1

Acceleration Techniques and Design Fundamentals

This chapter describes how the SteelHead accelerates data, the factors you need to consider when designing your SteelHead deployment, and how and when to implement the most commonly used SteelHead features.

How SteelHeads accelerate data

The causes for slow throughput in WANs are well known: high delay (round-trip time or latency), limited bandwidth, and chatty application protocols. Large enterprises spend a significant portion of their information technology budgets on storage and networks. Much of the budgets are spent to compensate for slow throughput by deploying redundant servers and storage and the required backup equipment. SteelHeads enable you to consolidate and centralize key IT resources to save money, simplify key business processes, and improve productivity.

RiOS is the software that powers the SteelHead and Client Accelerator. With RiOS, you can solve a range of problems affecting WANs and application performance, including:

- insufficient WAN bandwidth.
- inefficient transport protocols in high-latency environments.
- inefficient application protocols in high-latency environments.

RiOS intercepts client-server connections without interfering with normal client-server interactions, file semantics, or protocols. All client requests are passed through to the server normally, although relevant traffic is accelerated to improve performance.

Data streamlining

With data streamlining, SteelHeads and Client Accelerator can reduce WAN bandwidth utilization by 65 to 98% for TCP-based applications.

Scalable data referencing

In addition to traditional techniques like data compression, RiOS also uses a Riverbed proprietary algorithm called *scalable data referencing* (SDR). RiOS SDR breaks up TCP data streams into unique *data chunks* that are stored on the hard disks (RiOS *data store*) of the device running RiOS (a SteelHead or Client Accelerator host system). Each data chunk is assigned a unique integer label (*reference)* before it is sent to a peer RiOS device across the WAN. When the same byte sequence occurs in future transmissions from clients or servers, the reference is sent across the WAN instead of the raw data chunk. The peer RiOS device (a SteelHead or Client Accelerator host system) uses this reference to find the original data chunk on its RiOS data store and reconstruct the original TCP data stream.

Files and other data structures can be accelerated by data streamlining even when they are transferred using different applications. For example, a file that is initially transferred through CIFS is accelerated when it is transferred again through FTP.

Applications that encode data in a different format when they transmit over the WAN can also be accelerated by data streamlining. For example, Microsoft Exchange uses the MAPI protocol to encode file attachments prior to sending them to Microsoft Outlook clients. As a part of its MAPI-specific accelerated connections, the RiOS decodes the data before applying SDR. This decoding enables the SteelHead to recognize byte sequences in file attachments in their native form when the file is subsequently transferred through FTP or copied to a CIFS file share.

Bidirectional synchronized RiOS data store

Data and references are maintained in persistent storage in the data store within each RiOS device and are stable across reboots and upgrades. To provide further longevity and safety, local SteelHead pairs optionally keep their data stores fully synchronized bidirectionally at all times. Bidirectional synchronization ensures that the failure of a single SteelHead does not force remote SteelHeads to send previously transmitted data chunks. This feature is especially useful when the local SteelHeads are deployed in a network cluster, such as a primary and backup deployment, a serial cluster, or a WCCP cluster.

Unified RiOS data store

A key Riverbed innovation is the unified data store that data streamlining uses to reduce bandwidth usage. After a data pattern is stored on the disk of a SteelHead or Client Accelerator peer, it can be leveraged for transfers to any other SteelHead or Client Accelerator peer, across all accelerated applications. Data is not duplicated within the RiOS data store, even if it is used in different applications, in different data transfer directions, or with new peers. The unified data store ensures that RiOS uses its disk space as efficiently as possible, even with thousands of remote SteelHeads or Client Accelerator peers.

Adaptive acceleration

Using adaptive acceleration, SteelHead monitors its efficiency at improving network performance across all traffic. Where little benefit is gained, SteelHead bypasses that traffic, which frees the SteelHead’s resources to increase acceleration where acceleration is providing benefit. There are three types of bypass:

- auto-bypass based on hostname, where SteelHead identifies a pattern of poor performance for connections to the hostname for a minimum of 3 connections,
- ip-bypass based on server IP address, where SteelHead identifies consistently poor acceleration for connections going to this address. SteelHead only bypasses traffic after evaluating a minimum of 10 connections and identifies that many of the connections are not benefiting from acceleration. This is similar to the use of in-path bypass rules, except the feature operates without user interaction.
- hostname bypass, where SteelHead bypasses traffic for user-specified hostnames.

Connection details on the SteelHead console's Current Connection page show the bypass method used. Hostnames that are bypassed due to adaptive bypass are listed on the console's SSL Main Settings page.

When a hostname or IP is automatically bypassed, it is for a limited time. Default durations are two minutes for hostname bypass and ten minutes for IP address bypass.

Transport streamlining

SteelHeads use a generic latency acceleration technique called *transport streamlining*.

Overview of transport streamlining

TCP connections suffer a lack of performance due to delay, loss, and other factors. There are many articles written about and other information available regarding how to choose the client and server TCP settings and TCP algorithms most appropriate for various environments. For example, without proper tuning, a TCP connection might never be able to fill the available bandwidth between two locations. You must consider the TCP window sizes used during the lifespan of a connection. If the TCP window size is not large enough, then the sender cannot consume the available bandwidth. You must also consider packet loss due to congestion or link quality.

In many cases, packet loss is an indication to a TCP congestion avoidance algorithm that there is congestion, and congestion is a signal to the sender to slow down. The sender then can choose at which rate to slow down. The sender can:

- undergo a multiplicative decrease: for example, send at one half the previous rate.
- use other calculations to determine a slightly lower rate, just before the point at which congestion occurred.

A SteelHead deployed on the network can automate much of the manual analysis, research, and tuning necessary to achieve optimal performance, while providing you with options to fine tune. Collectively, these settings in the SteelHead are referred to as transport streamlining. The objective of transport streamlining is to mitigate the effects of WANs between client and server. Transport streamlining uses a set of standards-based and proprietary techniques to accelerate TCP traffic between SteelHeads. These techniques:

- ensure that efficient retransmission methods are used (such as TCP selective acknowledgments).
- negotiate optimal TCP window sizes to minimize the impact of latency on throughput.
- maximize throughput across a wide range of WAN links.

Additionally, a goal for selecting any TCP setting and congestion avoidance algorithm and using WAN acceleration appliances is to find a balance between two extremes: acting fair and being cooperative by sharing available bandwidth with coexisting flows on one end of the spectrum, or acting aggressive by trying to achieve maximum throughput at the expense of other flows on the opposite end of the spectrum. Being on the former end indicates that throughput suffers, and being on the latter end indicates that your network is susceptible to congestion collapse.

By default, the SteelHeads use standard TCP (as defined in RFC 793) to communicate between peers. This type of TCP algorithm is a loss-based algorithm that relies on the TCP algorithm to calculate the effective throughput for any given connection based on packet loss. Additionally, SteelHead peers automatically implement extensions to RFC 793 to provide further enhancements for congestion control beyond that of a standard TCP connection.

Alternatively, you can configure SteelHeads to use a delay-based algorithm called *bandwidth estimation*. The purpose of bandwidth estimation is to calculate the rate, based on the delay of the link, to recover more gracefully in the presence of packet loss.

In higher-throughput environments you can enable high-speed TCP (HS-TCP), which is a high-speed loss-based algorithm (as defined in RFC 3649) on the SteelHeads to achieve high throughput for links with high bandwidth and high latency. This TCP algorithm shifts toward the more aggressive side of the spectrum. Furthermore, you can shift even further toward the aggressive side of the spectrum, sacrificing fairness, by selecting the maximum TCP (MX-TCP) feature for traffic that you want to transmit over the WAN at a rate defined by the QoS class.

Configuring MX-TCP through the QoS settings leverages QoS features to help protect other traffic and gives you the parameters, such as minimum and maximum percentages of the available bandwidth that TCP connections matching the class can consume. Although not appropriate for all environments, MX-TCP can maintain data transfer throughput in which adverse network conditions, such as abnormally high packet loss, impair performance. Data transfer is maintained without inserting error correction packets over the WAN through forward error correction (FEC). MX-TCP effectively handles packet loss without a decrease in throughput typically experienced with TCP.

The TCP algorithms that rely on loss or delay calculations to determine the throughput should have an appropriate-sized buffer. You can configure the buffer size and choose the TCP algorithm in the Transport Settings page. The default buffer is 262,140 bytes, which should cover any connection of 20 Mbps or less with a round-trip delay up to 100 ms. This connection speed and round-trip delay composes most branch office environments connecting to a data center or hub site.

The following list is a high-level summary of each SteelHead TCP congestion avoidance algorithm:

- Standard TCP - Standard TCP is a standards-based implementation of TCP and is the default setting in the SteelHead. Standard TCP is a WAN-friendly TCP stack and is not aggressive towards other traffic. Additionally, standard TCP benefits from the higher TCP WAN buffers, which are used by default for each connection between SteelHeads.
- Bandwidth estimation - Bandwidth estimation is the delay-based algorithm that incorporates many of the features of standard TCP and includes calculation of RTT and bytes acknowledged. This additional calculation avoids the multiplicative decrease in rate detected in other TCP algorithms in the presence of packet loss. Bandwidth estimation is also appropriate for environments in which there is variable bandwidth and delay.

- **HighSpeed TCP (HS-TCP)** - HS-TCP is efficient in long fat networks (LFNs) in which you have large WAN circuits (50 Mbps and above) over long distances. Typically, you use HS-TCP when you have a few long-lived replicated or backup flows. HS-TCP is designed for high-bandwidth and high-delay networks that have a low rate of packet loss due to corruption (bit errors). HS-TCP has a few advantages over standard TCP for LFNs. Standard TCP will *backoff* (slow down the transmission rate) in the presence of packet loss, causing connections to under use the bandwidth.

  Also, standard TCP is not as aggressive during the TCP slow-start period to rapidly grow to the available bandwidth. HS-TCP uses a combination of calculations to rapidly fill the link and minimize backoff in the presence of loss. These techniques are documented in RFC 3649. HS-TCP is not beneficial for satellite links because the TCP congestion window recovery requires too many round trips or is too slow. HS-TCP requires that you adjust WAN buffers on the SteelHeads to be equal to 2 x BDP, where bandwidth-delay product (BDP) is the product of the WAN bandwidth and round-trip latency between locations.

- **SkipWare Space Communications Protocol Standards (SCPS) per connection** - SCPS per connection is for satellite links with few or no packet drops due to corruption. SCPS requires a license.

- **SCPS error tolerance** - SCPS error tolerance is for satellite links that have packet drops due to corruption. You must have a license to activate SCPS error tolerance.

The following list is a high-level summary of additional modes that alter the SteelHead TCP congestion avoidance algorithm:

- **MX-TCP** - MX-TCP is ideal for dedicated links, or to compensate for poor link quality (propagation issues, noise, and so on) or packet drops due to network congestion. The objective of MX-TCP is to achieve maximum TCP throughput. MX-TCP alters TCP by disabling the congestion control algorithm and sending traffic up to a rate you configure, regardless of link conditions. Additionally, MX-TCP can share any excess bandwidth with other QoS classes through adaptive MX-TCP. MX-TCP requires knowledge of the amount of bandwidth available for a given QoS class because, provided that enough traffic matches the QoS class, connections using MX-TCP attempt to consume the bandwidth allotted without regard to any other traffic.

- **Rate pacing** - Rate pacing is a combination of MX-TCP and a TCP congestion avoidance algorithm. You use rate pacing commonly in satellite environments, but you can use it in terrestrial connections as well. The combination of MX-TCP and a TCP congestion avoidance algorithm allows rate pacing to take the best from both features. Rate pacing leverages the rate configured for an MX-TCP QoS class to minimize buffer delays, but can adjust to the presence of loss due to network congestion.

Connection pooling

Connection pooling adds a benefit to transport streamlining by minimizing the time for an accelerated connection to set up.

Some application protocols, such as HTTP, use many rapidly created, short-lived TCP connections. To accelerate these protocols, SteelHeads create pools of idle TCP connections. When a client tries to create a new connection to a previously visited server, the SteelHead uses a TCP connection from its pool of connections. Thus the client and the SteelHead do not have to wait for a three-way TCP handshake to finish across the WAN. This feature is called *connection pooling*. Connection pooling is available only for connections using the correct addressing WAN visibility mode.

Transport streamlining ensures that there is always a one-to-one ratio for active TCP connections between SteelHeads and the TCP connections to clients and servers. Regardless of the WAN visibility mode in use, SteelHeads do not tunnel or perform multiplexing and demultiplexing of data across connections.

TCP automatic detection

One best practice you can consider for nearly every deployment is the TCP automatic detection feature on the data center SteelHeads. This feature allows the data center SteelHead to reflect the TCP algorithm in use by the peer. The benefit is that you can select the appropriate TCP algorithm for the remote branch office, and the data center SteelHead uses that TCP algorithm for connections. If SteelHeads on both sides of an accelerated connection use the automatic detection feature, then standard TCP is used.

Client Accelerator TCP transport modes

This section briefly describes specific transport streamlining modes that operate with Client Accelerator.

HS-TCP is not the best choice for interoperating in a Client Accelerator environment because it is designed for LFNs (high bandwidth and high delay). Essentially, the throughput is about equal to standard TCP.

MX-TCP is a sender-side modification (configured on the server side) and is used to send data at a specified rate. When Client Accelerator is functioning on the receiving side, it can be difficult to deploy MX-TCP on the server side. The issue is defining a sending rate in which it might not be practical to determine the bandwidth that a client can receive on their mobile device because it is unknown and variable.

Tuning SteelHeads for high-latency links

We recommend that you gather WAN delay (commonly expressed as RTT), packet-loss rates, and link bandwidth to better understand the WAN characteristics so that you can make adjustments to the default transport streamlining settings. Also, understanding the types of workloads (long-lived, high-throughput, client-to-server traffic, mobile, and so on) is valuable information for you to appropriately select the best transport streamlining settings.

The settings described in this chapter approximate when you can adjust the transport streamlining settings to improve throughput.

TCP algorithm selection

The default SteelHead settings are appropriate in most deployment environments. Based on RTT, bandwidth, and loss, you can optionally choose different transport streamlining settings. A solid approach to selecting the TCP algorithm found on the Transport Settings page is to use the automatic detection feature (auto-detect) on the data center SteelHead. The benefit to automatic detection is that the data center SteelHead reflects the choice of TCP algorithm in use at the remote site. You select the TCP algorithm at the remote site based on WAN bandwidth, RTT, and loss.

A general guideline is that any connection more than 50 Mbps can benefit from using HS-TCP, unless the connection is over satellite (delay greater than 500 ms). You can use MX-TCP for high data rates if the end-to-end bandwidth is known and dedicated.

When you are factoring in loss at lower-speed circuits, consider using bandwidth estimation. When planning, consider when packet loss is greater than 0.1%. Typically, MPLS networks are below 0.1% packet loss, while other communication networks can be higher. For any satellite connection, the appropriate choices are SCPS (if licensed) or bandwidth estimation.

WAN buffers

After you select the TCP algorithm, another setting to consider is the WAN-send and WAN-receive buffers. You can use bandwidth and RTT to determine the BDP. BDP is a multiplication of bandwidth and RTT, and it is commonly divided by 8 and expressed in bytes. To get better performance, the SteelHead as a TCP proxy typically uses two times BDP as its WAN-send and WAN-receive buffer. For asymmetry, you can have the WAN-send buffer reflect the bandwidth and delay in the transmit direction, while the WAN-receive buffer reflects the bandwidth and delay in the receive direction. You do not have to adjust the buffer settings unless there is a are only a few connections and you want to consume most or all of the available WAN bandwidth.

Application streamlining

You can apply application-specific acceleration for specific application protocols. Application streamlining includes the following protocols:

- CIFS for Windows and Mac clients (Windows file sharing, backup and replication, and other Windows-based applications)
- MAPI and MAPI over HTTP, Microsoft Exchange
- NFSv3 for UNIX file sharing
- TDS for Microsoft SQL Server
- HTTP
- HTTPS and TLS
- IMAP-over-SSL
- Oracle 9i, which comes with Oracle Applications 11i
- Oracle10gR2, which comes with Oracle E-Business Suite R12
- Lotus Notes 6.0 or later
- Encrypted Lotus Notes
- ICA Client Drive Mapping
- Multi-stream ICA and multi-port ICA support

Protocol-specific acceleration reduces the number of round trips over the WAN for common actions and helps move through data obfuscation and encryption by:

- opening and editing documents on remote file servers (CIFS).
- sending and receiving attachments (MAPI and Lotus Notes).

- viewing remote intranet sites (HTTP).
- securely performing RiOS SDR for SSL-encrypted transmissions (HTTPS).

For more information about application streamlining, see the *SteelHead Deployment Guide - Protocols*.

Management streamlining

Management streamlining simplifies the deployment and management of hardware and virtual appliances. It includes:

- autodiscovery, which enables appliances and Client Accelerator endpoints to automatically find peer server-side appliances and begin to accelerate traffic. Autodiscovery avoids the requirement of having to define lengthy and complex network configurations on SteelHeads.
- SCC, which enables the configuration, management, and monitoring of many appliances from a single location. It also provides a single view of the overall acceleration benefit and health of the SteelHead network.
- Client Accelerator Controller, which enables the configuration, management, and monitoring of remote Client Accelerator endpoints. Client Accelerator Controller enables you to see who is connected, view their data reduction statistics, and perform support operations such as resetting connections, pulling logs, and automatically generating traces for troubleshooting. You can perform all of these management tasks without end-user input.

Data store synchronization

Data store synchronization enables pairs of local SteelHeads to synchronize their data stores with each other, even while they are accelerating connections. RiOS data store synchronization is typically used to ensure that if a SteelHead fails, no loss of potential bandwidth savings occurs, because the data segments and references are on the other SteelHead.

You can use RiOS data store synchronization for physical in-path, virtual in-path, or out-of-path deployments. You enable synchronization on two SteelHeads: one as the synchronization primary, and the other as the synchronization backup.

The traffic for RiOS data store synchronization is transferred through either the SteelHead primary or auxiliary network interfaces, not the in-path interfaces.

RiOS data store synchronization is a bidirectional operation between two SteelHeads, regardless of which deployment model you use. The SteelHead *primary* and *backup* designations are relevant only in the initial configuration, when the primary SteelHead RiOS data store essentially overwrites the backup SteelHead RiOS data store.

Data store synchronization requirements

The synchronization primary and its backup:

- must have the same hardware model.
- must be running the same product version.

- do not have to be in the same physical location. If they are in different physical locations, they must be connected through a fast, reliable LAN connection with minimal latency.

Note: Before you replace a synchronization primary for any reason, we recommend that you make the synchronization backup the new synchronization primary. Therefore, the new primary (the former backup) can warm the new (replacement) SteelHead, ensuring that the most data is accelerated and none is lost.

For more details on data store synchronization, go to Knowledge Base article S12964.

Data store error alarms

You receive an email notification if an error occurs in the data store. The data store alarms are enabled by default.

If the alarm was caused by an unintended change to the configuration, you can change the configuration to match the old data store settings again and then restart the acceleration service (without clearing the data store) to reset the alarm. In certain situations you might need to clear the data store. Typical configuration changes that require a clear data store are changes to the data store encryption type or enabling the extended peer table.

To clear the data store of data, restart the acceleration service and click **Clear the Data Store**.

For more information about the RiOS data store error alarm, see the *SteelHead User Guide*.

Choosing the right SteelHead model

Generally, you select a SteelHead model based on the number of users, the bandwidth requirements, and the applications used at the deployment site. However:

- If you do not want to accelerate applications that transfer large amounts of data (for example, WAN-based backup or restore operations, system image, or update distribution), choose your SteelHead model based on the amount of bandwidth and number of concurrent connections at your site.
- If you want to accelerate applications that transfer large amounts of data, choose your SteelHead model based on the amount of bandwidth, the number of concurrent connections at your site, and the size of the RiOS data store.
- If you want to use SteelHead to enforce network Quality of Service (QoS), consider the SteelHead model QoS bandwidth in addition to the acceleration criteria you need.

You can also consider high availability, redundancy, data protection, or other design-related requirements when you select a SteelHead. SteelHead models vary according to the following attributes:

- Accelerated WAN bandwidth license limit
- Maximum number of concurrent TCP connections that can be accelerated
- Maximum number of possible in-path interfaces
- Availability of RAID and solid-state drives (SSD) for the RiOS data store
- Availability of fiber interfaces
- Availability of redundant power supplies

- Availability of hardware-based compression card
- Upgrade options through software licenses

All SteelHead models have the following specifications that are used to determine the amount of traffic that a single SteelHead can accelerate:

- **Number of concurrent TCP connections** - Each SteelHead model can accelerate a certain number of concurrent TCP connections.

  The number of TCP concurrent connections that you need for acceleration depends on the number of users at your site, the applications that you use, and whether you want to accelerate all applications or just a few of them. When planning corporate enterprise deployments, we recommend that you use ratios of 5 to 15 concurrent connections per user if full acceleration is desired, depending on the applications being used.

  If the number of connections that you want to accelerate exceeds the limit of the SteelHead model, the excess connections are passed through unaccelerated by the SteelHead.

  The TCP protocol only supports approximately 64,000 ports per IP address for outbound or inbound connections to or from a unique IP-Port pair. If your design or environment requires you to support more than 64,000 concurrent connections to a single IP address for acceleration, we recommend that you use multiple in-path interfaces or service port mapping.

  For details about service port mapping, go to Knowledge Base article S16309.

- **Number of encrypted TCP connections** - The SteelHead transparently accelerates encrypted connections by first decrypting it to accelerate the payload, and then reencrypting it for secure transport over the WAN. Examples of encrypted applications and protocols that a SteelHead can accelerate include HTTP Secure (or HTTPS/TLS) and Microsoft Outlook traffic (or encrypted-MAPI).

  If you have predominately encrypted connections, contact your Riverbed account team for assistance with determining the appropriate model.

- **WAN bandwidth limit** - Each SteelHead model has a limit on the rate at which it sends accelerated traffic toward the WAN. You might not need a SteelHead model that is rated for the same bandwidth available at the deployment site; however, we recommend that you make sure that the selected appliance is not a bottleneck for the outbound accelerated traffic. The accelerated WAN bandwidth limit applies only to accelerated traffic.

  When a SteelHead reaches its accelerated WAN bandwidth limit, it begins shaping accelerated traffic along this limit. New connections continue to be accelerated as long as the concurrent TCP connection count is not exceeded. The accelerated WAN bandwidth limit is not to be confused with the QoS WAN bandwidth limit that applies to both accelerated and pass-through traffic.

- **Data store size** - Each SteelHead model has a fixed amount of disk space available for RiOS SDR. Because SDR stores unique patterns of data, the amount of data store space needed by a deployed SteelHead differs from the amount needed by applications or file servers. For the best acceleration possible, the RiOS data store must be large enough to hold all of the commonly accessed data at a site. Old data that is recorded in the RiOS data store might eventually be overwritten by new data, depending on traffic patterns.