's architecture employs a datapath that is completely in hardware (no operating systems, interrupts, etc.) and a control plane in software that uses completely separate processing resources. Therefore, as client applications connect/disconnect, add/remove subscriptions, these activities do not impact the datapath performance and similarly, a high message forwarding rate does not impact the connect and subscription capabilities of clients. The system remains completely stable and predictable under high volumes of both types of activity. Furthermore, the management plane also uses separate resources, thereby ensuring that Solace appliances are always manageable by operators no matter what the messaging load.
- **Tolerant to hardware faults**. Conditions such as half-duplex connectivity, faulty NICs, failure to auto-negotiate at a client are detected locally as an issue between the client and the appliance, rather than by the symptoms of NACK storms.
- **No multicast leaks/loops**. The danger of inadvertently bridging together multicast domains, causing looping multicast packets cannot exist in this unicast system.

With Solace, the only packets received by the client machine are those containing messages desired by the client application, thereby considerably reducing the packet arrival rate and allowing interrupt coalescing to be turned off in low latency deployments.

Centralized Management and Monitoring

Since all client applications connect via TCP to a Solace appliance, the appliance provides a "one-stop-shop" for management, monitoring and policy enforcement, as opposed to doing this in a distributed manner across all applications when using multicast. Also, the use of hardware to implement the datapath ensures that a large number of statistics can be generated in real-time – more than with software solutions – and without any performance impact, so these statistics are always turned on. Example uses are:

- Real-time troubleshooting of slow consumers, fast publishers or perpetually crashing applications.
- Real-time monitoring of resources such as: queue depths (instantaneous and high water marks), per-client transmit/receive message rates, dropped messages, retransmitted packets, TCP round trip times, subscription lists.
- Asynchronous notification of events of interest: client connect/disconnect, subscribe/unsubscribe to topics, queue thresholds exceeded, etc. via SYSLOG or via the message bus itself.
- This rich set of easily-accessed information allows for a better understanding of the operational environment, which in turn allows for better optimization of the entire architecture – since you can't optimized what you don't understand.

Reduced Day-to-Day Management

The effort and cost of day-to-day management of the messaging infrastructure is reduced compared to multicast systems in the following ways:

- Since multicast is not used, there is no need to create, traffic engineer, re-balance, etc multicast groups. There is also no need to maintain the topic to multicast group mappings and adjust over time
- Trouble shooting at all layers from a single management point dramatically reduces problem resolution time since you can distinguish between a problem with the network vs the application vs the messaging system. Wall Street estimates that one hour of trader downtime costs approximately $35,000, so the ability to quickly identify and address problems is compelling and has significant value
- There is no need to create and deploy additional multicast monitoring software on each multicast segment.
- Moves and changes of application location and deployment (eg. traders to different floors or buildings, Citrix clients, etc.) are simpler since the application need only connect to a Solace appliance using TCP – no multicast bus need be extended to the application.

Improved Scalability

Multicast-based systems distribute market data to several multicast groups. You then have the typical trade-off of many multicast groups with low message rates each or fewer multicast groups with higher rates each. Supporting many multicast groups comes with its own cost since each group must be monitored, traffic engineering and rebalanced, topic-to-group discovery is more complex and dynamic, but this offers more traffic reduction to consuming applications. Fewer groups are easier to manage but generate more traffic for client machines. The value of the trade-off depends on the customer and application but in any case, needs to be architected and managed. This problem becomes more acute as the total message rates increases. Without creating more groups, the traffic to each client continues to increase and must be processed by each client. If processing is not done in a timely manner, then packet discards occur and retransmissions are required. Once again, the dilemma is to create more multicast groups and more management, or continuously invest in better client hardware and networking just to throw away more data faster! This investment must be made even if the message rate required by the application remains modest.

Each Solace appliance can route millions of messages per second, and filters message streams to just what each application needs – so client hardware can grow based on the needs of the application, not total data volumes.

With the Solace architecture, an aggregate message rate of millions of messages per second is easily accommodated and then filtered to just what each application needs – so client hardware can grow based on the needs of the application – not based on total market data rates.

Robust Security

The Solace appliance can enforce authentication of client applications based on username and password. Then, based on this identity, the appliance can allow/disallow connectivity from certain IP subnets, for example to stop test application from inadvertently accessing a production system. The appliance can also enforce per-topic publish and subscribe permissions and generate alerts when violations occur. This can be used as transport level enforcement of entitlements for consumers and to ensure that unauthorized systems don't maliciously or otherwise publish data into a production topic stream.

Increased Architectural Flexibility

The Solace routing system provides architectural flexibility for growth that multicast systems do not:

- **Networking**. Solace can distribute into a heterogeneous network of 100Mbps, 1GigE, 10GigE without issue since each client is treated individually with TCP delivery. Unlike with multicast systems, there is no tight coupling to the layer 2 network.
- **Client machines**. Processing speeds of client machines need not all be matched. This is an issue with some multicast systems to avoid the slow consumer problem. With Solace, delivery is tailored to each client and thus a mixture of fast and slow clients is easily accommodated.
- **Client Location**. Clients can be located either on the same or different subnet as the publisher. Reachability can be through IP routers if necessary, but Solace also supports multiple GigE interfaces on the appliance which can then be in different subnets, again, providing less coupling to the underlying network.
- **Distribution over the WAN**. No additional WAN gateways are required to bridge between multicast domains since Solace inherently performs message routing between the appliances without additional hardware or software components.
- **Guaranteed Messaging**. Because it is broker-based, Solace offers natural support for failsafe guaranteed messaging with a high availability architecture at rates that dramatically exceed those of software brokers.

Because it is broker-based, Solace's solution supports failsafe guaranteed messaging with a high availability architecture at rates that dramatically exceed those of software brokers.

Overcoming the Drawbacks of Multicast

Solace's technology does not suffer from the limitations of multicast. Traditionally customers needed to resort to multicast for 2 reasons:

- TCP fanout in software was unable to deal with volumes and created too much latency since the same message needed to be copied multiple times
- There was insufficient bandwidth in the network to support wide TCP fanout.

Each one of these issues is addressed below.

TCP fanout

Solace does all of its TCP processing in hardware, and it is massively parallel. Five million messages can be delivered per second. The cost of delivering the same 100 byte message to a second subscriber is approximately 1.2 microseconds when sending over a 1 GBit interface and only 0.12 microseconds when sending over a 10GigE interface. Serializing the same message to the 20th subscriber adds only 2.4 microseconds of delay using 10GigE. In an environment where the fanout is very large, then multicast *might* be quicker, but even in that scenario it depends greatly upon the amount of context switching and filtering that the client is doing.

Large fanout has historically been necessary because messaging systems were unable to cope with thousands or hundreds of thousands of topics. This drove the need to reduce the total number of topics in the application domain, resulting in overly broad topics containing a superset of most subscribers' interest. This in turn introduced the need for subscribers to filter away unwanted messages which uses up badly needed CPU cycles. The inability to support a large number of subscriptions also means that many applications are subscribed to the same topic because of interest in some *subset* of this topic's data set. So topics that aren't granular enough drive high fanout.

Since Solace can support five million unique subscriptions, it is rare that the number of shared subscriptions will reach a high level of fanout. In most environments, if applications subscribe to just the data they want, there is no need for very high fanout. In our experience most program trading environments will have very small fanout. Larger fanout is required to support a large community of manual traders, but since they are trading manually, any serialization delay that is incurred due to a larger fanout is measured in a few extra microseconds, which does not register to the human eye.

With millions of topics available, subscribers can subscribe to *exactly* and *only* what they want. The benefits are significant:

- More efficient client-based subscriptions, which removes the need to filter away unwanted data, resulting in more efficient CPU usage.
- Better use of the network, since only data that is subscribed to is sent to the subscriber—unsubscriblyed data never leaves the appliance.
- Reduced TCP fanout because of the ability to define topic subscriptions at a much finer grain.

As such, in most sites, with the proper use of subscriptions, message delivery is faster and more efficient than it would be using multicast. The more efficient topic routing with more intelligent TCP fanout (due to more precise subscriptions) will more than make up for a multicast approach that tries to send only one copy of each message, but relies upon client-side filtering to discard unwanted messages. In most cases, multicast distribution will result

In most cases, multicast distribution will result in greater end-to-end latency especially under high system load because the CPU cycles and context switching required to filter unwanted messages and the re-request of lost packets will have a more detrimental impact on performance than TCP fanout.

in greater end-to-end latency especially under high system load. This is because the CPU cycles and context switching required to filter away unwanted messages and the re-request of lost packets will have a more detrimental impact on performance than TCP fanout unless the fanout is substantial.

Network Bandwidth

Multicast became popular when 10 Mbit and 100Mbit networks were all that was available. Today almost everyone has 1Gbit, and 10GigE is either just around the corner or already available. The penalty of fanning out the same message multiple times and risk of saturating network resources are no longer issues. Furthermore, using a centralized device to provide TCP fanout means it's only necessary to ensure there is sufficient bandwidth to handle egress from the device itself. There is no need to scale networking infrastructure to support each subscriber and client across hundreds of subnets as is the case with multicast. This results in an easier to manage and less costly network infrastructure.

New Network Technologies

Solace makes use of the latest networking technologies both within our products and as recommendations to our customer to develop an end-to-end system with the lowest possible latency:

- Network Processors and FGPAs are used on Solace blades depending on which device is best for the task at hand. These technologies provide superior performance in rate and latency, ensure low latency variability and a superior "message rate per Watt" ratio as compared to software solutions.
- 10GigE network interfaces – this reduces transmission latency by an order of magnitude over 1GE. Furthermore, the interfaces can be separated and half the client population can connect to the first 10GigE interface and half can connect to the second.
- Optional client-side TCP hardware acceleration, currently in beta test, can provide zero copy of TCP data and complete kernel bypass. Solace's API will write directly to the acceleration card. Performance that was once achievable only via Infiniband/RDMA will be achievable through Ethernet, cutting client-side latency to under 10µs.
- Solace supports the use of high-speed cut-through switches to further reduce end-to-end latency. Additional latency through the cut-through switch is typically between 1-2µs and is independent of packet size.

Solace uses and recommends the latest networking technologies to help customers develop an end-to-end system with the best possible performance, manageability and robustness.

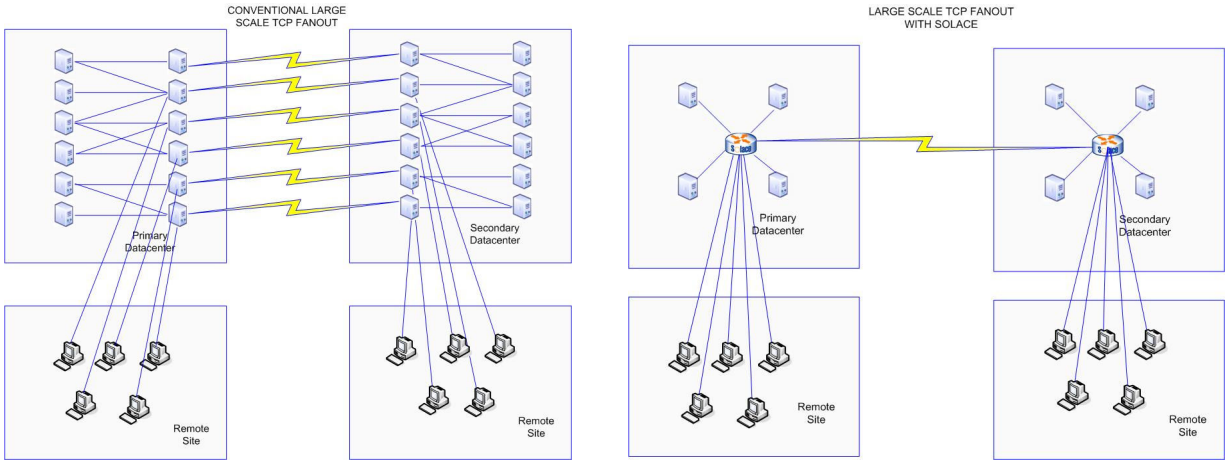
Large TCP Fanout

Many firms require hundreds if not thousands of TCP connections to 'remote' across a WAN or a MAN, or on different floors in the same building. They may be connections to the firm's satellite offices or clients, and they may be via a dedicated leased connection or a shared extranet like BT Radianz. Each location has a different networking capability and is often limited in network capacity, and it's not economical or possible to build out a huge networking infrastructure in these locations.

The goal then is to distribute data as quickly and efficiently as possible to different users with different profiles, supported by different networking bandwidths and capabilities. This is only achievable via TCP. To do this by conventional software-based solutions requires an enormous investment in hardware, communications and infrastructure.

Throughput is a key objective here because very often huge amounts of data need to be sent out within a narrow time window. Since each server is limited in its throughput capacity, traditionally the only way to solve the throughput requirement was through massive scaling in both hardware and communication links.

Solace's appliance can reduce the infrastructure by an order of magnitude while delivering data faster with greater efficiency and reliability. The following images depict how a Solace implementation simplifies the architecture and reduces the hardware and communications infrastructure.



Some of the other features that make the Solace implementation of TCP a compelling business decision:

- **Flexible TCP window sizing** – enables very large TCP buffers for more efficient throughput. Most companies cannot fully utilize their WAN links because of application or TCP limitations. They need to deliver data within a timeframe, but the only way to do this in a software-based solution is to have multiple (expensive) COMS links working in parallel.
- **Support for 17 Million reliable messages/sec and many thousands of concurrent client connections** – supporting this throughput and number of connections would require a very large server footprint in a server-based solution.

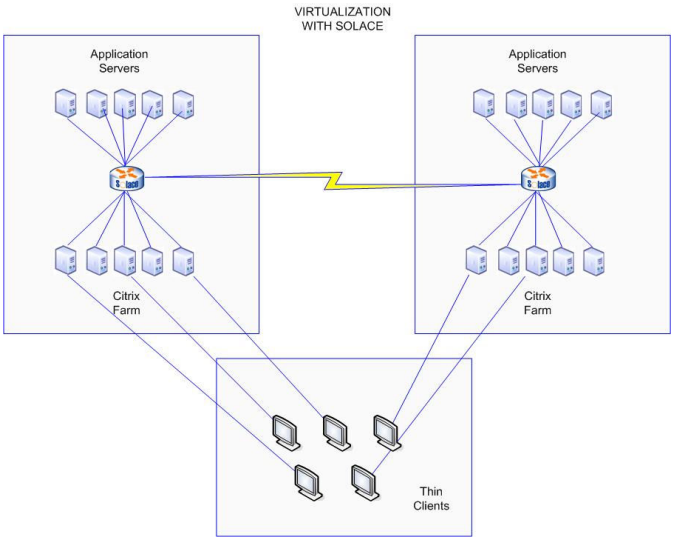
- **Support for 200,000 guaranteed messages/sec** – this message rate cannot be achieved by conventional software and again requires a large infrastructure to provide the same capability as a single solace appliance.
- **Automatic failover** – failover to another Solace appliance occurs automatically and with no downtime.
- **No impact of slow consumers** – a poorly behaving client application connecting to the Solace appliance has no meaningful impact on other clients, nor the rest of the messaging infrastructure.
- **Session-based Compression** reduces the amount of data transmitted over the wire to enable faster delivery and reduce COMS costs, especially across the WAN. Although the compression ratio is dependent on the data patterns, it is not uncommon to achieve 85% compression ratios.

Thin Client and Virtualization

The other trend that is gaining popularity is a drive to thin client applications and virtualized desktops. With virtualization, a user's actual workstation is housed in a datacenter and is no longer physically located at the user's desk location. In the past, consumers of multicast data were isolated to dedicated subnets built specially to support the traffic needs of that particular business unit. The network design that went into creating dedicated multicast-based subnets for user traffic based on their physical location is no longer achievable, or certainly not without a tremendous loss of flexibility in achieving the design goals of virtualization.

Maximum flexibility requires that a user be housed in any location in any datacenter without any regard to the other users sharing the same networking infrastructure, the business units that they belong to and the traffic patterns that they elicit. Furthermore he/she may be in one location one day and in another the next. Different thin client users will often be pointed to different Citrix server farms in different buildings. As such, the only feasible way to distribute data to these users is via TCP.

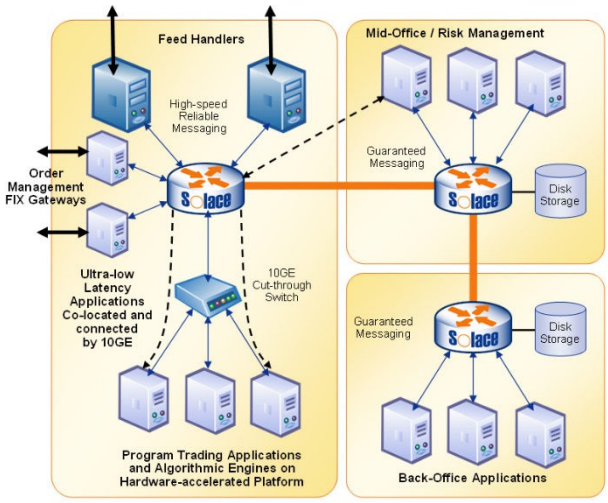
The picture to the right depicts the typical architecture with a Solace implementation.



Ideally all users within a Citrix farm will belong to the full spectrum of diverse business units. Firms would like to deploy users to these farms without regard to whether they are equities or options traders, investment bankers, operations support, or software developers. As such, there is no need to segment subnets in the farms to dedicated business units where the traffic patterns are already known. Since all data is delivered to the end-users via TCP, there is no need to build out expensive multicast infrastructure. Maximum flexibility is achieved and the Solace Appliances will still be delivering messages at the rate of five million messages per second.

End to End Enterprise Messaging

Solace provides a complete enterprise solution from front office to back office, from high speed low latency messaging to high throughput guaranteed messaging. The picture below depicts market data feeds coming in and being distributed using Solace's Topic Routing Blade. Market data is delivered with ultra low latency to algo trading engines using reliable messaging. The algo trading engines are hosted on hardware accelerated platforms and connected by 10 gigabit Ethernet via cut-through L2 switches to minimize latency.



Market data is consumed by program trading apps and orders are sent using guaranteed messaging to the order management systems and FIX gateways. Average end-to-end latency using guaranteed messaging is under 100 microseconds at 150,000 messages per second, with a very tight latency distribution at the 99.9th percentile.

The order management and FIX gateways may or may not be located in the same datacenter as the algo trading applications. Trade executions are then published to mid-office application using guaranteed messaging and they in turn feed back-office applications. The entire end-to-end solution is enabled via Solace technology.

Summary

By embedding support for point-to-point 'unicast' distribution and message routing into purpose-built hardware, Solace enables TCP-based message distribution with the performance, scalability, robustness and manageability required to meet the needs of publishers, subscribers and administrators.

To learn more visit solacesystems.com or call +1 613-271-1010.