Stampede FX Series Technology Descriptions

Compression

There is a difference in how compression works depending on whether the FX is in single-sided or two-sided mode. In single-sided mode, compression is applied to HTTP traffic that can be properly handled by the browser. This includes HTTP payload and JPEG images. GZIP compression is a standard, lossless compression, returning exactly what was sent after de-compression. On data that it can compress, typical compression ratios are 2.5:1 or better.

GZIP compression is handled on-the-fly from the servers to the clients. This reduces bandwidth consumption and improves application delivery and client response time. The FX Series uses GZIP compression to reduce the payload size to deliver more data across the satellite link, enabling more applications to be delivered and the ability to support more users. Squeezing the data reduces network traffic and accelerates the delivery of time-sensitive information. GZIP compression uses standard techniques to compress data sent to browsers. While compression exists in many forms throughout Web deployments, the FX Series is able to more effectively apply compression resulting in better compression ratios. The most common use of compression in Web environments is accomplished by enabling GZIP functionality at the Web server. This is useful for reducing the text portions of pages, but GZIP is not normally used for attachment compression or for inbound compression from the browser. In addition, GZIP cannot be used to compress HTTP headers or image data. When in two sided mode, in addition to the payload from the server to the client, the FX Series compresses all HTTP headers, application cookies, and all attachments in both directions.

Image reduction and smoothing is another compression technique that reduces the amount of data required to represent an image without significantly altering the visual perception of the image. This is accomplished in two ways. Smoothing reduces the high frequency components or the sharpness of an image. A moderate amount of smoothing can significantly reduce the amount of data. The quality factor of a JPEG image relates to the precision of the samples. Sample precision can be reduced without visible detection.

The goal of the JPEG quality and smoothing values is to reduce the amount of data while maintaining a usable image. Depending on the JPEG, the compression is often in the range 9:1. A number between 1 and 100 specifies the tradeoff between size of the JPEG data and quality of the original image. A higher number will retain a higher quality, but will not conserve as much bandwidth. If no value is specified, then the FX Series value is inherited from a higher level policy; a default value of 50 is used if no higher level policy is defined. Images that have been transformed are typically not significantly changed by running through the algorithm again. What this means is that if an image has been compressed with particular smoothing and quality factor, if the same factors are used again, the image is not significantly changed.

Caching

Caching is done in multiple ways, and has two effects. The first is to reduce the amount of data that traverses the link. The second is to bring data, or at least portions of the data, closer to the client which in effect reduces round trip time.

Object/File Caching

There is one form of caching available in single-sided mode: Object or File Caching. In this form of caching, the files are stored in the ADC memory or hard disk when any client downloads the data. Subsequent accesses to that same file or object are served by the ADC instead of going to the backend server.

Once a remote device is added, there are more caching options. Caching brings information closer to the end user by storing recently accessed data in local memory or on hard disk, which reduces the time it takes to bring back needed information and makes the user experience more positive and action oriented. While today's browsers maintain their own cache, they tend to be overly conservative. This means they will err on the side of requesting a new piece of data or object, usually when it really hasn't been changed. This not only impacts response time to the end-user, but also saturates bandwidth with unnecessary data transmission. Cache Differencing takes the concept one step further and maintains identical copies of the browser's cache at the local device and on the FX Series appliance. The FX Series then uses intelligent differencing technology to understand what data has actually changed, and then transfers only the changed data. The local device functions normally. However, with less data being transferred, satellite network utilization is improved and end user productivity is increased.

Traditionally, pages can be marked as cacheable and will have expiration dates. When they expire, they must be retrieved from the original server, resulting in additional traffic and data being transmitted across the satellite network. Within a two-sided environment, the FX Series remote appliance caches all pages returned to the browser (even pages that are marked as non-cacheable), and performs validation when needed to ensure that no stale data is returned to the browser. When the browser asks for a page or an item that has expired or been marked as non-cacheable, the FX Series remote appliance sends a validation request to the FX Series appliance at the head-end. If the FX Series appliance is aware of the last page the client cache contains and can compute differences in the page, it sends just the differences to an expired page or non-cached page. If the differences are too big, or if the FX Series appliance no longer has retained the last version that the client has, then the entire page is returned and subsequently cached for future possible differencing. The client in turn reconstructs the requested page, caches it, and returns it to the browser. Checksums are calculated by the FX Series appliance at the head-end and verified at the FX Series remote appliance so that pages will never be delivered incorrectly. While this technique adds value on expired pages, it is extremely effective for dynamic page generation.

An important aspect of the FX Series' Cache Differencing is the ability to perform differencing not only on HTML GET requests, but also on POST requests. This is significant because a) responses to posts are always marked non-cacheable, and b) most applications that are based on SOAP and XML (including most AJAX applications) issue SOAP requests via the HTML POST command.

Dynamic Data Suppression (DDS)

Dynamic Data Suppression is a form of de-duplication. When doing DDS, blocks of the file are coded and stored. The remote device is synchronized with the ADC device to keep the same data, and to use the same codes for representing the data. This can be done with no increase of link activity while the caches are being built. DDS comes into play when files with the same or similar content pass through the link with different names and/or different protocols. When this happens, file caching does not work, because it keys in on the actual URL, but DDS will work.

When using DDS, a file is broken up into blocks and stored in a block cache. Each block is assigned a unique identifier. These identifiers are small, usually less than 1% of the bytes of the original block. Both the server side and the client side build identical block caches when a file is transmitted.

In subsequent transmissions, in addition to storing blocks in the cache, the file being transmitted is searched for blocks that are cached. When they are found, the identifier is sent instead of the block. When a file is essentially the same as one previously transmitted, compression ratios of 100:1 are achievable.

Of course, when DDS is being successfully matched, not only will the amount of data be reduced, it will actually get there faster. This happens because there is significantly less data traversing the slowest and the highest latency part of the path. By significantly reducing the number of bytes used to represent the data, it effectively increases the link speed, getting the file to the user faster.

Microsoft® Update Caching

Intelligently caches Microsoft updates on the client side saving significant bandwidth attributed to "Patch Tuesday". The FX Series caching methodology handles the rather complicated procedures employed by Microsoft and other AV vendors to request updates by requesting "partial objects". When there are multiple users on the same link, this reduces the amount of data sent over satellite links to reduce bandwidth consumption and provide faster response times for end users.

The FX Series Remote can dramatically curb bandwidth consumption by caching software updates published frequently by Microsoft, Symantic, Adobe, Apple and many other leading software vendors. Most satellite service providers are aware of the bandwidth impact of "Patch Tuesday" –the day that Windows updates are distributed. The delivery of these updates is performed when software that resides on client devices downloads the new content in the background by requesting "partial content" over HTTP. The complex nature of "partial-content" HTTP requests thwarts the capabilities of most caching devices, however the FX Series Remote appliance caching engine can handle these requests. Once the content is cached by the FX Series Remote, subsequent retrievals by the updating agents that request "partial-content" will be satisfied by the FX Series Remote appliance, eliminating the need to repetitively transfer the same updates over satellite links.

Traffic Shaping

Traffic shaping or packet shaping is a process in which the ability to delay and sometimes drop is applied to certain traffic on the link in order to optimize the link to a pre-defined profile. There are basically two kinds of traffic on most networks. First, there are reliable protocols, which use some form of windowing and acknowledgments to control how much traffic is allowed to be "in the network". By selectively delaying some of the traffic, the natural response of the connection is to respond by slowing the rate of that connection. Second, there are also real-time protocols that typically send data at a fixed data rate. Some of this data is also very sensitive to latency. Typically this latency-sensitive traffic, if it is delayed long enough, might as well be dropped.

Traffic shaping consists of two steps. The first is to classify or filter the traffic and separate it into various queues. The second step involves draining the queues according to the appropriate drain algorithms.

In the FX Series products, classification takes place on a per packet basis. Each packet is directed into a queue, where it can be handled by the second phase of the traffic shaping engine. Classification can be done by VLAN, by source or destination IP address, source or destination port, by protocol, and by DSCP bits. Filters are defined using these fields, or combinations of these fields. The FX Series will support more than 10,000 filters. Each filter directs traffic into a queue, and the FX Series supports over a 1000 queues. More than one filter can direct traffic into the same queue. One of the issues in shaping is that traffic can match more than one filter. In the FX Series, filters are prioritized; the first filter that is matched is the filter that will be used.

Queues have priority. Up to 8 levels of priority are supported. Two basic drain algorithms are supported. The first, Strict Priority drains queues in order of priority. If there are any packets in a higher priority queue, they will be sent before a packet in a lower priority queue. Queues of the same priority will be handled in fair manner, traffic flows in a manner proportional to the total traffic in the queues. Because the highest priority queues suffer little or no latency, traffic that is sensitive to latency should be directed into the higher priority queues.

In the second, Min-Max, besides priority, there are two additional parameters. These are CIR, or committed information rate and MIR or maximum information rate. If the requested CIR for each queue can be met, it will be. By requested CIR, we refer to the lesser of the current traffic rate and the CIR. Because queues are prioritized, the higher priority queues will experience less latency than the lower. If the requested CIR cannot be met, then traffic will be dropped, starting with the lowest priority queues. The queues first delay and then drop the packets.

When more traffic can be supported than the requested CIR, each queue is allowed to drain traffic up to the MIR, again in priority order. This too is a requested MIR.

Traffic Shaping with ACM Option

The FX Series can read the current data rate for two Comtech EF Data modems, the CDM-750 and the CDM-625. When these modems are operating in ACM mode, the FX Series will update its rate multiple times a second. The drain algorithms are followed for each of the queues, but the total traffic will match the link rate. As an ACM link gets impaired, the data rate will start to drop. If the link is full, this will, by necessity, require that some packets get dropped until the overall traffic gets in synch with the link rate. The FX Series traffic shaping will ensure that it is the lower priority traffic that gets dropped. Higher priority traffic will continue to be carried by the link.

Reliable Multicast

The FX Series has a reliable multicast, called the Multicator. Typical multicast protocols are based on UDP or similar non-reliable protocols. In this case, the information is sent and forgotten. If the clients get the data, so much the better, but if they don’t, then they are just out of luck. This is appropriate for many real-time multicasts where if the data isn’t received, it is no longer needed. However, when distributing large files to multiple sites, it is important that all the data be received correctly.

The Multicator consists of a transmitter, multiple receivers and a controller. At any time, any of the FX Series in the network can be a transmitter with all or some of the other FX Series units being receivers. The transmitter function can be passed from one FX Series appliance to another, but only one FX Series units can transmit at a time.

The transmit rate is a parameter that is set, and can be as high as 155 Mbps. It is limited by the licensed rate of the transmitter. Files are FTP’d from a client to the transmitter. It is reliably multicast to each of the receivers, and once received, each FTPs it to a server. Transmissions can consist of a single file, a set of files or a directory of files.

Instead of being ACK based, the Multicator uses NACKs. With high data rates and satellite latencies, an ACK based protocol would retransmit a significant amount of data that had already been received. Each receiver sends NACK messages identify any missing data back to the transmitter. The transmitter will resend any data that is NACK’d. The transmission session will continue until all receivers have notified the transmitter that the file has been received successfully. At this point, any other FX Series appliance can become the transmitter.

VLAN Trunking Support

The FX Series is designed to support trunked VLAN networks, where multiple 802.1Q tagged VLANs flow through the same physical connections. The FX Series can handle up to 1024 active VLANs for which it maintains separate routing tables and HTTP caches allowing it to support Virtual Routing and Forwarding (VRF) environments (where the same subnets can exist on different VLANs). Accelerated traffic maintains VLAN tag affinity end-to-end. In addition, the FX-1010 can be configured to add VLAN tags for inbound traffic from up to 8 different LAN ports.

Data that is trunked will also be optimized. All of the FX Series optimization is available for traffic whether it is trunked or not, including all caching.

TurboStreaming

The FX Series multiplexes large data objects using our patented TurboStreaming™ (multiplexed TCP sessions) that enables HTTP browser traffic to be intermixed across multiple "pipelines". All browser activity is optimized, including the network-intensive polling associated with Web 2.0 and AJAX applications. A key advantage of TurboStreaming is that communication resources can be shared across multiple applications, and all HTTP requests and responses from any application (including multiple browsers) are intermixed simultaneously across multiple concurrent sessions.

TurboStreaming serves as a platform for the consolidation and aggregation of all Web-based traffic from a given user. Multiple HTTP protocol streams are logically aggregated across a few TCP sessions. Individual objects or pieces of objects can be split into any size and then multiplexed with other object data and reconstructed as needed SNSPs that deliver mixed payloads consisting of business-critical applications and data, streaming media, Voice over Internet Protocol (VoIP) and other network-intensive traffic. The end result is improved throughput and faster response time for the end user.

TurboStreaming enables the browser to open multiple pipelines (10s or even 100s) that communicate with the FX Series remote appliances. All of this data, from all browsers and all browser windows, is intelligently multiplexed over multiple TCP sessions back to the head-end FX Series appliance. This fully utilizes all available bandwidth, and enables the browser to function at its full potential. This is only possible because of advanced, industry leading two-sided acceleration technology.

Connection Management

Connection management removes the burden of establishing and terminating TCP connections from the web servers, allowing the server to handle more traffic. Stampede manages network connections in several ways to optimize the flow of data and reduce the impact on the network, application servers and end-user devices. The FX Series appliance maintains a consistent pool of connections between itself and the servers. The servers are then offloaded from managing the connections, and are isolated from inadvertent session disconnects.

With the remote appliances working with the head-end appliance, a persistent connection between the client and server is always maintained, even when the browser may close and reopen a session. These sessions are also multiplexed across multiple connections, improving throughput and response time. This persistent connection is extremely important for AJAX and Web 2.0 applications which constantly open and close sessions as they poll and access various Web services. The FX Series eliminates this potentially network intrusive overhead.

Load Balancing via WCCP

The Web Cache Communications Protocol (WCCP) allows satellite network service providers to transparently inject acceleration into their satellite network infrastructure by redirecting traffic flows in real-time to network devices such as the FX Series. WCCP has built-in load balancing, scaling, fault tolerance, and service-assurance (failsafe) mechanisms to ensure network devices can scale and have high-availability. For fault tolerance, if one of the FX Series appliances incurs a hardware failure, the WCCP-enabled router will stop sending traffic to that device and redirect traffic to the other FX Series appliances with zero down-time.

Load balancing via WCCP intelligently distributes the TCP and HTTP workload across multiple FX Series appliances. For flexible scalability, service providers can simply add an FX Series appliance to the cluster, and WCCP will split the traffic load among all the FX Series appliances. Up to 32 FX Series appliances can be set up within a cluster and dynamically load balanced.

WCCP enables network service providers to implement the FX Series into their network with greater deployment flexibility, without requiring the FX Series to be physically in-line. The FX Series can be deployed "virtually" in-line, hence, not all traffic is required to pass through the FX Series appliance. The network administrator programs the router to redirect traffic to the FX Service appliance in-bound and out-bound based on the router policies. This allows the administrators to make changes to their network environment by simply changing the router policies.

The FX Series (running WCCP) localizes content, and responds to content requests in order to reduce the amount of data going over the WAN. This improves application delivery response times, and allows the WAN link to support more traffic. Using WCCP, traffic is transparently redirected to the FX series appliance for TCP and HTTP acceleration, compression, caching and other optimization services.

With WCCP configured, the router redirects traffic to the FX Series to perform the application acceleration and WAN optimization functions. When an end user makes a request, the router intercepts the request, and redirects the request to the FX Series inside a generic routing encapsulation (GRE) frame to prevent any modifications to the original packet. The FX Series with WCCP can be used to transparently intercept traffic, so users don't have to make changes to Web browsers, and configure the FX Series as a proxy server to offload servers, accelerate application delivery and optimize the network.

Network Protocol Optimization

The FX Series provides application-aware modules for HTTP, CIFS, MAPI, POP3, SMTP, and FTP that dramatically reduce costly handshakes and intelligently apply compression to lower bandwidth consumption and reduce latency.

The products specialize in optimizing protocols by consolidating multiple transactions into a single transaction, which eliminates round-trips, performing cache differencing on dynamically generated content, and bi-directional data compression. In addition, our patented TurboStreaming technology enables the transfer of previously compressed objects up to five times faster through intelligent multiplexing across multiple TCP sessions.

- TCP and HTTP applications have chatty protocols that put added delay in satellite networks, as do delay-sensitive applications such as VoIP and applications such as Microsoft Exchange and CIFS.
- IT managers are placing thousands of applications on their satellite links. Many of these applications are mission-critical, and compete over a limited amount of bandwidth.

Advanced protocol optimizations drive significant improvements in bandwidth efficiencies and time savings (reducing payload and latency). WAN optimization and application acceleration technologies are deployed to improve satellite network performance and increase the amount of applications that can be delivered over the satellite link. The FX Series manages all TCP sessions, and handles establishing and tearing down TCP connections locally (at LAN speeds) to avoid satellite network congestion problems. This helps to increase link utilization and improve the user experience. TCP termination offloads the responsibility from servers having to handle the overhead imposed by the volume of TCP connections from web applications.

Additionally, application level multiplexed TCP streams take advantage of all other TCP or protocol optimization done at the link level, and application-level handshakes are eliminated by consolidating transaction requests.

Contact us for more information
U.S. & Canada: +1.800.763.3423
Outside U.S. & Canada: +1.937.291.5035