



## **Part 2: PLAN - BUILD**

### **Implement WAN Optimization**

- **Control Optimization with In-path & Peering Rules**
- **Deploy the SteelCentral Controller for SteelHead**

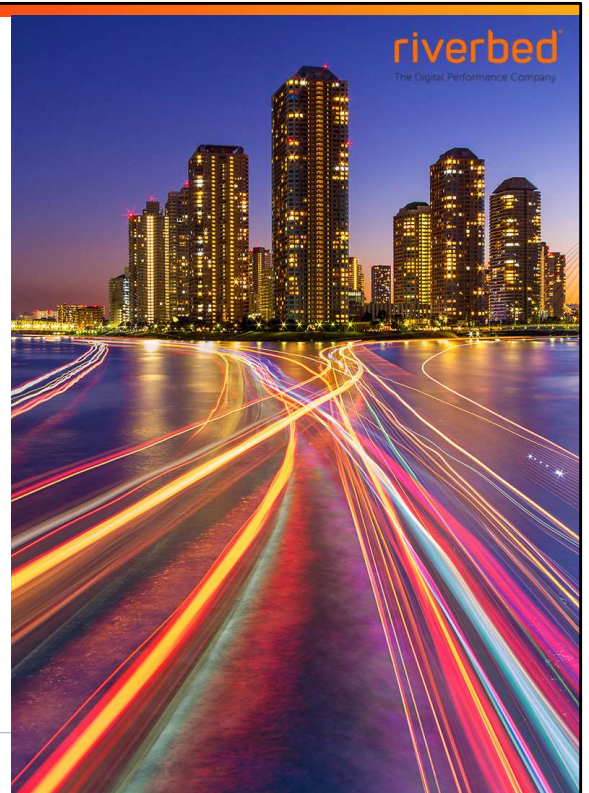


## Learning Objectives

After completing this module, you will be able to:

- Administer in-path rules.
- Control SteelHead peering.
- Optimize UDP traffic.

© 2020 Riverbed Technology, Inc. All rights reserved.







## In-Path and Peering Rules – Purpose

- To enable or disable optimization for each connection
- Processed in order
- The first match ends the process
- Therefore the order of the lists are very important

## In-Path and Peering Rules – Comparison

- In-path rules
  - How to deal with SYN messages on the *LAN* interfaces
    - Ergo: how to handle *locally initiated* connections
- Peering Rules
  - How to deal with Probes on *LAN OR WAN* interfaces

## In-Path Rules & Whether to Attempt Optimization

- A list of traffic classifications used to answer the following questions:

*Do we optimize? If so, how?*

- Consulted following a SYN *only* on the LAN interface
  - (If it is Logical in-path it assumes LAN without subnet-side rules, more later)
- An ordered list, where the first match ends the process
  - A SYN hits exactly *ONE* in-path rule
- There is an implicit Catch All rule at the end
  - Pre-empt this with an implicit deny if required for more control

Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cost Accrual	Kcutoff	Status
1	Pass Through	AS-IP*	AS-IP*Secure	All	TCP	--	--	--	Auto	--	Enabled
2	Pass Through	AS-IP*	AS-IP*Intracache	All	TCP	--	--	--	Auto	--	Enabled
3	Pass Through	AS-IP*	AS-IP*NET-Admins	All	TCP	--	--	--	Auto	--	Enabled
default	Auto Discover	AS-IP*	AS-IP*	All	--	None	Normal	Normal	Auto	No	Enabled

Description: Default In-Path Rule

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 7

In-path rules are an ordered list of fields a SteelHead appliance uses to match with SYN packet fields (for example, source or destination subnet, IP address, VLAN, or TCP port). Each in-path rule has an action field. When a SteelHead appliance finds a matching in-path rule for a SYN packet, the SteelHead appliance treats the packet according to the action specified in the in-path rule.

Most SteelHead appliance deployments use auto-discovery to initiate optimization. You can also manually configure SteelHead appliance pairing using fixed-target in-path rules. However this approach requires ongoing configuration, such as tracking new subnets that are present in the network, and which SteelHead appliances are responsible for optimizing particular subsets of traffic.

A SteelHead appliance optimizes a TCP connection by intercepting it at the two endpoints of a WAN connection:

- The client's TCP connection to the server is transparently terminated on the client-side SteelHead.
- The server-side SteelHead emulates the client's original TCP connection to the server.
- Optimized data is transferred to the client-side SteelHead over a separate TCP connection between the SteelHeads.
- Think of it as a TCP proxy with optimizations inserted in the flow.

## In-Path Rules – Six Types

- Auto Discover
- Fixed-Target
- FT: PMO
- Pass Through
- Discard
- Deny

The screenshot shows the 'In-Path Rules' configuration window. The 'Type' dropdown is open, displaying the following options: Auto Discover, Fixed-Target, Fixed-Target (Packet Mode Optimization), Pass Through, Discard, and Deny. The 'Auto Discover' option is currently selected. Other visible fields include Source Subnet, Destination Subnet, VLAN Tag ID, Preoptimization Policy (set to None), Latency Optimization Policy (set to Normal), Data Reduction Policy (set to Normal), Cloud Acceleration (set to Auto), Auto Kickoff (unchecked), Neural Framing Mode (set to Always), WAN Visibility Mode (set to Correct Addressing), Position, Description, and an 'Add' button at the bottom left.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 8

**Auto Discover** - Uses the auto discovery process to determine if a remote SteelHead appliance is able to optimize the connection attempting to be created by this SYN packet. By default, auto discover is applied to all IP addresses and ports that are not secure, interactive, or default Riverbed ports. Defining in-path rules modifies this default setting.

**Fixed-Target** - Skips the auto discovery process and uses a specified remote SteelHead appliance as an optimization peer. You must specify at least one remote target SteelHead appliance to optimize (and, optionally, which ports and backup SteelHead appliances).

**Fixed-Target (Packet Mode Optimization)** - Skips the auto discovery process and uses a specified remote SteelHead appliance as an optimization peer to perform bandwidth optimization on TCP and UDP, over both IPv4 and IPv6 connections.

**Pass Through** - Enables the SYN packet to pass through the SteelHead appliance unoptimized. No optimization is performed on the TCP connection initiated by this SYN packet. You define pass-through rules to exclude subnets from optimization. Note that traffic is also passed through when the SteelHead appliance is in admission control or in bypass mode.

**Discard** - Drops the SYN packets silently. The SteelHead appliance filters out traffic that matches the discard rules. This process is similar to how routers and firewalls drop disallowed packets: the connection-initiating device has no knowledge of the fact that its packets were dropped until the connection times out.

**Deny** - Drops the SYN packets, sends a message back to its source, and resets the TCP connection being attempted. Using an active reset process rather than a silent discard enables the connection initiator to know that its connection is disallowed.



## Streamline In-Path Rules With Labels

- Types:
  - Port Labels
  - Host Labels
  - Domain Labels
- In practice:
  - Simplifies in-path rule configuration
  - Applies to various system settings, including QoS configuration
  - Enables one rule to replace many

## Streamline In-Path Rules: Port Labels

- Default Labels:
  - Secure
  - Interactive
  - RBT-Proto
  - SteelFusion
- Allows handling a group of ports in the same way
  - For example, “Auto-Kickoff” for SteelFusion

Label	Ports
Interactive	7, 23, 37, 107, 179, 513-514, 1494, 1718-1720, 2000-2003, 2427, 2598, 2727, 3389, 5060, 5631, 5900-5903, 6000
RBT-Proto	7744, 7800-7801, 7810, 7820, 7850, 7860, 7870
Secure	22, 49, 88, 261, 322, 443, 448, 465, 563, 585, 614, 636, 684, 695, 902, 989-990, 992-995, 1701, 1723, 2252, 2478-2479, 2482, 2484, 2492, 2675, 2762, 2998, 3077-3078, 3183, 3191, 3220, 3269, 3410, 3424, 3471, 3496, 3509, 3529, 3539, 3660-3661, 3713, 3747, 3864, 3888, 3896-3897, 3995, 4031, 5007, 5661, 5723, 7674, 9802, 11751, 12109
SteelFusion	7950-7954, 7970

▼ SteelFusion 7950-7954, 7970

Editing Port Label SteelFusion:

Ports: 7950-7954, 7970

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 10

Port labels are names given to sets of port numbers. You use port labels when configuring rules. For example, you can use port labels to define a set of ports for which the same in-path, peering, QoS classification, and QoS marking rules apply. Default Port Labels include:

- **SteelFusion** – Use this port label to automatically pass-through traffic on Riverbed Granite ports 7950 - 7954 (data transfers), and 7970 (management).
- **Interactive** – Use this port label to automatically pass-through traffic on interactive ports (for example, Telnet, TCP ECHO, remote logging, and shell). **RBT-Proto** – Use this port label to automatically pass-through traffic on ports used by the system: 7800-7801 (in-path), 7810 (out-of-path), 7820 (failover), 7850 (connection forwarding), 7860 (Interceptor appliance), 7870 (SteelHead Mobile Controller)
- **Secure** – Use this port label to automatically pass-through traffic on commonly secure ports (for example, ssh, https, and smtps).

## Streamline In-Path Rules: Port Labels

- Similar to Port Labels, but for hosts
- Used by in-path rules and QoS
- Configured as:
  - Hostnames (DNS required)
  - IP addresses
  - IP subnets
- Not compatible with IPv6 Auto Discovery
- Only one default entry (read only)
  - “\_cloud-accel-saas\_” used for SaaS and is automatically populated
  - Internet access from the Primary is required

Summary of Hostname Resolution

- 0 Unique Hostnames
- 0 Checking DNS
- 0 Unresolvable

Hostnames are automatically resolved once every day.

Resolve Hostnames

Show resolved IPs for the hostnames in the table below

☒ Add a New Host Label ☐ Remove Selected

Name:

Hostnames/Subnets:

Entries can be separated with commas, spaces, or newlines.

Add New Host Label

Label	Hostnames	Subnets
_cloud-accel-saas_		

Host labels are names given to sets of hostnames and subnets to streamline configuration. Host labels provide flexibility because you can create a logical set of hostnames to use in place of a destination IP/ subnet and then apply a rule, such as a QoS rule or an in-path rule, to the entire set instead of creating individual rules for each hostname or IP subnet.

When you define hostnames in host labels (as opposed to subnets), RiOS performs a DNS query and retrieves a set of IP addresses that correspond to that fully qualified domain name (hostname). It uses these IP addresses to match the destination IP addresses for a rule using the host label. You can also specify a set of IP subnets in a host label to use as the destination IP addresses for a rule using the host label.

Host labels are compatible with autodiscover, passthrough, and fixed-target (not packet mode) in-path rules. Host labels aren't compatible with IPv6.

Host labels are optional.

## Streamline In-Path Rules: Domain Labels

- Used in in-path rules
- Useful when a single server hosts many applications
- Also used for Web-Proxy in a single-ended deployment
- Order is important
  - Best practice is to put these rules at the end
- Limitations and caveats:
  - Max of 63 domain labels
  - Max of 64 characters
  - Default ports are 80 and 443, HTTP and HTTPS respectively
    - (Looks into the HTTP GET for the HOST field and SNI field of HTTPS)
  - IPv4 only
  - Wildcards “\*” are allowed, e.g. \*twitface.\*
  - Not compatible with SaaS

The screenshot shows the 'Domain Labels' configuration page in the Riverbed NetworkMiner dashboard. At the top, there's a breadcrumb trail: 'App Definitions > Domain Labels'. Below this, there are two tabs: 'Add a New Domain Label' (active) and 'Remove Selected'. The 'Add a New Domain Label' form has a 'Name' field and a 'Domains' field. A note below the 'Domains' field states: 'Entries can be separated with commas, spaces, or newlines.' There is an 'Add New Domain Label' button. Below the form, there is a table with columns 'Label' and 'Domains'. The table is currently empty, with a message 'No Domain Labels.' displayed. At the bottom, there is a link for 'Related Topics: In-Path Rules'.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 12

Domain labels are names given to a group of domains to streamline configuration. You can specify an internet domain with wildcards to define a wider group. For example, you can create a domain label called Office365 and add \*.microsoftonline.com, \*.office365.com, or \*.office.com.

Domain labels are optional, apply only to HTTP & HTTPS traffic, and require both client-side & server-side SteelHeads be running RiOS v9.2 or later.

### Use Cases

- match a specific set of services—domain labels can be especially useful when an IP address and subnet hosts many services and you don't need your in-path rule to match them all.
- replace a fixed IP address for a server—Some SaaS providers and the O365 VNext architecture that serve multiple O365 applications such as SharePoint, Lync, and Exchange no longer provide a fixed IP address for the server. With many IP addresses on the same server, a single address is no longer enough to match with an in-path rule. Let's suppose you need to select and optimize a specific SaaS service. Create a domain label and then use it with a host label and an in-path rule to intercept and optimize the traffic.

As of RiOS v9.9.2, domain labels are incompatible with IPv6, connection forwarding, and QoS.

## In-Path Rules: Optimization Options Overview

Other Options – the ‘if (optimize), how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

The screenshot displays a configuration form for In-Path Rules. The form includes the following fields and options:

- Preoptimization Policy:** A dropdown menu with 'None' selected.
- Latency Optimization Policy:** A dropdown menu with 'Normal' selected.
- Data Reduction Policy:** A dropdown menu with 'Normal' selected.
- Cloud Acceleration:** A dropdown menu with 'Auto' selected. A note next to it states: "Must be set to 'Pass Through' if a Domain Label (see above) is selected".
- Auto Kickoff:** An unchecked checkbox.
- Neural Framing Mode:** A dropdown menu with 'Always' selected.
- WAN Visibility Mode:** A dropdown menu with 'Correct Addressing' selected.
- Position:** A dropdown menu with 'End' selected.
- Description:** A text input field.
- Enable Rule:** A checked checkbox.
- Add:** A button at the bottom of the form.



## In-Path Rules: Preoptimization Policy

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

Preoptimization Policy: **None**

Latency Optimization Policy:

Data Reduction Policy:

Cloud Acceleration:

Auto Kickoff:

Neural Framing Mode:

WAN Visibility Mode:

Position:

Description:

Enable Rule: ☒

**Add**

*Domain Label (see above) is selected*

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 14

Select a traffic type from the drop-down list:

- None - If the Oracle Forms, SSL, or Oracle Forms-over-SSL preoptimization policy is enabled and you want to disable it for a port, select None. This is the default setting.
  - Note that Port 443 always uses a preoptimization policy of SSL even if an in-path rule on the client-side SteelHead sets the preoptimization policy to None. To disable the SSL preoptimization for traffic to port 443, you can either:
    - disable the SSL optimization on the client-side or server-side SteelHead.
  - or—
  - modify the peering rule on the server-side SteelHead by setting the SSL Capability control to No Check.
- Oracle Forms - Enables preoptimization processing for Oracle Forms. This policy is not compatible with IPv6.
- Oracle Forms over SSL - Enables preoptimization processing for both the Oracle Forms and SSL encrypted traffic through SSL secure ports on the client-side SteelHead. You must also set the Latency Optimization Policy to HTTP. This policy is not compatible with IPv6.
  - If the server is running over a standard secure port—for example, port 443—the Oracle Forms over SSL in-path rule needs to be *before* the default secure port pass-through rule in the in-path rule list.
- SSL - Enables preoptimization processing for SSL encrypted traffic through SSL secure ports on the client-side SteelHead.

## In-Path Rules: Latency Optimization Policy

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

Preoptimization Policy: None

Latency Optimization Policy: **Normal** (selected), HTTP, Outlook Anywhere, Citrix, Exchange Autodetect, None

Data Reduction Policy:

Cloud Acceleration:

Auto Kickoff:

Neural Framing Mode:

WAN Visibility Mode:

Position: End

Description:

Enable Rule: ☒

Add

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 15

Select one of these policies from the drop-down list:

- Normal - Performs all latency optimizations (HTTP is activated for ports 80 and 8080). This is the default setting.
- HTTP - Activates HTTP optimization on connections matching this rule.
- Outlook Anywhere - Activates RPC over HTTP(S) optimization for Outlook Anywhere on connections matching this rule. To automatically detect Outlook Anywhere or HTTP on a connection, select the Normal latency optimization policy and enable the Auto-Detect Outlook Anywhere Connections option in the Optimization > Protocols: MAPI page.
- Citrix - Activates Citrix-over-SSL optimization on connections matching this rule. This policy is not compatible with IPv6. Add an in-path rule to the client-side SteelHead that specifies the Citrix Access Gateway IP address, select this latency optimization policy on both the client-side and server-side SteelHeads, and set the preoptimization policy to SSL (the preoptimization policy must be set to SSL).
- Exchange Autodetect - Automatically detects MAPI transport protocols (Autodiscover, Outlook Anywhere, and MAPI over HTTP) and HTTP traffic. For MAPI transport protocol optimization, enable SSL and install the SSL server certificate for the Exchange Server on the server-side SteelHead.
- None - Do not activate latency optimization on connections matching this rule. For Oracle Forms-over-SSL encrypted traffic, you must set the Latency Optimization Policy to HTTP.

Note: Setting the Latency Optimization Policy to None excludes *all* latency

optimizations, such as HTTP, MAPI, and SMB.

## In-Path Rules: Data Reduction Policy

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 16

Optionally, if the rule type is Auto-Discover or Fixed Target, you can configure these types of data reduction policies:

- Normal - Performs LZ compression and SDR.
- SDR-Only - Performs SDR; doesn't perform LZ compression.
- SDR-M - Performs data reduction entirely in memory, which prevents the SteelHead from reading and writing to and from the disk. Enabling this option can yield high LANside throughput because it eliminates all disk latency. This data reduction policy is useful for:
  - a very small amount of data: for example, interactive traffic.
  - point-to-point replication during off-peak hours when both the server-side and client-side SteelHeads are the same (or similar) size.
- Compression-Only - Performs LZ compression; doesn't perform SDR.
- None - Doesn't perform SDR or LZ compression.

To configure data reduction policies for the FTP data channel, define an in-path rule with the destination port 20 and set its data reduction policy. Setting QoS for port 20 on the client-side SteelHead affects passive FTP, while setting the QoS for port 20 on the serverside SteelHead affects active FTP.

To configure optimization policies for the MAPI data channel, define an in-path rule with the destination port 7830 and set its data reduction policy.

## In-Path Rules: Cloud Acceleration

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- **Cloud Acceleration**
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

The screenshot shows the configuration interface for Cloud Acceleration. It includes several drop-down menus and checkboxes. The 'Cloud Acceleration' drop-down is set to 'Auto', with a tooltip that says 'Pass Through if a Domain Label (see above) is selected'. Other settings include 'Preoptimization Policy' set to 'None', 'Latency Optimization Policy' set to 'Normal', 'Data Reduction Policy' set to 'Normal', 'Auto Kickoff' set to 'Pass Through', 'Neural Framing Mode' set to 'Always', 'WAN Visibility Mode' set to 'Correct Addressing', 'Position' set to 'End', and 'Enable Rule' checked. An 'Add' button is at the bottom.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 17

After you subscribe to a SaaS platform and enable it, ensure that cloud acceleration is ready and enabled. When cloud acceleration is enabled, connections to the subscribed SaaS platform are optimized by the SteelHead SaaS. You don't need to add an in-path rule unless you want to optimize specific users and exclude others.

Select one of these choices from the drop-down list:

- Auto - If the in-path rule matches, the connection is optimized by the SteelHead SaaS connection.
- Pass Through - If the in-path rule matches, the connection is not optimized by the SteelHead SaaS, but it follows the other rule parameters so that the connection might be optimized by this SteelHead with other SteelHeads in the network, or it might be passed through.

Domain labels and cloud acceleration are mutually exclusive. When using a domain label, the Management Console dims this control and sets it to Pass Through. You can set cloud acceleration to Auto when domain label is set to n/a.

Using host labels is not recommended for SteelHead SaaS traffic.

Note: This applies to the legacy cloud acceleration service and not the SaaS Accelerator.



## In-Path Rules: Auto Kickoff

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

The screenshot shows a configuration form for an In-Path Rule. The settings are as follows:

Setting	Value	Notes
Preoptimization Policy:	None	
Latency Optimization Policy:	Normal	
Data Reduction Policy:	Normal	
Cloud Acceleration:	Auto	Must be set to "Pass Through" if a Domain Label (see above) is selected
Auto Kickoff:	<input checked="" type="checkbox"/>	
Neural Framing Mode:	Always	
WAN Visibility Mode:	Correct Addressing	
Position:	End	
Description:		
Enable Rule:	<input checked="" type="checkbox"/>	

An "Add" button is located at the bottom left of the form.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 18

You use automatic kickoff primarily when you deploy SteelHeads in data protection environments. In data protection deployments, connections carrying the data replication traffic between the two storage arrays are often long-lived. This poses a problem if the connections are established as unoptimized or pass-through (for example, if the SteelHead is offline during connection setup), because the connections can remain unoptimized for a long time. Without the automatic kickoff on a SteelHead, you must manually intervene to reset the connections carrying data replication traffic on one of the storage arrays.

Although you can use automatic kickoff for any type of optimizable connection, the majority of connections for office applications—web, email, and so on—are comparatively short-lived and begin to be optimized after a brief period of time without any need for a reset.

When using the automatic kickoff feature, be aware of the following behaviors:

- Automatic kickoff does not have a timer. A preexisting connection that remains inactive for a period of time is reset as soon as there is packet flow and it matches an in-path rule that has auto kickoff enabled. After the connection has been reset, an internal flag is set to prevent further kick offs for the connection unless the optimization service is restarted.
- Take note when you enable automatic kickoff make sure you do not cause undesired behavior.

For example, in a design in which there is network asymmetry, if one or more SteelHead neighbors are configured and an in-path rule with automatic kickoff matches the connection, then the connection is kicked off even after detecting only one side of the handshake conversation.

## In-Path Rules: Auto Kickoff vs ‘General Service Kickoff’

### A Note on Kickoff vs Auto Kickoff

- The ‘Kickoff’ feature applies to all connections on a SteelHead
- “Auto Kickoff” applies to individual in-path rules on a SteelHead

VCX255-A / SteelHead VCX

#### General Service Settings Network Services > General Service Settings

##### In-Path Settings

- ☒ Enable In-Path Support
  - ☒ Reset Existing Client Connections on Start Up *(not recommended for production networks)*
  - ☐ Enable L4/PBR/WCCP/Interceptor Support
  - ☒ Enable Optimizations on Interface **inpath0\_0**

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 19

The Kickoff feature provides you with a simple way to ensure that unoptimized active TCP connections passing through the SteelHead can be reset. When a connection is reset, it tries to reestablish itself using the SYN, SYN-ACK, ACK handshake. The SteelHead uses its in-path rule table to determine if the connection should be optimized.

Connections can pass through a SteelHead unoptimized when they are set up and active before the SteelHead optimization service is running. By default, the SteelHead does not reset legacy connections and reports them as preexisting.

The main difference between the auto kickoff feature and the kickoff feature is that kickoff has a global setting that can affect all existing connections passing through a SteelHead. The global setting sends a reset to all connections, regardless of whether they need one. The global setting is not recommended for production networks, but you can use it in lab test scenarios. By default, this setting is not enabled. You can enable this setting in the Management Console or with the **in-path kickoff** command.

## In-Path Rules: Neural Framing Mode

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable rule

Preoptimization Policy: None

Latency Optimization Policy: Normal

Data Reduction Policy: Normal

Cloud Acceleration: Auto Must be set to "Pass Through" if a Domain Label (see above) is selected

Auto Kickoff: ☐

Neural Framing Mode: Never ✓ Always TCP Hints Dynamic

WAN Visibility Mode: ☐

Position:

Description:

Enable Rule: ☒

Add

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 20

The Nagle algorithm is a means of improving the efficiency of TCP/IP networks by reducing the number of packets that need to be sent over the network. It works by combining several small outgoing messages and sending them all at once. Neural framing enables the system to select the optimal packet framing boundaries for SDR and creates a set of heuristics to intelligently determine the optimal moment to flush TCP buffers. The system continuously evaluates these heuristics and uses the optimal heuristic to maximize the amount of buffered data transmitted in each flush, while minimizing the amount of idle time that the data sits in the buffer.

Select a neural framing setting:

- **Never** - Do not use the Nagle algorithm. All the data is immediately encoded without waiting for timers to fire or application buffers to fill past a specified threshold. Neural heuristics are computed in this mode but are not used. In general, this setting works well with time-sensitive and chatty or real-time traffic.
- **Always** - Use the Nagle algorithm. This is the default setting. All data is passed to the codec, which attempts to coalesce consume calls (if needed) to achieve better fingerprinting. A timer (6 ms) backs up the codec and causes leftover data to be consumed. Neural heuristics are computed in this mode but aren't used. This mode is not compatible with IPv6.
- **TCP Hints** - If data is received from a partial frame packet or a packet with the TCP PUSH flag set, the encoder encodes the data instead of immediately coalescing it. Neural heuristics are computed in this mode but aren't used. This mode is not compatible with IPv6.
- **Dynamic** - Dynamically adjust the Nagle parameters. In this option, the system discerns the optimum algorithm for a particular type of traffic and switches to the best algorithm based on traffic characteristic changes. This mode is not compatible with IPv6.

## In-Path Rules: WAN Visibility Mode

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- **WAN Visibility Mode**
- Position
- Enable/Disable Rule

Preoptimization Policy: None  
Latency Optimization Policy: Normal  
Data Reduction Policy: Normal  
Cloud Acceleration: Auto Must be set to "Pass Through" if a Domain Label (see above) is selected  
Auto Kickoff: ☐ Always  
Neural Framing Mode:   
WAN Visibility Mode:   
Position:   
Description:   
Enable Rule:   
**Add**

- ✓ Correct Addressing
- Port Transparency
- Full Transparency
- Full Transparency with Reset

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 21

Enables WAN visibility, which pertains to how packets traversing the WAN are addressed.

RiOS provides three types of WAN visibility: correct addressing, port transparency, and full address transparency.

You configure WAN visibility on the client-side SteelHead (where the connection is initiated).

## In-Path Rules: Enable/Disable Rule

Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

Preoptimization Policy:

Latency Optimization Policy:

Data Reduction Policy:

Cloud Acceleration:  Must be set to "Pass Through" if a Domain Label (see above) is selected

Auto Kickoff: ☐

Neural Framing Mode:

WAN Visibility Mode:

Position:

Description:

Enable Rule: ☒

Simply provides ability to enable or disable the in-path rule. This allows creation of complex rules in advance, and enabling them when desired.



## In-Path Rules: Rule Position

### Other Options – the ‘if so how’ part...

- Preoptimization Policy
- Latency Optimization Policy
- Data Reduction Policy
- Cloud Acceleration
- Auto Kickoff
- Neural Framing Mode
- WAN Visibility Mode
- Position
- Enable/Disable Rule

Preoptimization Policy: None  
Latency Optimization Policy: Normal  
Data Reduction Policy:  
Cloud Acceleration:  
Auto Kickoff:  
Neural Framing Mode:  
WAN Visibility Mode:  
Position: Start, 1, 2, 3, 4, 5, End (selected)  
Description:  
Enable Rule: ☒  
Add

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 23

Select Start, End, or a rule number from the drop-down list. SteelHeads evaluate rules in numerical order starting with rule 1.

If the conditions set in a rule match, then the rule is applied, and the system moves on to the next packet. If the conditions set in the rule don't match, the system consults the next rule. For example, if the conditions of rule 1 don't match, rule 2 is consulted. If rule 2 matches the conditions, it's applied, and no further rules are consulted.

In general, list rules in this order:

1. Deny
2. Discard
3. Pass-through
4. Fixed-Target
5. Auto-Discover

Place rules that use domain labels below others.

Note: The default rule, Auto-Discover, which optimizes all remaining traffic that has not been selected by another rule, can't be removed and is always listed last.

# In-Path Rules: To Move a Rule (1/5)

## Other Options – To Move a Rule

VCX255-A /SteelHead VCX

ip 10.1.30.25 • VCX (VCX255) (s86\_64) • 9.9.1 • uptime 2 weeks, 5 days • Mon 14:02 GMT +C

DASHBOARD NETWORKING OPTIMIZATION REPORTS ADMINIS

### In-Path Rules

Network Services > In-Path Rules ⓘ

Save to Disk

➕ Add a New In-Path Rule ➖ Remove Selected Rules ⓘ If Move Selected Rules...

Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cloud Acceleration	Kickoff	Rule Status	Email Notify	Ignore Detect
▶ 1	Auto Discover	All-IPv4:*	10.1.30.103/32-443	All	--	SSL	HTTP	Normal	Auto	No	Enabled	n/a	✖
▶ 2	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
▶ 3	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
▶ 4	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
▶ 5	Auto Discover	All-IPv4:*	10.1.30.104/32-443	All	--	SSL	Outlook Anywhere	Normal	Auto	No	Enabled	n/a	✖
default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal	Normal	Auto	No	Enabled	n/a	✖

Description: Default In-Path Rule

Related Topics: General Service Settings, Peering Rules, SSL Main Settings, Certificate Authorities, Advanced Settings, HTTP Configuration, MAPI, Domain Labels, Host Labels, Port Labels, Current Connections, Connection History

To here

This rule needs to be moved from here

# In-Path Rules: To Move a Rule (2/5)

## Other Options – To Move a Rule

VCX255-A /SteelHead VCX ip 10.1.30.25 • VCX (VCX255) (s86\_64) • 9.9.1 • uptime 2 weeks, 5 days • Mon 14:02 GMT +C

DASHBOARD NETWORKING OPTIMIZATION REPORTS ADMINIS

### In-Path Rules Network Services > In-Path Rules ⓘ

[Save to Disk](#)

<input type="checkbox"/>	Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cloud Acceleration	Kickoff	Rule Status	Email Notify	Ignore Detect
<input type="checkbox"/>	1	Auto Discover	All-IPv4:*	10.1.30.103/32-443	All	--	SSL	HTTP	Normal	Auto	No	Enabled	n/a	✖
<input type="checkbox"/>	2	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
<input type="checkbox"/>	3	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
<input type="checkbox"/>	4	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
<input checked="" type="checkbox"/>	5	Auto Discover	All-IPv4:*	10.1.30.104/32-443	All	--	SSL	Outlook Anywhere	Normal	Auto	No	Enabled	n/a	✖
<input type="checkbox"/>	default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal	Normal	Auto	No	Enabled	n/a	✖

Description: Default In-Path Rule

Related Topics: General Service Settings, Peering Rules, SSL Main Settings, Certificate Authorities, Advanced Settings, HTTP Configuration, MAPI, Domain Labels, Host Labels, Port Labels, Current Connections, Connection History

First highlight the rule with the tick

# In-Path Rules: To Move a Rule (3/5)

## Other Options – To Move a Rule

Click on Move  
Selected Rules

VCX255-A /SteelHead-VCX

ip 10.1.30.25 - VCX (VCX255) [s86\_64] - 9.9.1 - uptime 2 weeks, 5 days - Mon 14:02 GMT +C

DASHBOARD NETWORKING OPTIMIZATION REPORTS ADMINIS

### In-Path Rules

Network Services > In-Path Rules ⓘ

Save to Disk

⊕ Add a New In-Path Rule ⊖ Remove Selected Rules ⓘ Move Selected Rules...

Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cloud Acceleration	Kickoff	Rule Status	Email Notify	Ignore Detect
1	Auto Discover	All-IPv4:*	10.1.30.103/32-443	All	--	SSL	HTTP	Normal	Auto	No	Enabled	n/a	✖
2	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
3	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
4	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
5	Auto Discover	All-IPv4:*	10.1.30.104/32-443	All	--	SSL	Outlook Anywhere	Normal	Auto	No	Enabled	n/a	✖
default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal	Normal	Auto	No	Enabled	n/a	✖

Description: Default In-Path Rule

Related Topics: General Service Settings, Peering Rules, SSL Main Settings, Certificate Authorities, Advanced Settings, HTTP Configuration, MAPI, Domain Labels, Host Labels, Port Labels, Current Connections, Connection History

# In-Path Rules: To Move a Rule (4/5)

## Other Options – To Move a Rule

Choose the new position with the arrow

VCX255-A /SteelHead VCX

ip 10.1.30.25 • VCX (VCX255) (s86\_64) • 9.9.1 • uptime 2 weeks, 5 days • Mon 14:02 GMT +C

DASHBOARD NETWORKING OPTIMIZATION REPORTS ADMINIS

### In-Path Rules

Network Services > In-Path Rules ⓘ

Save to Disk

⊕ Add a New In-Path Rule ⊖ Remove Selected Rules ⓘ If Move Selected Rules...

Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cloud Acceleration	Kickoff	Rule Status	Email Notify	Ignore Detect
1	Auto Discover	All-IPv4:*	10.1.30.103/32-443	All	--	SSL	HTTP	Normal	Auto	No	Enabled	n/a	✖
2	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
3	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
4	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
5	Auto Discover	All-IPv4:*	10.1.30.104/32-443	All	--	SSL	Outlook Anywhere	Normal	Auto	No	Enabled	n/a	✖
default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal	Normal	Auto	No	Enabled	n/a	✖

Description: Default In-Path Rule

Related Topics: General Service Settings, Peering Rules, SSL Main Settings, Certificate Authorities, Advanced Settings, HTTP Configuration, MAPI, Domain Labels, Host Labels, Port Labels, Current Connections, Connection Hi



# In-Path Rules: To Move a Rule (5/5)

## Other Options – To Move a Rule

Moved.

☐ Add a New In-Path Rule
 ☐ Remove Selected Rules
 ☐ Move Selected Rules...

<input type="checkbox"/>	Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cloud Acceleration	Kickoff	Rule Status	Email Notify	Ignore Latency Detection
<input type="checkbox"/>	1	Auto Discover	All-IPv4:*	10.1.30.103/32:443	All	--	SSL	HTTP	Normal	Auto	No	Enabled	n/a	✖
<input type="checkbox"/>	2	Auto Discover	All-IPv4:*	10.1.30.103/32:443	All	--	SSL	HTTP	Normal	Auto	No	Enabled	n/a	✖
<input type="checkbox"/>	3	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
<input type="checkbox"/>	4	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
<input type="checkbox"/>	5	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--	--	Auto	--	Enabled	✖	n/a
<input type="checkbox"/>	default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal	Normal	Auto	No	Enabled	n/a	✖

Description: Default In-Path Rule

# In-Path Rules – Example 1: Implicit Optimize

## Examples

Pass Through this server

Optimize this server

Standard Pass Through rules

Optimize these servers

Implicit optimize everything

### In-Path Rules Network Services > In-Path Rules

Add a New In-Path Rule Remove Selected Rules If Move Selected Rules...								
<input type="checkbox"/>	Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy
<input type="checkbox"/>	1	Pass Through	All-IPv4:*	10.1.30.105/32:*	All	TCP	--	--
<input type="checkbox"/>	2	Auto Discover	All-IPv4:*	10.1.30.104/32:443	All	--	SSL	Outlook Anywhere
<input type="checkbox"/>	3	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--
<input type="checkbox"/>	4	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--
<input type="checkbox"/>	5	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--
<input type="checkbox"/>	6	Auto Discover	All-IPv4:*	10.1.41.16/28:*	All	--	SSL	HTTP
<input type="checkbox"/>	7	Auto Discover	All-IPv4:*	10.1.51.248/30:8443	All	--	SSL	HTTP
	default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal

# In-Path Rules – Example 2: Explicit Pass-Through Examples

Pass Through this server

Optimize this server

Standard Pass Through rules

Optimize these servers

Implicit Pass through everything

Pre-empted

## In-Path Rules Network Services > In-Path Rules ⓘ


⊕ Add a New In-Path Rule ⊖ Remove Selected Rules ⓘ Move Selected Rules...

<input type="checkbox"/>	Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy
<input type="checkbox"/>	▶ 1	Pass Through	All-IPv4:*	10.1.30.105/32:*	All	TCP	--	--
<input type="checkbox"/>	▶ 2	Auto Discover	All-IPv4:*	10.1.30.104/32:443	All	--	SSL	Outlook Anywhere
<input type="checkbox"/>	▶ 3	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--
<input type="checkbox"/>	▶ 4	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--
<input type="checkbox"/>	▶ 5	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--
<input type="checkbox"/>	▶ 6	Auto Discover	All-IPv4:*	10.1.41.16/28:*	All	--	SSL	HTTP
<input type="checkbox"/>	▶ 7	Auto Discover	All-IPv4:*	10.1.51.248/30:8443	All	--	SSL	HTTP
<input type="checkbox"/>	▶ 8	Pass Through	All-IP:*	All-IP:*	All	TCP	--	--
	default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal





# In-Path Rules – Hit Rate Report

Reports > Rule Statistics > In-Path Rule Statistics

Only available if configured on the in-path rule

In-Path Rule Statistics [Reports > In-Path Rule Statistics](#)  [Save to Disk](#) [Restart Services](#)

☐ Clear All Statistics [Submit](#)

Rule Id	Rule Summary	Description	Hit Count	Last Hit Time	Counter Clear Time	Creation Time	Logged In From	Created By	Clear Stats	Email Notify	Ignore Latency Detection
1	Type- Pass Through Src- All-IP Dst- All-IP	n/a	228989	03/31/20-01:11:28	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>		n/a
2	Type- Pass Through Src- All-IP Dst- All-IP	n/a	179	10/31/19-15:38:06	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>		n/a
3	Type- Pass Through Src- All-IP Dst- All-IP	n/a	7921	10/14/19-14:04:42	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>		n/a
default	Type- auto Src- All-IP Dst- All-IP	Default In-path rule	51532050	03/31/20-07:16:06	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>	n/a	

1 of 1

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 31

## In-Path Rules – Enabling Passthrough Email Notification

▼ Add a New In-Path Rule ✕ Remove Selected Rules ⬆⬆ Move Selected Rules...

Type: Auto Discover

Enable Email Notification: ☐

Ignore Latency Detection: ☐

Source: { Subnet: All IP (IPv4 + IPv6)

Note: email for pass through rules only

## In-Path Rules – Notification Only of Pass Through

▼ Add a New In-Path Rule ✕ Remove Selected Rules ⇅ Move Selected Rules...

Type: 

Auto Discover

Enable Email Notification: ☐

▼ Add a New In-Path Rule ✕ Remove Selected Rules ⇅ Move Selected Rules...

Type: 

Pass Through

Enable Email Notification: ☒



## OOB Splice TCP Connection

- Critical connection - required for optimization to occur
- Used for 'control information' for SteelHeads
- Automatically set up during optimized connection between any two SteelHeads:
  - Check **Reports > Optimization > Peers** report
  - Also Syslog, search for "oob"
- Consider QoS:
  - Available in the App Flow Engine list
  - Mark it, prioritize it, or both
  - Especially important in satellite environments



Name	IP Address	Model	Version	Licenses
No peers.				



App Flow Name	Description
Riverbed Control Traffic (Client)	Internal Riverbed control channel traffic generated by SteelHead appliances.
Riverbed Control Traffic (Server)	Internal Riverbed control channel traffic generated by SteelHead appliances.
CAAP	Microsoft Active Directory Security Account Manager

Enables out-of-band (OOB) connection destination transparency. The OOB connection is a single, unique TCP connection that is established by a pair of SteelHead appliances that are optimizing traffic. The pair of SteelHead appliances use this connection strictly to communicate internal information required by them to optimize traffic.



## OOB Splice TCP Connection – Transparency Options

- In some environments, it is necessary to make OOB connections use a form of network transparency; for example, if network is unable to route between the in-path IP addresses or VLANs of SteelHead appliances.
- CLI command *only*; with three options for OOB transparency:
  - **None** – Specify correct addressing (this is the default setting)
  - **Destination transparency** – Use if the client-side SteelHead appliance cannot establish the OOB connection to the server-side SteelHead appliance
  - **Full transparency** – Use if your network is unable to route between SteelHead appliance in-path IP addresses or in-path VLANs, or you do not want to see SteelHead appliance IP addresses used for the OOB connection
- Syntax: [no] in-path peering oobtransparency mode [none | destination | full] | [port <port>]

The three options for OOB transparency:

- **none (Default)** - Specify correct addressing. The OOB connection is established between the two SteelHead appliances, without any TCP/IP header manipulation.
- **destination** - Specify destination mode. In this mode, the OOB connection has the form C-SHip:C-SHport<->Sip:Sport, where C-SHip is the client-side SteelHead appliance IP address, C-SHport is an ephemeral port chosen by CSH, Sip is the server IP address, and Sport is the server port number. The Sip and Sport parameters are taken from the first connection optimized by the pair of SteelHead appliances.
- **full** - Specify full mode. In this mode, the OOB connection has the form Cip:CSHfixed<-> Sip:Sport, where Cip is the client IP address, C-SHfixed is a predetermined port chosen by the client-side SteelHead appliance, Sip is the server IP address, and Sport is the server port number. The Cip, Sip, and Sport parameters are taken from the first connection optimized by the pair of SteelHead appliances.

**Note:** If you use WAN visibility full address transparency, you have the following transparency options for the OOB connection: OOB connection destination transparency and OOB connection full transparency. You configure OOB transparent addressing on the client-side SteelHead appliance (where the connection is initiated). By default, the OOB connection uses correct addressing.

## Peering Rules – Handling Probed Packets

### How to Deal with Probes on LAN OR WAN Interfaces

Annotations in the screenshot:

- Action**: Points to the 'Rule Type' dropdown menu.
- Classification**: Points to the 'Source Subnet' and 'Destination Subnet' fields.
- Cloud Accelerator**: Points to the 'Cloud Accelerator' dropdown menu.
- Specific to SSL**: Points to the 'SSL Capability' dropdown menu.

Number	Type	Source	Destination	Port
1	Pass	All-IP	All-IP	All
Description: Default rule to passthrough connections destined to currently bypassed SSL client-server pairs				
2	Auto	All-IP	All-IP	44
Description: Default rule to auto-discover and attempt to optimize connections destined to port 443 as SSL				
default	Auto	All-IP	All-IP	All

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 37

Peering rules control SteelHead behavior when the appliance detects probe queries. Peering rules (displayed using the **show in-path peering rules** command) are an ordered list of fields that a SteelHead uses to match with incoming SYN packet fields (for example, source or destination subnet, IP address, or TCP port) and with the in-path IP address of the probing SteelHead. If more than one in-path interface exists on the probing SteelHead, apply one peering rule for each in-path interface. Peering rules are especially useful in complex networks.

Peering rule actions are as follows:

- **Pass** - The receiving SteelHead does not respond to the probing SteelHead, and it allows the SYN+probe packet to continue through the network.
- **Accept** - The receiving SteelHead responds to the probing SteelHead and becomes the remote-side SteelHead (that is, the peer SteelHead) for the optimized connection.
- **Auto** - If the receiving SteelHead is not using enhanced auto discovery, this rule has the same effect as the Accept peering rule action. If enhanced auto discovery is enabled, the SteelHead becomes the optimization peer only if it is the last SteelHead in the path to the server.

If a packet does not match any peering rule in the list, the default rule applies.

## Peering Rules – Default Peering Rules

- The two rules in the list by default are specific to SSL
- SSL Licensing information is carried over the OOB splice
- Catch all rule at the end – *Order is important!*

Add a New Peering Rule Remove Selected Rules If Move Selected Rules...								
<input type="checkbox"/> Number	Type	Source	Destination	Port	Peer	SSL	Cloud Acceleration	
<input type="checkbox"/> 1	Pass	All-IP	All-IP	All	All-IPv4	Incapable	Auto	
Description: Default rule to passthrough connections destined to currently bypassed SSL client-server pairs								
<input type="checkbox"/> 2	Auto	All-IP	All-IP	443	All-IPv4	Capable	Auto	
Description: Default rule to auto-discover and attempt to optimize connections destined to port 443 as SSL								
default	Auto	All-IP	All-IP	All	All-IPv4	No Check	Auto	

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 38

The default peering rules are adequate for typical network configurations, such as in-path configurations. However, you might need to add peering rules for complex network configurations. For details about deployment cases requiring peering rules, see the *SteelHead Deployment Guide*.

Note: We recommend using in-path rules to optimize SSL connections on destination ports other than the default port 443.

- Default peering rule number 1, with the SSL incapable flag, matches any SSL connection whose IP address and destination port appear in the list of bypassed clients and servers in the Networking > SSL: SSL Main Settings page. The bypassed list includes the IP addresses and port numbers of SSL servers that the SteelHead is bypassing because it couldn't match the common name of the server's certificate with one in its certificate pool. The list also includes servers and clients whose IP address and port combination have experienced an SSL handshake failure. For example, a handshake failure occurs when the SteelHead can't find the issuer of a server certificate on its list of trusted certificate authorities.

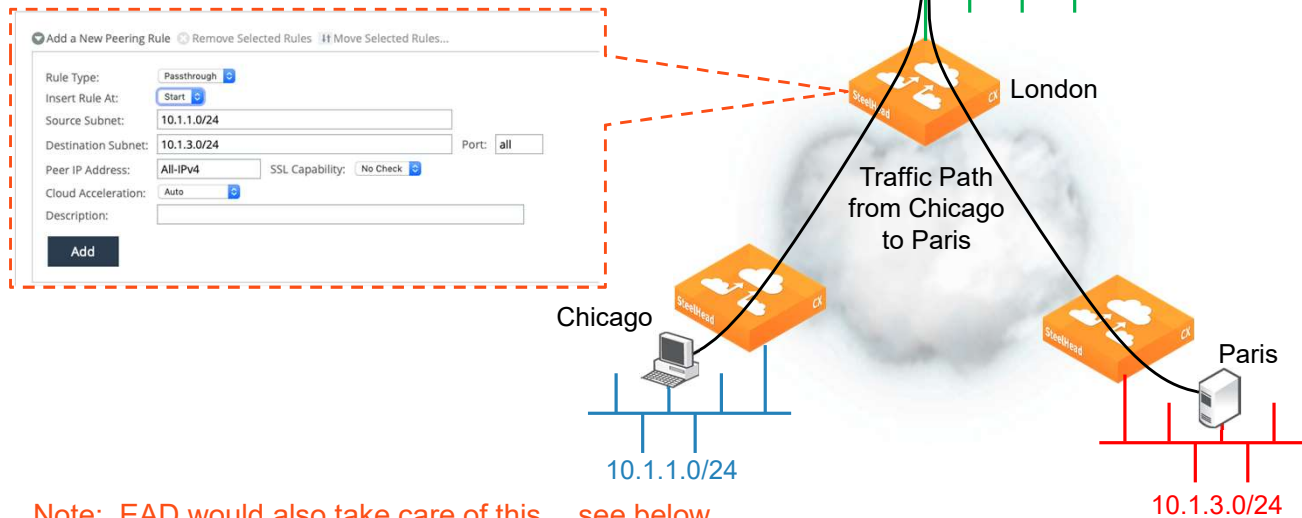
After a server or client appears in the bypassed servers list, follow-on connections to the same destination IP and port number always match rule number 1.

- The default peering rule number 2 with the SSL capable flag matches connections on port 443 that did not match default peering rule number 1. The SteelHead attempts to automatically discover certificate matches for servers answering on port 443. For all connections that match, the SteelHead performs both enhanced autodiscovery (finding the

nearest and farthest SteelHead pair) and SSL optimization.

## Peering Rule Example 1

### An Example



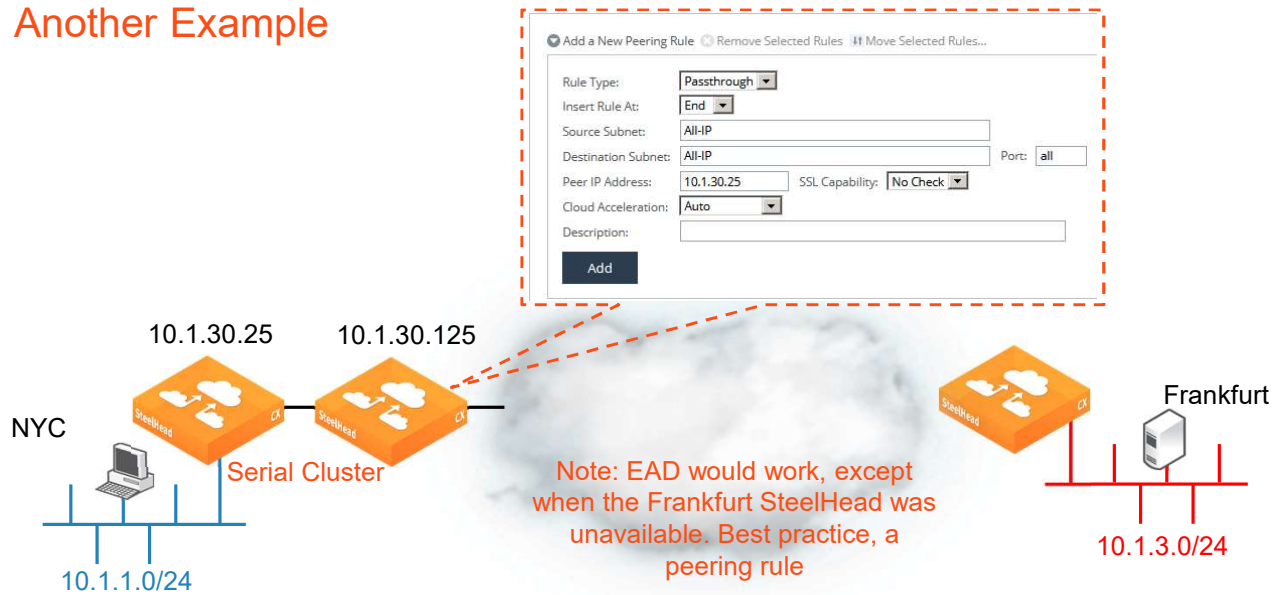
© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 39

This example is to show one possible use of peering rules. Do realize this example is fairly historical, because Enhanced Auto Discovery (EAD) would handle this without need for – and even better than – peering rules. For instance, if the SteelHead in Paris was unavailable or in admissions control, the peering rule would cause the entire connection to be passed through, instead of optimizing at least the Chicago-London link; with EAD, the first hop would be optimized, London to Paris would be passed through. It is a design choice.

## Peering Rule Example 2

### Another Example



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 40

## Peering Rules – Latency Detection

- When peer SteelHead appliances are geographically very close, such as might occur in full-mesh topologies, the network latency between them can be very low
- In cases where the latency between peers is low enough that simply passing traffic through would be faster than transmitting optimized traffic
- Default setting 10ms
  - If the latency between the peers is found to be less than this threshold (configurable), then pass through traffic
- Client-side SteelHead calculates the latency

Introduced in RiOS v9.9.0

You can use latency detection policies to globally manage how peer SteelHead appliances determine whether to pass through traffic or to continue to optimize it. You can still disable the feature on specific connections by setting an in-path rule and selecting the option to ignore latency detection.

The latency threshold is in milliseconds and the default is 10ms. If the latency between the peers is found to be less than the threshold latency (configurable), pass through traffic. The client-side SteelHead calculates the latency.

## Peering Rules – Latency Detection Configuration

VCX255-A / SteelHead VCX

ip 10.1.30.25 • VCX (VCX255L) (x86\_64) • 9.9.1 • uptime 2 weeks • Wed 12:57 GMT+0000 admin | Sign out

DASHBOARD NETWORKING OPTIMIZATION REPORTS ADMINISTRATION HELP

Peering Rules Network Services > Peering Rules ⓘ

Save to Disk Restart Services

Peering rules allow you to define appliance peering relationships. Note that only the first matching rule will be applied.

**Settings**

- ☒ Enable Enhanced IPv4 Auto-Discovery
- ☐ Enable Enhanced IPv6 Auto-Discovery
- ☐ Enable Extended Peer Table ⓘ
- ☒ **Enable Latency Detection**
  - Latency Threshold Value:  Milliseconds

Apply

⊕ Add a New Peering Rule ⊖ Remove Selected Rules ⓘ Move Selected Rules...

Number	Type	Source	Destination	Port	Peer	SSL	Cloud Acceleration
1	Pass	All-IP	All-IP	All	All-IPv4	Incapable	Auto

Description: Default rule to passthrough connections destined to currently bypassed SSL client-server pairs

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 42

**Latency Detection Policy in SCC:** Use latency detection policies to manage how peer SteelHead appliances determine whether to pass through traffic or to continue to optimize it when latency between the peers is low.

**Enable Latency Detection Peering Rule:** Enables peer appliances to pass through traffic without optimizing it when the latency between the peers is below the configured threshold. When enabled, you can specify the Ignore Latency Detection flag in peer in-path rules to disable the feature on specific rules as needed.

**Ignore Latency Detection In-Path Rule:** Indicates if the rule applies latency-detection pass-through.

**Appliance Details Report: Latency Detected Peers** report displays the connections that are optimized and the connections that are passed through.



## Peering Rule Latency Detection & In-Path Rules

In-Path Rules Network Services > In-Path Rules ?

▼ Add a New In-Path Rule ✕ Remove Selected Rules ⇅ Move Selected Rules...

Type: Auto Discover

Enable Email Notification: ☐

Ignore Latency Detection: ☐

Source: { Subnet: All IP (IPv4 + IPv6)

Destination: { Subnet: All IP (IPv4 + IPv6)  
Port: All Ports  
Domain Label: n/a

Only available when LD is globally configured from a SCC policy

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 43

**Latency Detection Policy in SCC:** Use latency detection policies to manage how peer SteelHead appliances determine whether to pass through traffic or to continue to optimize it when latency between the peers is low.

**Enable Latency Detection Peering Rule:** Enables peer appliances to pass through traffic without optimizing it when the latency between the peers is below the configured threshold. When enabled, you can specify the Ignore Latency Detection flag in peer in-path rules to disable the feature on specific rules as needed.

**Ignore Latency Detection In-Path Rule:** Indicates if the rule applies latency-detection pass-through.

**Appliance Details Report: Latency Detected Peers** report displays the connections that are optimized and the connections that are passed through.

## Latency Detection Support

### Policy, Rules, Reports

- Latency Detection Policy
- Enable Latency Detection Peering Rule
- Ignore Latency Detection In-Path Rule
- Appliance Details Report

In-Path Rule Statistics [Reports > In-Path Rule Statistics](#) [Save to Disk](#) [Report Settings](#)

☐ Clear All Statistics

[Subnets](#)

Rule ID	Rule Summary	Description	Hit Count	Last Hit Time	Counter Clear Time	Creation Time	Logged In From	Created By	Clear Stats	Email Notify	Ignore Latency Detection
1	Type-Pass Through Src: All IP Dst: All IP	n/a	0	Never	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Type-Pass Through Src: All IP Dst: All IP	n/a	0	Never	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Type-Pass Through Src: All IP Dst: All IP	n/a	0	Never	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
default	Type-Auto Src: All IP Dst: All IP	Default In-path rule	10	12/11/18-01:07:57	Never	n/a	n/a	n/a	<a href="#">Clear Stats</a>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

### Latency Detected Peers [Networking > Latency Detected Peers](#) [?](#)

[Save to Disk](#) [Restart Services](#)

Hostname / Peer IP	Latency	Optimized Connections	Passthrough Connections	Current Peer Status
<a href="#">192.168.3.2 / oak-vsh426</a>	54 ms	0	1	passthrough

**Latency Detection Policy in SCC:** Use latency detection policies to manage how peer SteelHead appliances determine whether to pass through traffic or to continue to optimize it when latency between the peers is low.

**Enable Latency Detection Peering Rule:** Enables peer appliances to pass through traffic without optimizing it when the latency between the peers is below the configured threshold. When enabled, you can specify the Ignore Latency Detection flag in peer in-path rules to disable the feature on specific rules as needed.

**Ignore Latency Detection In-Path Rule:** Indicates if the rule applies latency-detection pass-through.

**Appliance Details Report:** Latency Detected Peers report displays the connections that are optimized and the connections that are passed through.



## Optimize General UDP Traffic

### Packet Mode Optimization

- Most of the material covered so far could also apply to UDP traffic
- UDP traffic is usually interactive in nature and sensitive to variable delay (*jitter*) and packet loss
  - You would normally want to configure QoS
- Most UDP traffic is either voice or video
  - Not suitable for bandwidth streamlining
- Occasionally UDP is used to carry bulky traffic, e.g., TFTP
  - In these scenarios use *Packet Mode*

Most of the material we have covered here can be applied to UDP traffic. The important point to bear in mind is that, as UDP tends to transport interactive or real-time traffic, it is generally not suitable for standard SDR and tends to be more sensitive to variable delay or *jitter* and generally benefits from QoS.

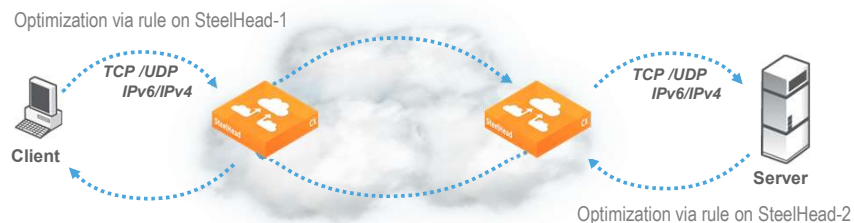
However, there are times when the UDP traffic is carrying a pay-load, such as TFTP. TFTP can be used for Cisco IOS images and even PXE boot images so there will be times when it is advantageous to optimize this traffic. We can do that using a SteelHead feature known as *Packet Mode Optimization*.

The key point with packet mode is that optimization is performed on a per-packet basis, which means that there is no round-trip reduction. This means that, from a performance perspective, the reduction in time for transferring a certain amount of data will not be as great as if we were using standard optimization. The key benefit here is bandwidth reduction, specifically when we are transferring large, bulky files using UDP.

Remember, do not use this for voice or video as will not give you any benefit.

## Packet Mode Overview

- Packet Mode initially designed for UDP and IPv6 optimization
  - No termination – flows not connections
  - Requires fixed-target rules
- Packet mode has several important aspects
  - No latency optimization: SDR + LZ only
  - Packet-by-packet inspection of data (what else can it be?)
  - Unidirectional, so rules needed on each client-side SH



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 47

In RiOS 8.5, support was added for TCP-level optimization through SDR and latency optimization for these protocols using IPv6: FTP, HTTP, and SSL. The integration and support includes IPv6 addressing for native IPv6 traffic, SNMP, network configuration of the in-path, and compliance with IPv6 RFCs.

Remember that UDP is connectionless, therefore no concept of SYNs and SYN/ACKSs.

## How to Configure Packet Mode

1. Globally enable packet mode.
  - Recall that Intercept would normally bypass the flow
  - Activated per flow using fixed-target in-path rules
  - Check-box on the GUI or via the command: **packet-mode enable**
  - This will require a service restart
  - This must be performed on both SteelHead appliances
2. Configure a fixed target rule for packet mode optimization on client-side SteelHead.
3. Click **Add**.
  - Repeat for SteelHead at the other site if bidirectional optimization is required.

The screenshot shows the configuration interface for a 'Fixed-Target (Packet Mode Optimization)' rule. The 'Type' is set to 'Fixed-Target (Packet Mode Optimization)'. The 'Source' is configured with 'Subnet: All IPv4' and 'Port: All Ports'. The 'Destination' is configured with 'Subnet: IPv4' and 'Port: All Ports', with a specific IP address '10.124.101.12/32' entered in the subnet field. The 'VLAN Tag ID' is set to 'all'. The 'Protocol' is set to 'UDP'. The 'Target Appliance IP Address' is '10.32.9.211' with 'Port: 7800'. The 'Backup Appliance IP Address' is empty with 'Port: 7810'. The 'Data Reduction Policy' is set to 'Normal'. The 'Position' is set to 'End'. The 'Description' field is empty. The 'Enable Rule' checkbox is checked. An 'Add' button is at the bottom.

Packet mode optimization is configured using the in-path rules, with the particular type being *Fixed-Target (Packet Mode Optimization)*. This must also be done on both SteelHeads in order for the optimization to be bi-directional. When you think about it, as there is no SYN or SYN/ACK with UDP, there is no client- or server- side SteelHead, just a pair of SteelHeads. The steps for configuring packet mode optimization are shown here.

## Verify Packet Mode

### ■ Reports > Networking > Current Connections

	CT	Notes	Source:Port	Destination:Port	LAN kB	WAN kB	Reduction	Start Time	Application
▶	»»	📄	10.1.21.110:49270	10.1.31.130:445	331	261	21%	2016/11/16 03:38:12	
▶	»»	📄	10.1.31.130:445	10.1.21.110:49270	21,153	8,594	59%	2016/11/16 03:38:13	

Packet mode

### ■ When will you see connections?

- Packet mode not enabled: nothing in connection table
- Packet mode enabled: pass-through connections in table
- Fixed-target packet-mode rule; optimized connections in table

2	100%	All current connections
2	100%	Established
0	0%	RIOS
0	0%	RIOS + SCPS
0	0%	SCPS
0	0%	TCP proxy
2	100%	Packet-mode optimized
0	0%	Establishing
0	0%	Opening
0	0%	Closing
0	0%	Forwarded
0	0%	Passthrough (unoptimized)
0	0%	Failed terminated
0	0%	Failed packet-mode
0	0%	Intentional
0	0%	Errors

## Packet Mode Reporting - CLI

- “show flows” CLI command

- “show flows” is superset of “show connections”
- Use “show flows” instead of “show connections” when packet-mode is enabled

```
chief-sh193 (config) # show flows ?  
all                show all flow types  
packet-mode        show packet-mode optimized flows only  
tcp-term           show list of terminated connections
```

- Can get more detail

```
chief-sh193 (config) #  
show flow srcip * srcport * dstip * dstport * [protocol *]
```



## Packet Mode – Real World Reality Check

- Packet Mode Optimization will often not result in huge savings in transfer times
  - Why?
- Packet mode is applying bandwidth streamlining only!
- Round trips (or *turns*) are not reduced
- This **will** result in reduced bandwidth utilization but user experience will be mixed
- Watch out for high latency environments
  - WAN utilization may be low BUT
  - User experience will not improve that much
- Take care when deciding whether or not to use packet mode optimization
- Transparency, server-side out-of-path, connection forwarding and Interceptor deployments are not supported for packet mode
- Most UDP traffic is better off as pass-through with QoS applied

# Control Optimization with In-path & Peering Rules

In this lab, you will:

- Configure In-Path and Peering Rules to Control Optimization

Duration: **45 minutes**

## HOL1118



*eLab system: link and access details  
provided in your course confirmation email*

## Module Review

You should now be able to:

- Administer in-path rules.
- Control SteelHead peering.
- Optimize UDP traffic.

© 2020 Riverbed Technology, Inc. All rights reserved.

**riverbed**  
The Digital Performance Company