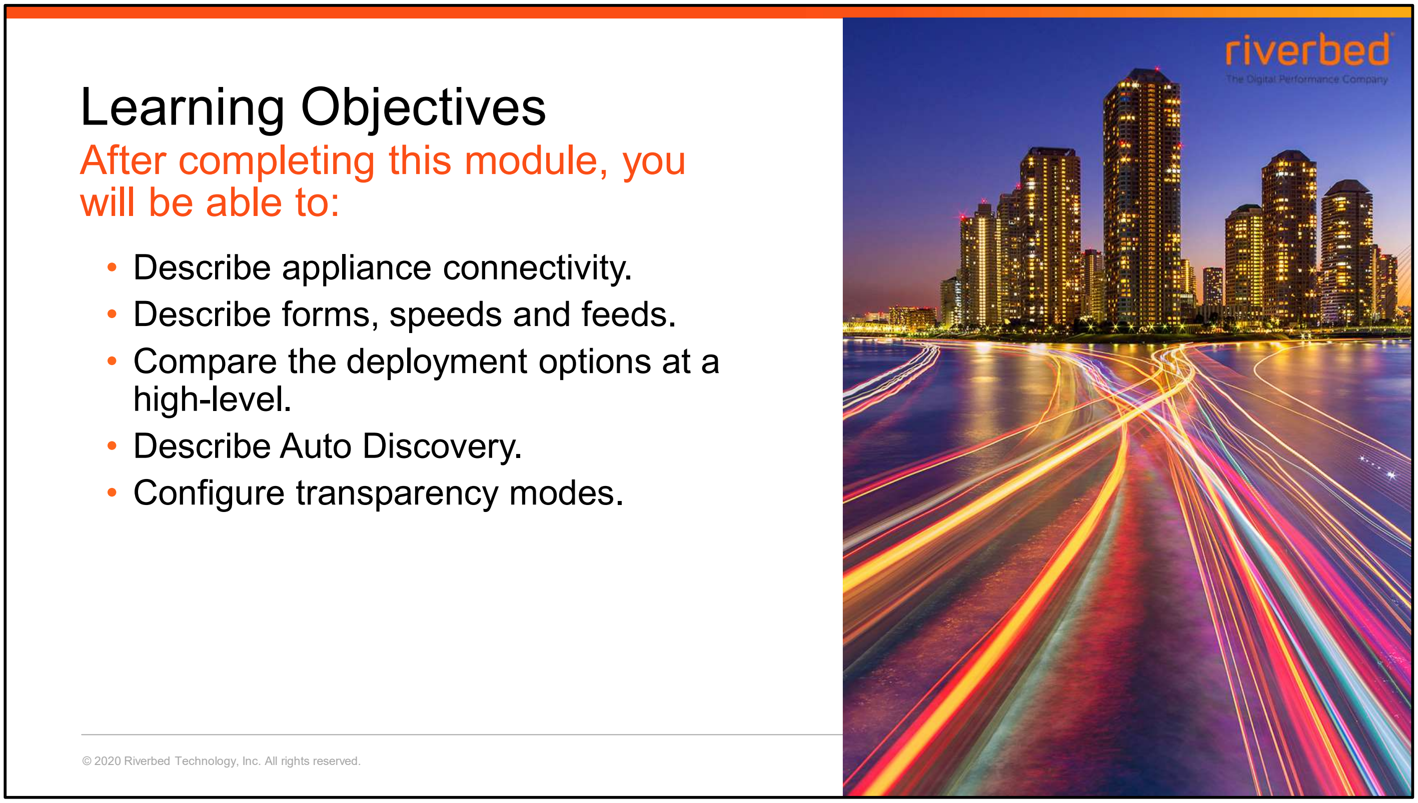
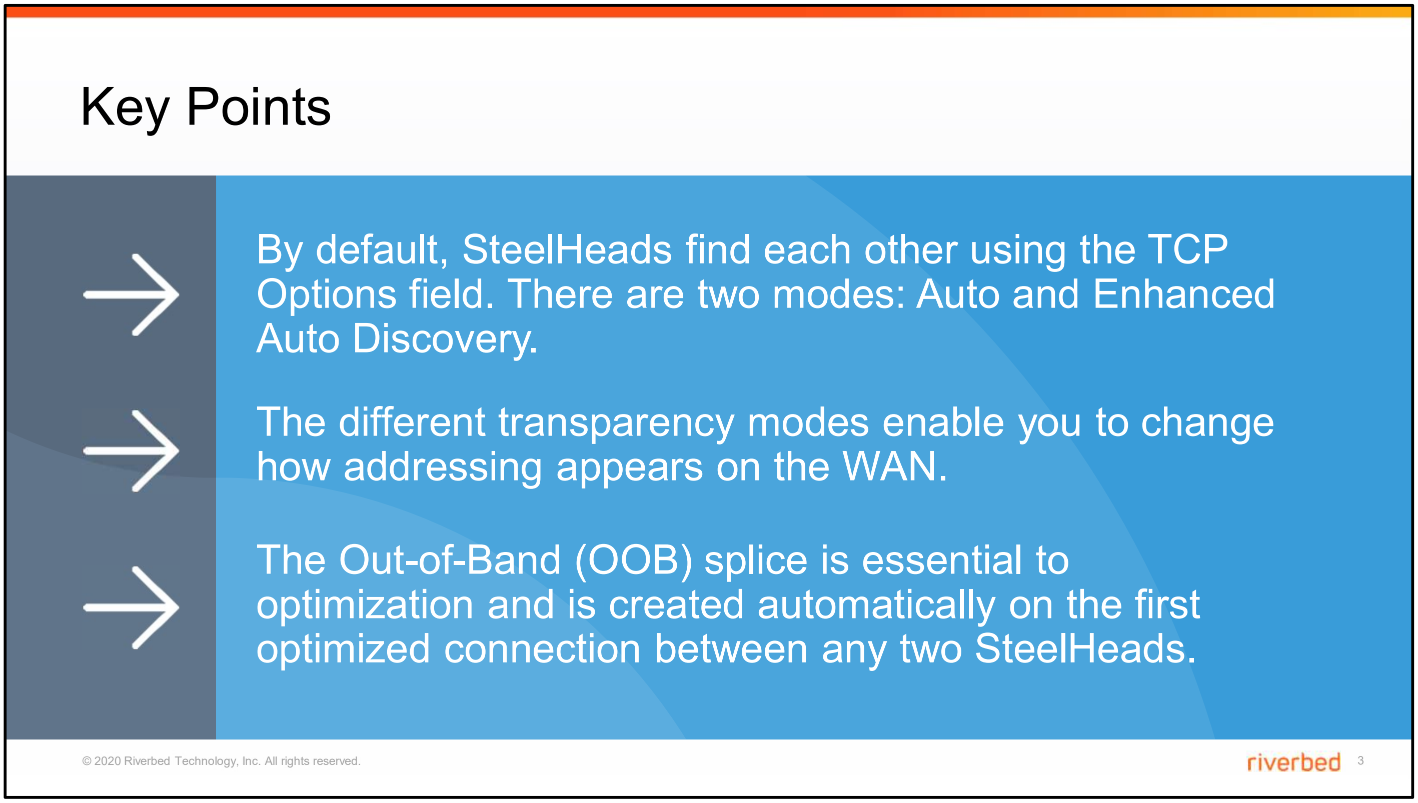
Describe the Solution Architecture



Learning Objectives

After completing this module, you will be able to:

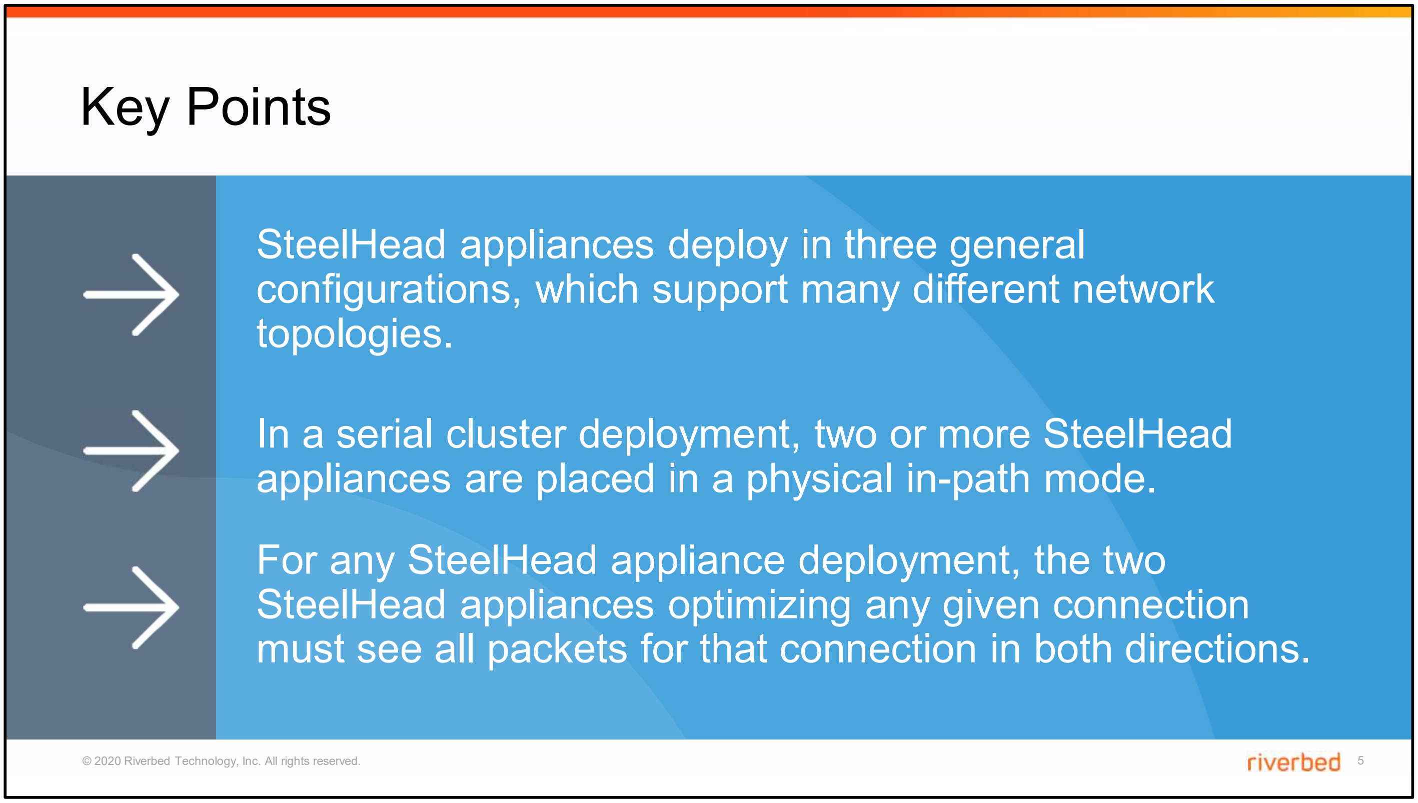
- Describe appliance connectivity.
- Describe forms, speeds and feeds.
- Compare the deployment options at a high-level.
- Describe Auto Discovery.
- Configure transparency modes.



Key Points

- By default, SteelHeads find each other using the TCP Options field. There are two modes: Auto and Enhanced Auto Discovery.
- The different transparency modes enable you to change how addressing appears on the WAN.
- The Out-of-Band (OOB) splice is essential to optimization and is created automatically on the first optimized connection between any two SteelHeads.

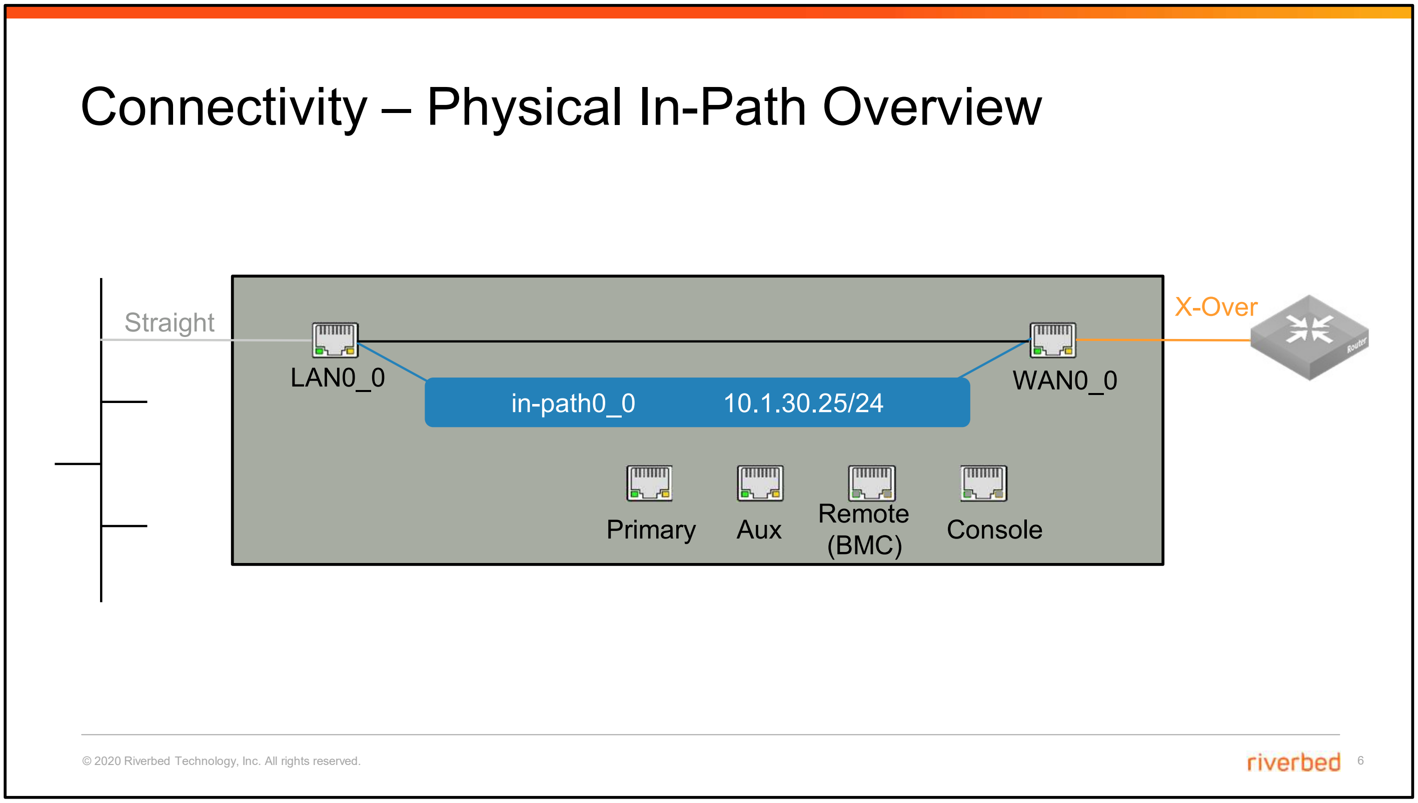
Describe Appliance Connectivity



Key Points

- SteelHead appliances deploy in three general configurations, which support many different network topologies.
- In a serial cluster deployment, two or more SteelHead appliances are placed in a physical in-path mode.
- For any SteelHead appliance deployment, the two SteelHead appliances optimizing any given connection must see all packets for that connection in both directions.

riverbed 5



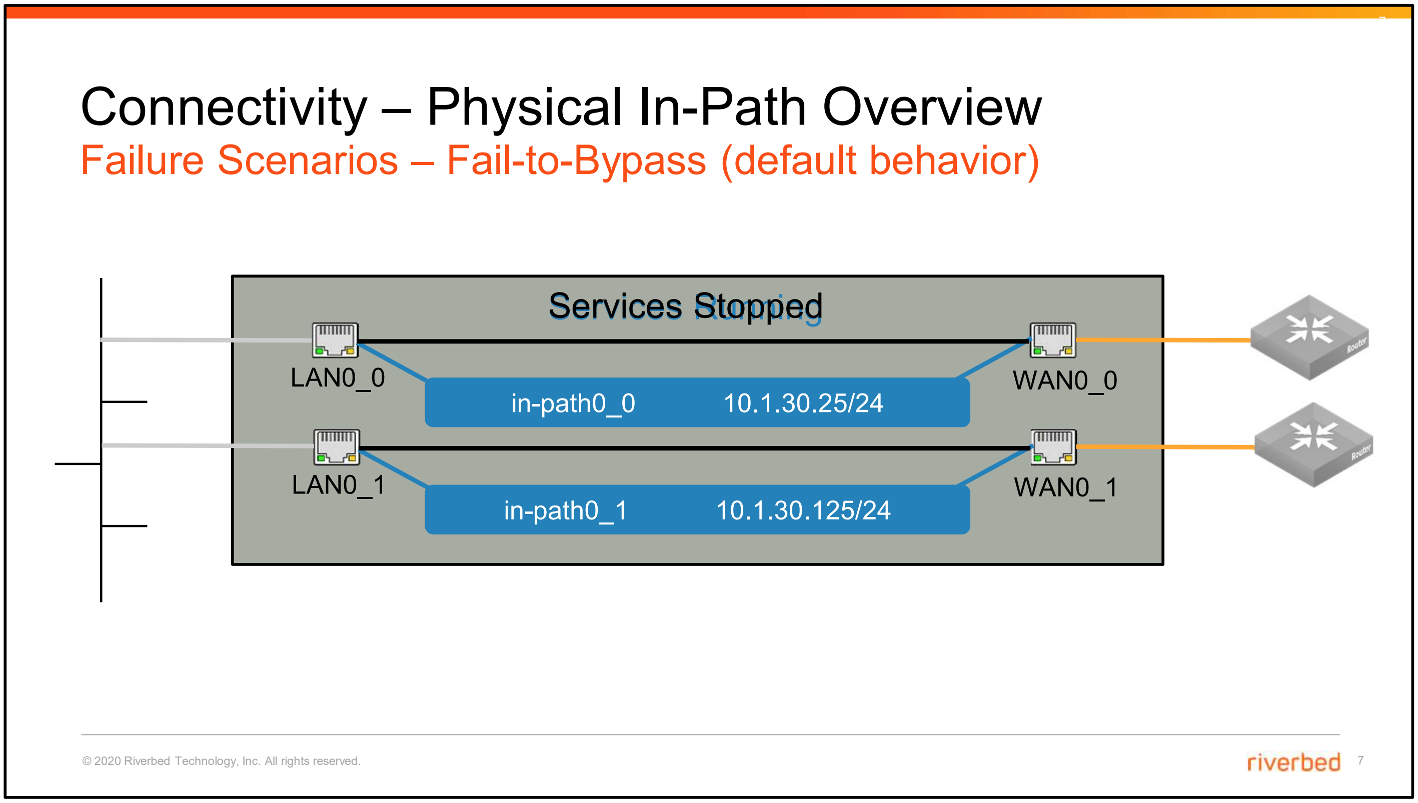
Connectivity – Physical In-Path Overview

riverbed 6

- **PRIMARY** – Used for the system management or OOP optimization
- **AUX** – truly Auxiliary; can be used for Management, VSP, segstore replication
- **INPATH0_0** – A logical in-path interface
  - Independent, two-port bridge, with its own IP addresses per WAN/LAN pair
- **LAN0_0** – Connects towards the LAN network, usually to a switch
- **WAN0_0** – Connects towards the WAN network, usually to a router
- **Console** – Local access to Command Line (CLI)
- **Remote** – Some models have remote port for out-of-band access

In-path interface – For each appliance, the Management Console detects LAN/WAN pairs, including those added through bypass cards, and identifies them according to slot (for example, inpath0_0, inpath0_1, inpath1_0, inpath1_1, and so forth). The logical in-path interface acts as an independent, two-port bridge, with its own IP address. All SteelHead appliance in-path interfaces have a “normally closed” relay, which means that when the appliance has no power or the in-path interface is disabled the associated LAN and WAN interfaces default to a fail-to-wire state. Fail-to-wire mode allows the SteelHead appliance WAN and LAN interfaces to serve in the same capacity as an Ethernet crossover cable. In fail-to-wire mode, SteelHead appliances cannot view or optimize traffic. Instead, all traffic is passed through the SteelHead appliance unoptimized.

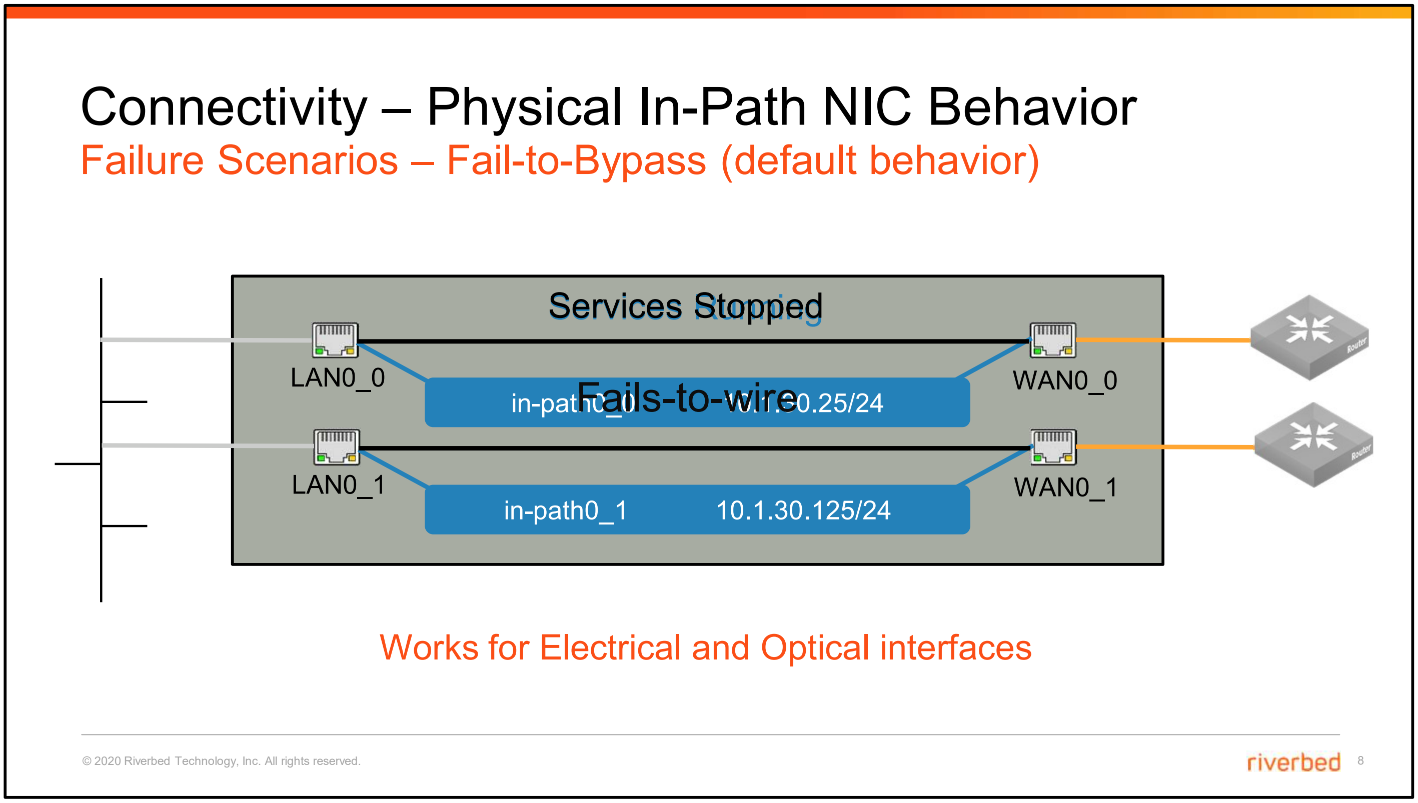
AUX interface – IP address must be on a subnet different from the Primary and In-path interface subnet(s).



Connectivity – Physical In-Path Overview

Failure Scenarios – Fail-to-Bypass (default behavior)

riverbed 7



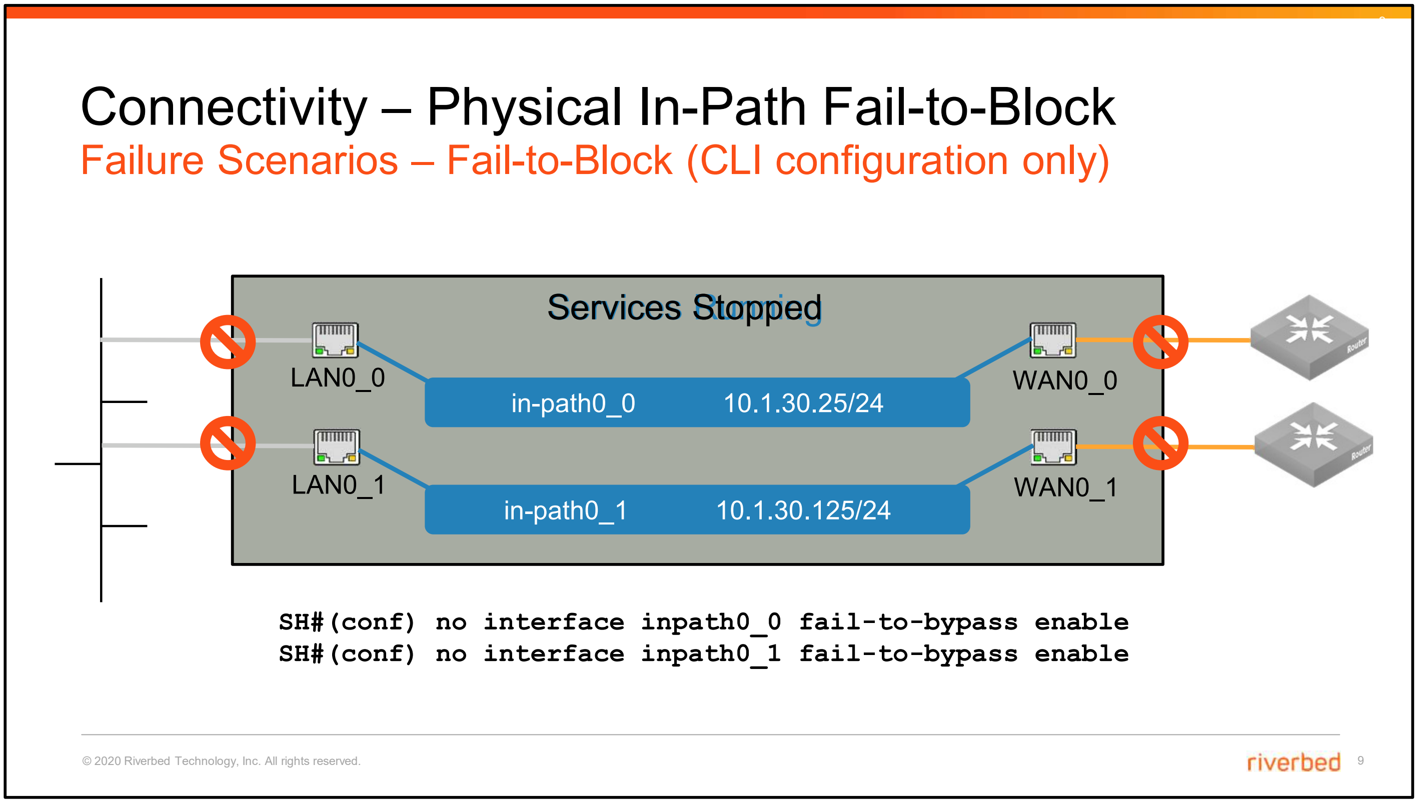
Connectivity – Physical In-Path NIC Behavior

Failure Scenarios – Fail-to-Bypass (default behavior)

Works for Electrical and Optical interfaces

If the SteelHead appliance is in or enters bypass mode, you are may be notified in the following ways:

- The Intercept/Bypass status light on the bypass card is triggered.
- The Home page of the Management Console displays Critical in the Status bar.
- SNMP traps are sent (if you have set this option).
- The event is logged to system logs (syslog).
- Email notifications are sent (if you have set this option).



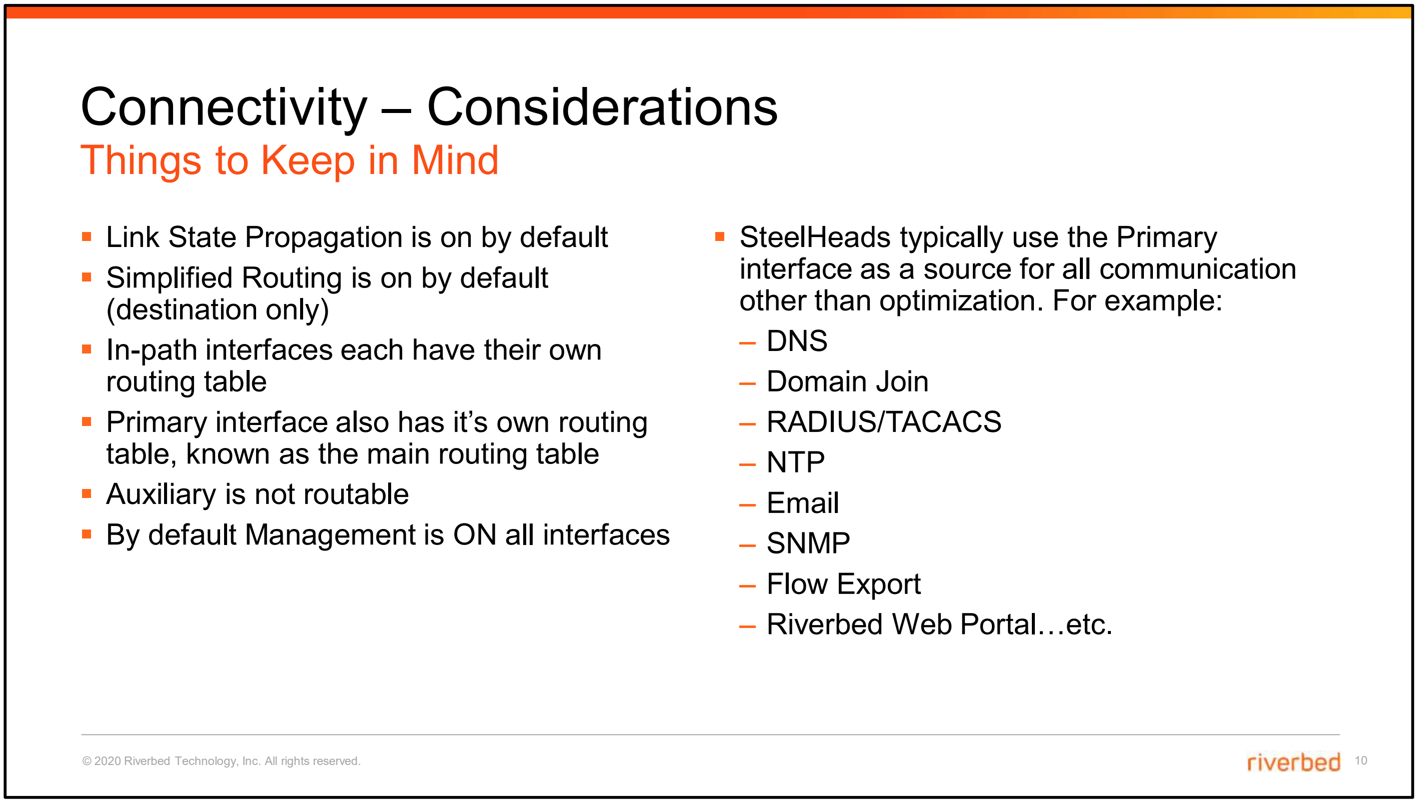
Connectivity – Physical In-Path Fail-to-Block

Failure Scenarios – Fail-to-Block (CLI configuration only)

```
SH#(conf) no interface inpath0_0 fail-to-bypass enable
SH#(conf) no interface inpath0_1 fail-to-bypass enable
```

riverbed 9

When fail-to-block is enabled, in the event of a failure or loss of power, the SteelHead LAN and WAN interfaces completely lose link status. The failed SteelHead blocks traffic along its path, with idea being attached Layer-3 devices will note interface failure and routing algorithms will cause traffic to be rerouted onto other paths (where ideally the remaining SteelHeads are deployed). For details on fail-to-block, see the *SteelHead Deployment Guide*.

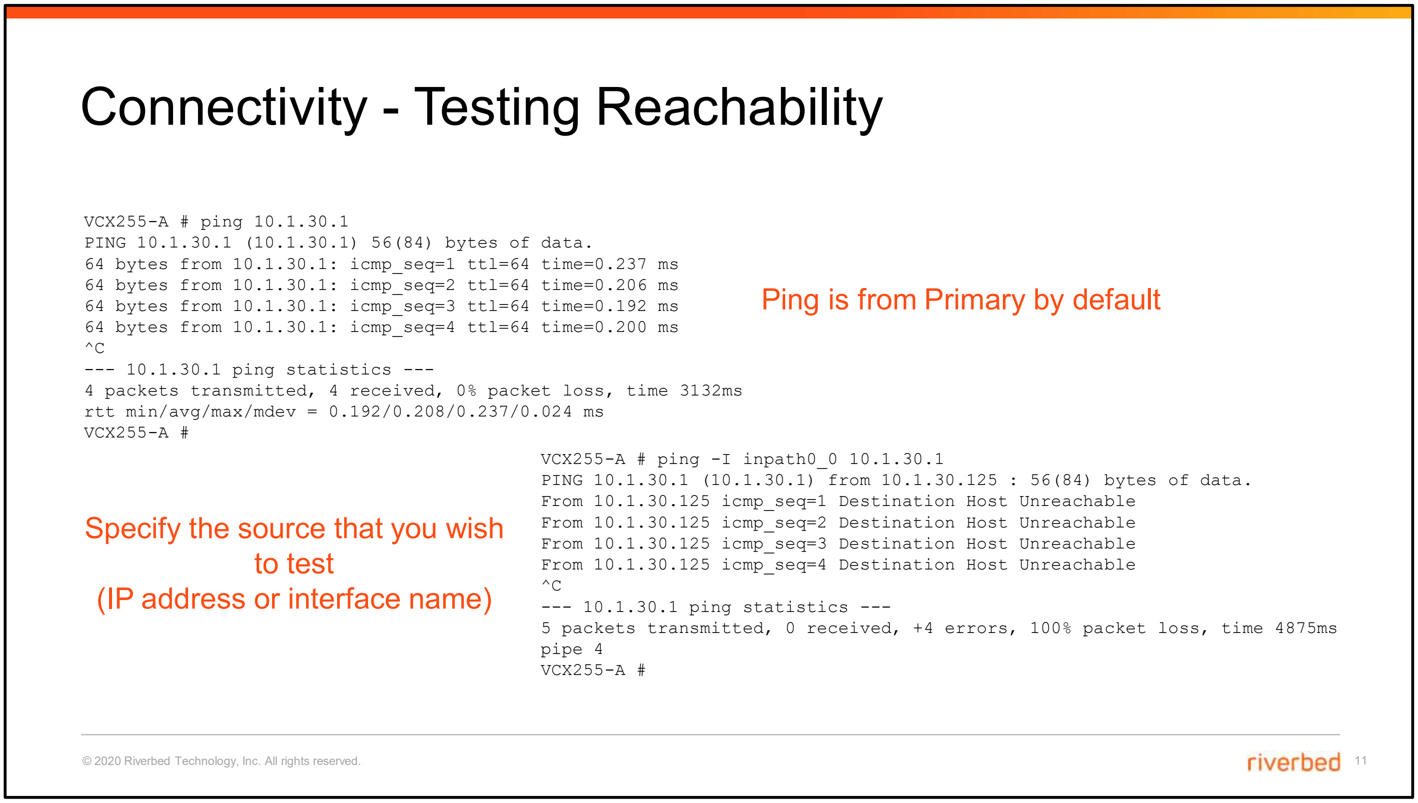


Connectivity – Considerations

Things to Keep in Mind

- Link State Propagation is on by default
- Simplified Routing is on by default (destination only)
- In-path interfaces each have their own routing table
- Primary interface also has it's own routing table, known as the main routing table
- Auxiliary is not routable
- By default Management is ON all interfaces

- SteelHeads typically use the Primary interface as a source for all communication other than optimization. For example:
  - DNS
  - Domain Join
  - RADIUS/TACACS
  - NTP
  - Email
  - SNMP
  - Flow Export
  - Riverbed Web Portal…etc.



Connectivity - Testing Reachability

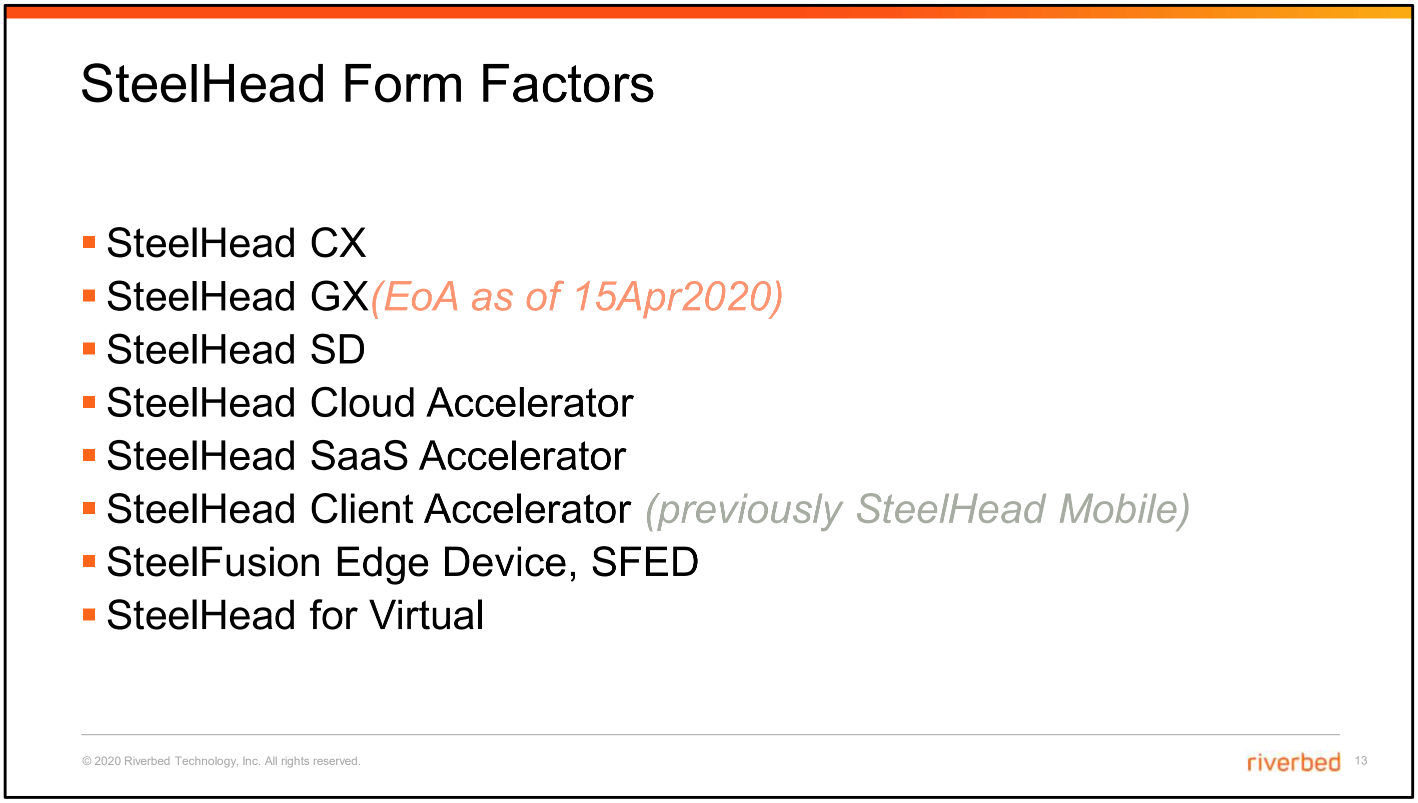
```
VCX255-A # ping 10.1.30.1
PING 10.1.30.1 (10.1.30.1) 56(84) bytes of data.
64 bytes from 10.1.30.1: icmp_seq=1 ttl=64 time=0.237 ms
64 bytes from 10.1.30.1: icmp_seq=2 ttl=64 time=0.206 ms
64 bytes from 10.1.30.1: icmp_seq=3 ttl=64 time=0.192 ms
64 bytes from 10.1.30.1: icmp_seq=4 ttl=64 time=0.200 ms
^C
--- 10.1.30.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3132ms
rtt min/avg/max/mdev = 0.192/0.208/0.237/0.024 ms
VCX255-A #
```

Ping is from Primary by default

```
VCX255-A # ping -I inpath0_0 10.1.30.1
PING 10.1.30.1 (10.1.30.1) from 10.1.30.125 : 56(84) bytes of data.
From 10.1.30.125 icmp_seq=1 Destination Host Unreachable
From 10.1.30.125 icmp_seq=2 Destination Host Unreachable
From 10.1.30.125 icmp_seq=3 Destination Host Unreachable
From 10.1.30.125 icmp_seq=4 Destination Host Unreachable
^C
--- 10.1.30.1 ping statistics ---
5 packets transmitted, 0 received, +4 errors, 100% packet loss, time 4875ms
pipe 4
VCX255-A #
```

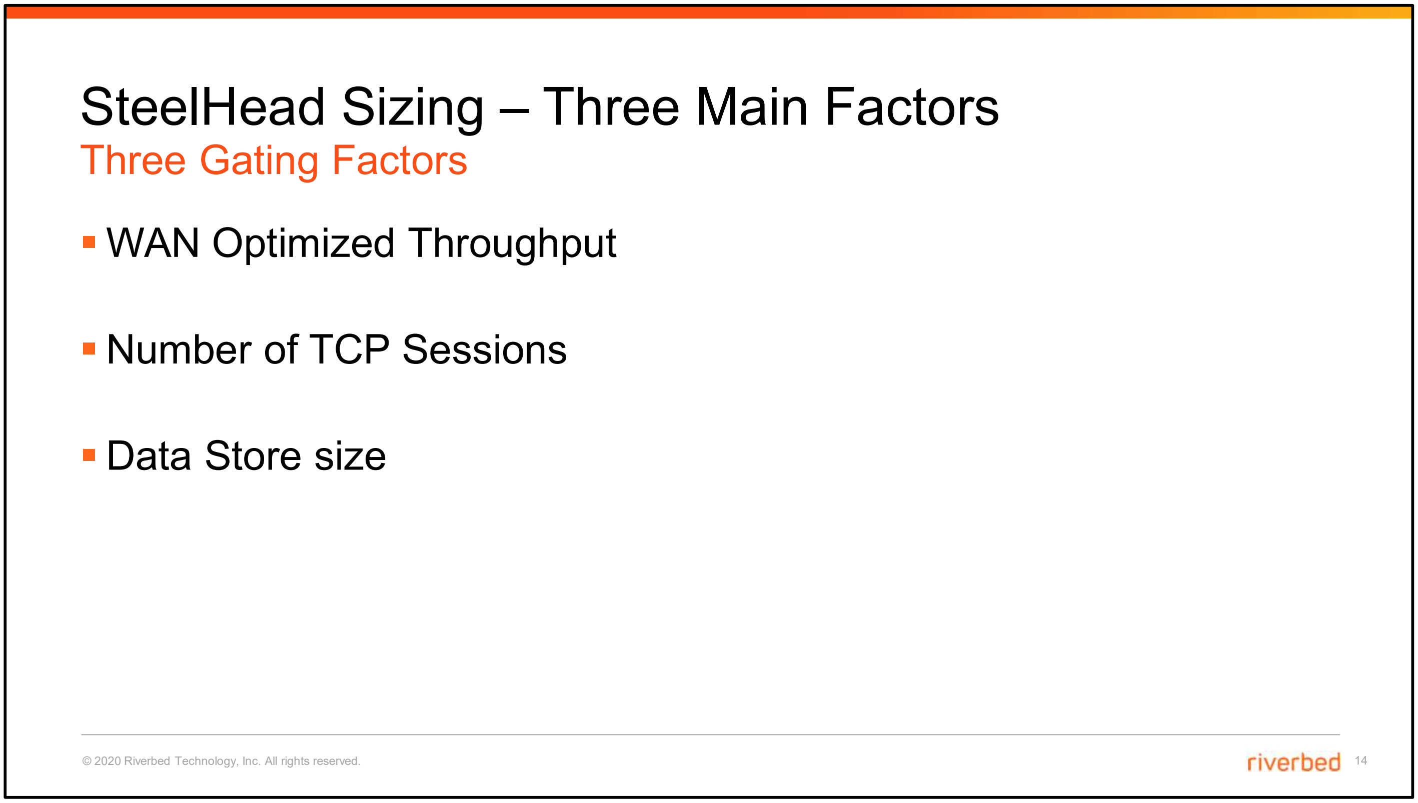
Specify the source that you wish to test
(IP address or interface name)

Describe Forms, Speeds and Feeds



SteelHead Form Factors

- SteelHead CX
- SteelHead GX*(EoA as of 15Apr2020)*
- SteelHead SD
- SteelHead Cloud Accelerator
- SteelHead SaaS Accelerator
- SteelHead Client Accelerator *(previously SteelHead Mobile)*
- SteelFusion Edge Device, SFED
- SteelHead for Virtual

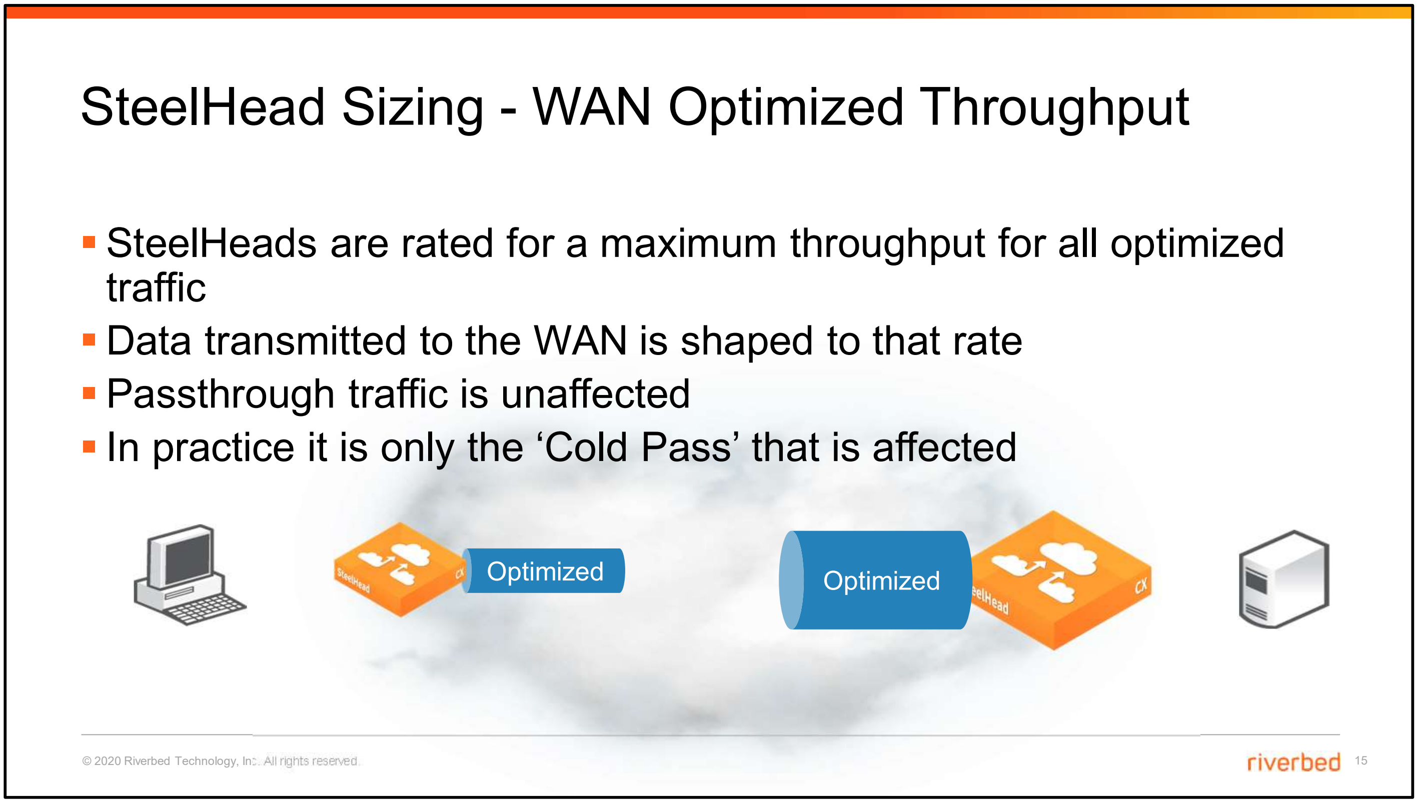


SteelHead Sizing – Three Main Factors

Three Gating Factors

- WAN Optimized Throughput
- Number of TCP Sessions
- Data Store size

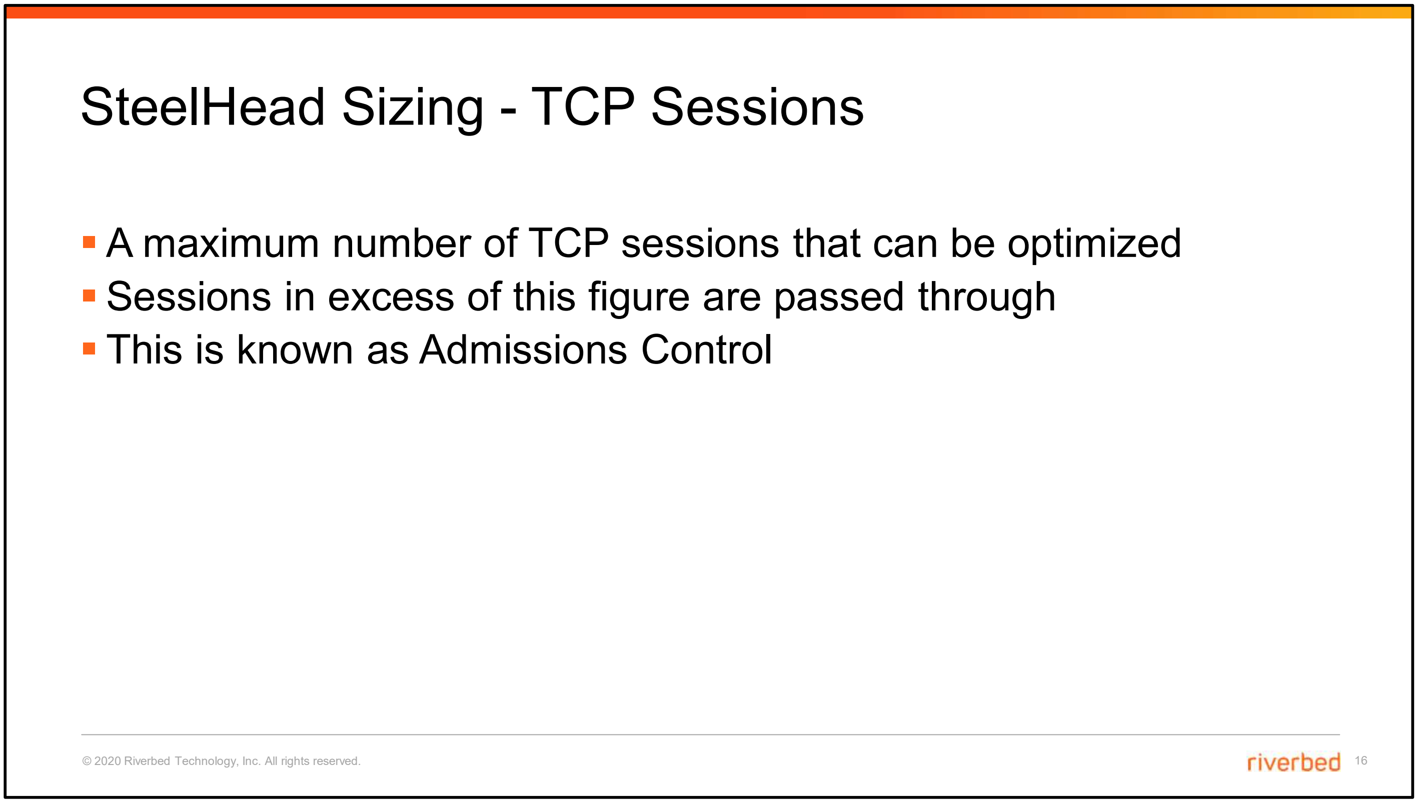
riverbed 14



SteelHead Sizing - WAN Optimized Throughput

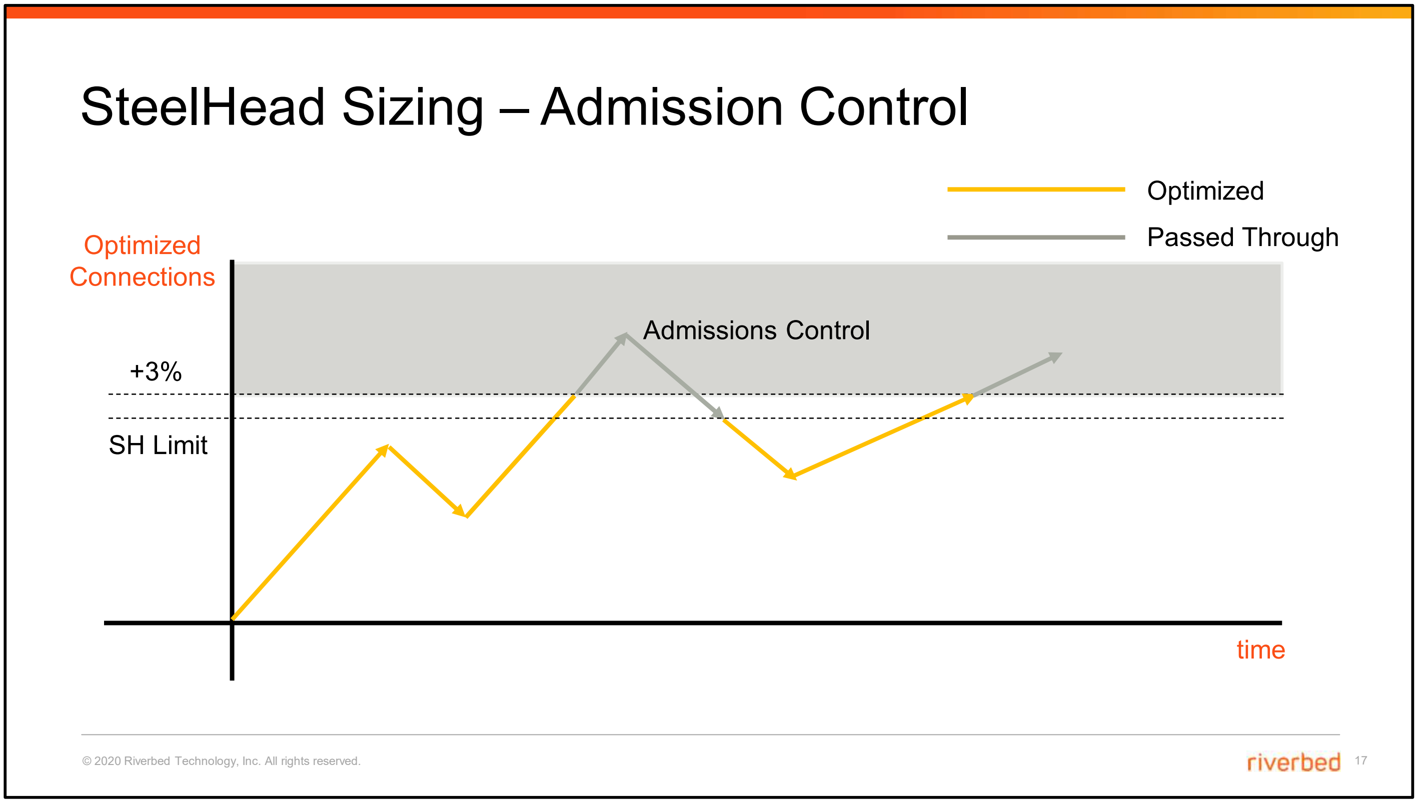
- SteelHeads are rated for a maximum throughput for all optimized traffic
- Data transmitted to the WAN is shaped to that rate
- Passthrough traffic is unaffected
- In practice it is only the ‘Cold Pass’ that is affected

riverbed 15



SteelHead Sizing - TCP Sessions

- A maximum number of TCP sessions that can be optimized
- Sessions in excess of this figure are passed through
- This is known as Admissions Control



SteelHead Sizing – Admission Control

riverbed 17

Admissions Control prevents the SteelHead from processing traffic when overloaded. It also controls the connection count limits. Admission Control stops the interception of connections for optimization but still allows the connections to pass through without optimization. Admissions Control is in one of two states:

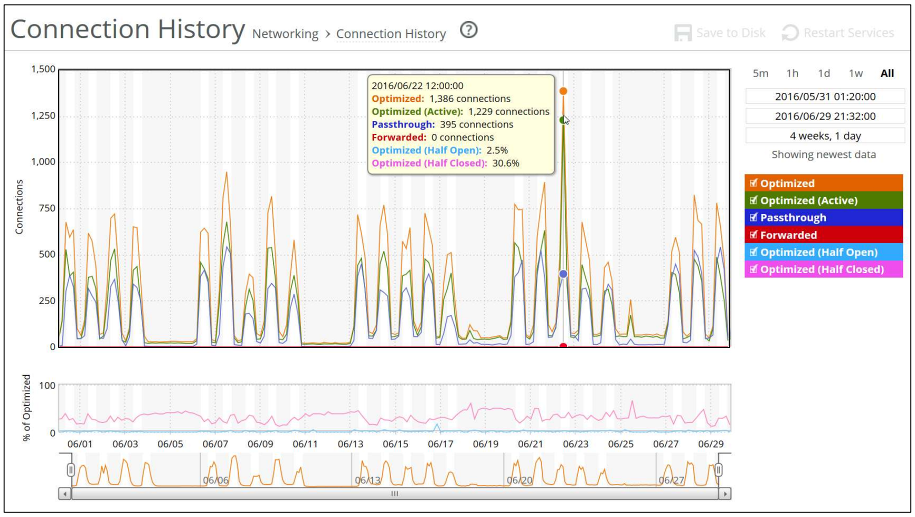
Flowing - In the flowing state, connections are intercepted as normal. Every 30 seconds or every 20 connections, admission control reevaluates whether the system is within limits. If the system exceeds certain limits, admission control moves into the paused state.

Paused - In the paused state, the SteelHead does not intercept connections. The connections currently intercepted continue to be optimized, although new connections are passed through. Every 30 seconds or every 20 connections, admission control reevaluates whether the system falls below certain limits. If so, admission control moves back into the flowing state.

Connection limits
Each model contains connection limits to limit the total number of connections that is accepted into the system. The connection limits have rising and falling thresholds. The rising threshold is the cutoff limit. While the system is in flowing state, if the connection count rises above this threshold, admission control moves to the paused state. The falling threshold is the enable limit. While the system is in the paused state, until the connection count falls below this threshold, admission control keeps the system in the paused state.

The SteelHead can also enter admission control due to memory consumption. If so, the SteelHead is optimizing traffic beyond its rated capability and is unable to handle the amount of traffic passing through the WAN link. During this event, the SteelHead continues to optimize existing connections, but new connections are passed through without optimization. The alarm clears automatically when the traffic has decreased and no other action is needed. This could be because the TCP window size has been manually expanded (See Bandwidth Delay Product).

SteelHead Sizing – Connection History Report

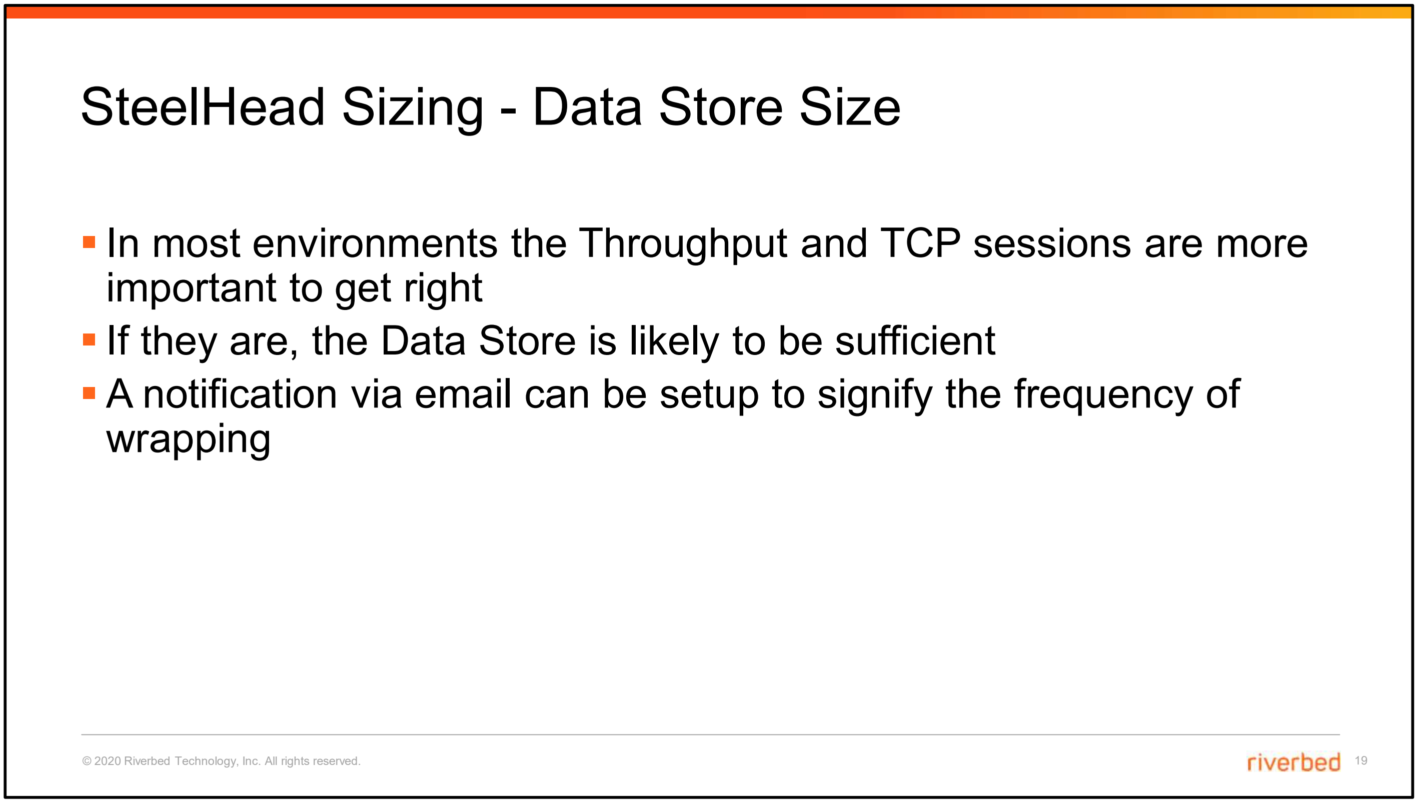


Connection History Networking › Connection History

2016/06/22 12:00:00
Optimized: 1,386 connections
Optimized (Active): 1,229 connections
Passthrough: 395 connections
Forwarded: 0 connections
Optimized (Half Open): 2.5%
Optimized (Half Closed): 30.6%

2016/05/31 01:20:00
2016/06/29 21:32:00
4 weeks, 1 day
Showing newest data

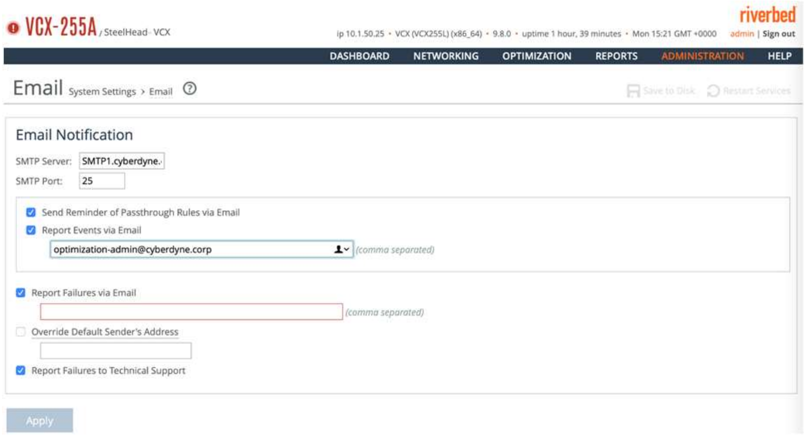
☑ Optimized
☑ Optimized (Active)
☑ Passthrough
☑ Forwarded
☑ Optimized (Half Open)
☑ Optimized (Half Closed)



SteelHead Sizing - Data Store Size

- In most environments the Throughput and TCP sessions are more important to get right
- If they are, the Data Store is likely to be sufficient
- A notification via email can be setup to signify the frequency of wrapping

Data Store Wrapping – Email Notification Requirements


VCX-255A / SteelHead- VCX
riverbed
ip 10.1.50.25 • VCX (VCX255L) (x86_64) • 9.8.0 • uptime 1 hour, 39 minutes • Mon 15:21 GMT +0000
admin | Sign out
DASHBOARD
NETWORKING
OPTIMIZATION
REPORTS
ADMINISTRATION
HELP
Email System Settings > Email
Save to Disk
Restart Services
Apply

Email Notification

SMTP Server: SMTP1.cyberdyne.

SMTP Port: 25

☑ Send Reminder of Passthrough Rules via Email

☑ Report Events via Email

optimization-admin@cyberdyne.corp (comma separated)

☑ Report Failures via Email

(comma separated)

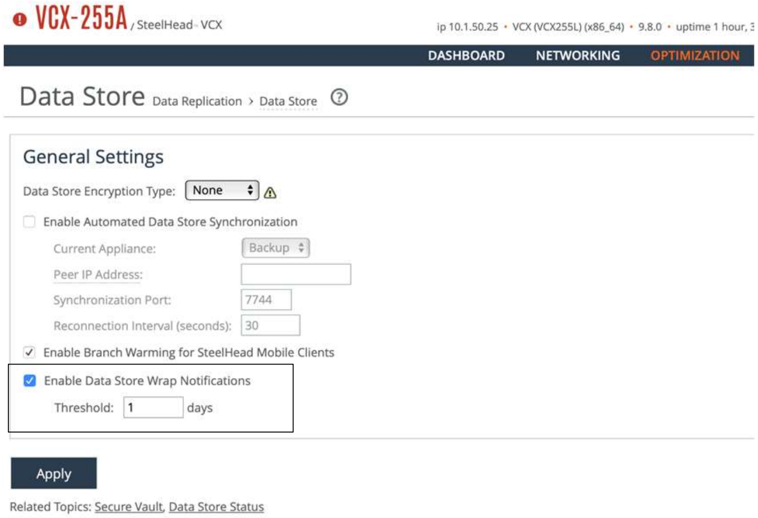
☐ Override Default Sender's Address

☑ Report Failures to Technical Support

Configure email settings for the SteelHeads under Administration

Data Store Wrapping – Enable System Notification

Notify About Wrapping



VCX-255A / SteelHead VCX
ip 10.1.50.25 • VCX (VCX255L) (x86_64) • 9.8.0 • uptime 1 hour, 3
DASHBOARD
NETWORKING
OPTIMIZATION
Data Store Data Replication › Data Store

General Settings

Data Store Encryption Type: None

□ Enable Automated Data Store Synchronization

Current Appliance: Backup

Peer IP Address:

Synchronization Port: 7744

Reconnection Interval (seconds): 30

☑ Enable Branch Warming for SteelHead Mobile Clients

☑ Enable Data Store Wrap Notifications

Threshold: 1 days

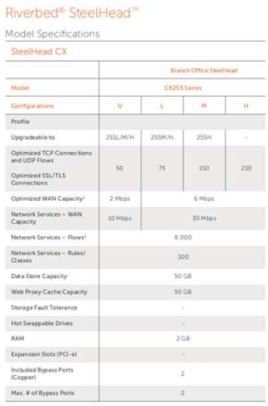
Related Topics: Secure Vault, Data Store Status

SteelHead Sizing – Physical Appliances

Options

Subject to change, check the website

Small



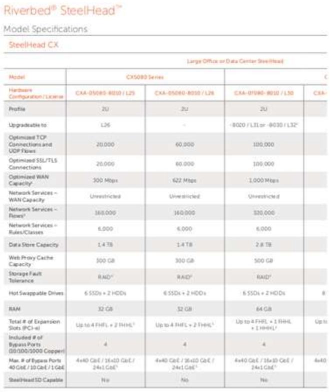
Riverbed® SteelHead™

Model Specifications

SteelHead CX

	Branch Office SteelHead			
Model	CX255 Series			
Configurations	U	L	M	H
Profile				
Upgradeable to	255L/M/H	255M/H	255H	-
Optimized TCP Connections and UDP Flows Optimized SSL/TLS Connections	50	75	150	230
Optimized WAN Capacity[1]	2 Mbps	6 Mbps		
Network Services – WAN Capacity	10 Mbps	30 Mbps		
Network Services – Flows[2]	8,000			
Network Services – Rules/Classes	300			
Data Store Capacity	50 GB			
Web Proxy Cache Capacity	50 GB			
Storage Fault Tolerance	-			
Hot Swappable Drives	-			
RAM	2 GB			
Expansion Slots (PCI-e)	-			
Included Bypass Ports (Copper)	2			
Max. # of Bypass Ports	2			

Medium



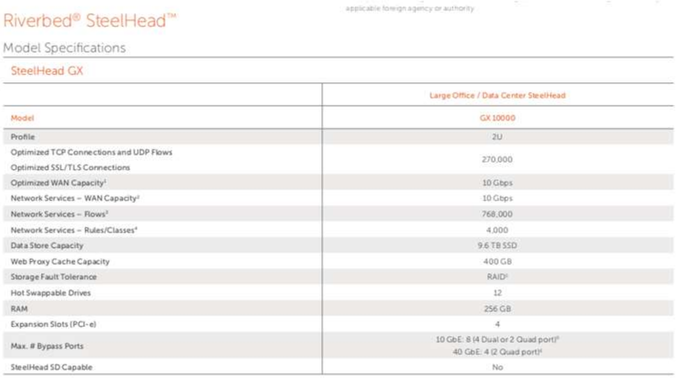
Riverbed® SteelHead™

Model Specifications

SteelHead CX

	Large Office or Data Center SteelHead			
Model	CX5080 Series			C
Hardware Configuration / License	CXA-05080-B010 / L25	CXA-05080-B010 / L26	CXA-07080-B010 / L30	CXA-
Profile	2U	2U	2U	
Upgradeable to	L26	-	-B020 / L31 or -B030 / L32[2]	
Optimized TCP Connections and UDP Flows	20,000	60,000	100,000	
Optimized SSL/TLS Connections	20,000	60,000	100,000	
Optimized WAN Capacity[1]	300 Mbps	622 Mbps	1,000 Mbps	
Network Services – WAN Capacity	Unrestricted	Unrestricted	Unrestricted	
Network Services – Flows[3]	160,000	160,000	320,000	
Network Services – Rules/Classes	6,000	6,000	6,000	
Data Store Capacity	1.4 TB	1.4 TB	2.8 TB	
Web Proxy Cache Capacity	300 GB	300 GB	500 GB	
Storage Fault Tolerance	RAID[4]	RAID[4]	RAID[4]	
Hot Swappable Drives	6 SSDs + 2 HDDs	6 SSDs + 2 HDDs	6 SSDs + 2 HDDs	8
RAM	32 GB	32 GB	64 GB	
Total # of Expansion Slots (PCI-e)	Up to 4 FHFL + 2 FHHL[5]	Up to 4 FHFL + 2 FHHL[5]	Up to 4 FHFL + 1 FHHL + 1 HHHL[5]	Up t
Included # of Bypass Ports (10/100/1000 Copper)	4	4	4	
Max. # of Bypass Ports 40 GbE / 10 GbE / 1 GbE	4x40 GbE / 16x10 GbE / 24x1 GbE[5]	4x40 GbE / 16x10 GbE / 24x1 GbE[5]	4x40 GbE / 16x10 GbE / 24x1 GbE[5]	4x40
SteelHead SD Capable	No	No	No	

Large



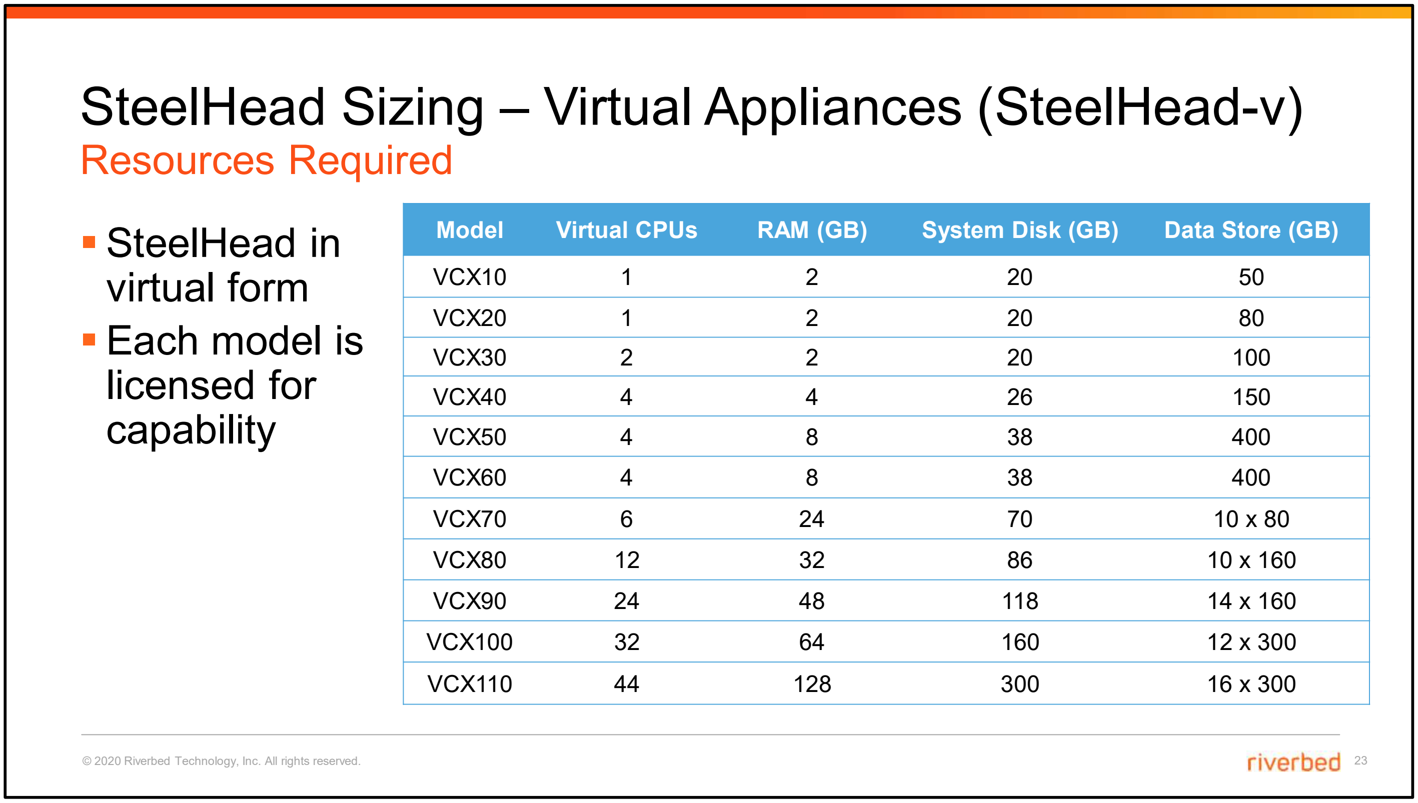
applicable foreign agency or authority

Riverbed® SteelHead™

Model Specifications

SteelHead GX

	Large Office / Data Center SteelHead
Model	GX 10000
Profile	2U
Optimized TCP Connections and UDP Flows Optimized SSL/TLS Connections	270,000
Optimized WAN Capacity[1]	10 Gbps
Network Services – WAN Capacity[2]	10 Gbps
Network Services – Flows[3]	768,000
Network Services – Rules/Classes[4]	4,000
Data Store Capacity	9.6 TB SSD
Web Proxy Cache Capacity	400 GB
Storage Fault Tolerance	RAID[5]
Hot Swappable Drives	12
RAM	256 GB
Expansion Slots (PCI-e)	4
Max. # Bypass Ports	10 GbE: 8 (4 Dual or 2 Quad port)[6] 40 GbE: 4 (2 Quad port)[6]
SteelHead SD Capable	No



SteelHead Sizing – Virtual Appliances (SteelHead-v)

Resources Required

- SteelHead in virtual form
- Each model is licensed for capability

Model	Virtual CPUs	RAM (GB)	System Disk (GB)	Data Store (GB)
VCX10	1	2	20	50
VCX20	1	2	20	80
VCX30	2	2	20	100
VCX40	4	4	26	150
VCX50	4	8	38	400
VCX60	4	8	38	400
VCX70	6	24	70	10 x 80
VCX80	12	32	86	10 x 160
VCX90	24	48	118	14 x 160
VCX100	32	64	160	12 x 300
VCX110	44	128	300	16 x 300

riverbed 23

SteelHead-v is software that delivers the benefits of WAN optimization, similar to those offered by the SteelHead hardware, while also providing the flexibility of virtualization.

Built on the same RiOS technology as the SteelHead, SteelHead-v reduces bandwidth utilization and speeds up application delivery and performance.

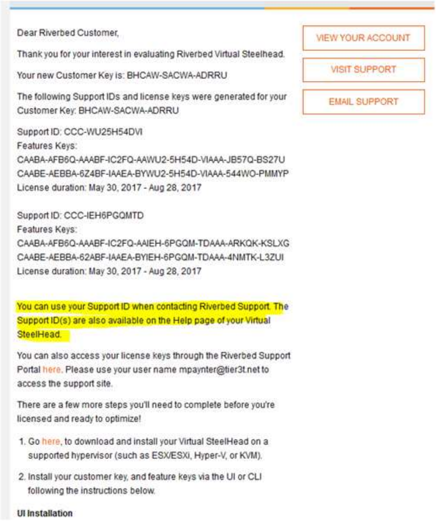
SteelHead-v on VMware vSphere is certified for the Cisco Service-Ready Engine (SRE) module with Cisco Services-Ready Engine Virtualization (Cisco SRE-V).

SteelHead-v runs on VMware ESXi, Microsoft Hyper-V, and Linux KVM hypervisors installed on industry-standard hardware servers.

SteelHead-v Licensing

Common License Management Framework

- Perpetual or Subscription Based
- Based on a Support ID



Dear Riverbed Customer,

Thank you for your interest in evaluating Riverbed Virtual Steelhead.

Your new Customer Key is: BHCAW-SACWA-ADRRU

The following Support IDs and license keys were generated for your Customer Key: BHCAW-SACWA-ADRRU

Support ID: CCC-WU25H54DVI
Features Keys:
CAABA-AFB6Q-AAABF-IC2FQ-AAWU2-5H54D-VIAAA-JB57Q-BS27U
CAABE-AEBBA-6Z4BF-IAAEA-BYWU2-5H54D-VIAAA-544WO-PMMYP
License duration: May 30, 2017 - Aug 28, 2017

Support ID: CCC-IEH6PGQMTD
Features Keys:
CAABA-AFB6Q-AAABF-IC2FQ-AAIEH-6PGQM-TDAAA-ARKQK-KSLXG
CAABE-AEBBA-62ABF-IAAEA-BYIEH-6PGQM-TDAAA-4NMTK-L3ZUI
License duration: May 30, 2017 - Aug 28, 2017

You can use your Support ID when contacting Riverbed Support. The Support ID(s) are also available on the Help page of your Virtual SteelHead.

You can also access your license keys through the Riverbed Support Portal here. Please use your user name mpaynter@tier3t.net to access the support site.

There are a few more steps you'll need to complete before you're licensed and ready to optimize!

1. Go here, to download and install your Virtual SteelHead on a supported hypervisor (such as ESX/ESXi, Hyper-V, or KVM).
2. Install your customer key, and feature keys via the UI or CLI following the instructions below.

UI Installation

CLI license add sequence needs to include the "clmf" option. e.g.

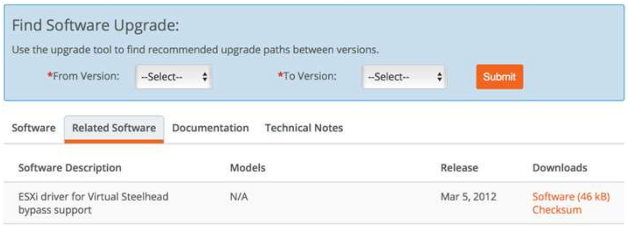
```
Sh # config term
sh (config) # license clmf install DAABB-AAHMQ-AAAEO-YAAAQ-AQCKF-BGIMT-DSAAA-3DXQO-4OXSU
sh (config) # write mem
sh (config) # end
sh #
```

This is a licensing framework that addresses both Hardware (HW) and Software (SW) product demands. Essentially, as Riverbed includes software products that have different licensing demands, such as floating licensing, where a license can be used across software instances rather than being locked or fixed to a hardware serial number, CLMF will be the framework that provides it.

SteelHead-v and Riverbed Bypass NIC

Riverbed Bypass NIC – A physical card for the virtual device

- ESX/ESXi
  - Direct Path feature
    - Allows SteelHead-v to control the bypass hardware
  - ESXi driver available from the support website Under related Software
- Hyper-V & KVM support



Find Software Upgrade:

Use the upgrade tool to find recommended upgrade paths between versions.

*From Version: --Select-- *To Version: --Select-- Submit

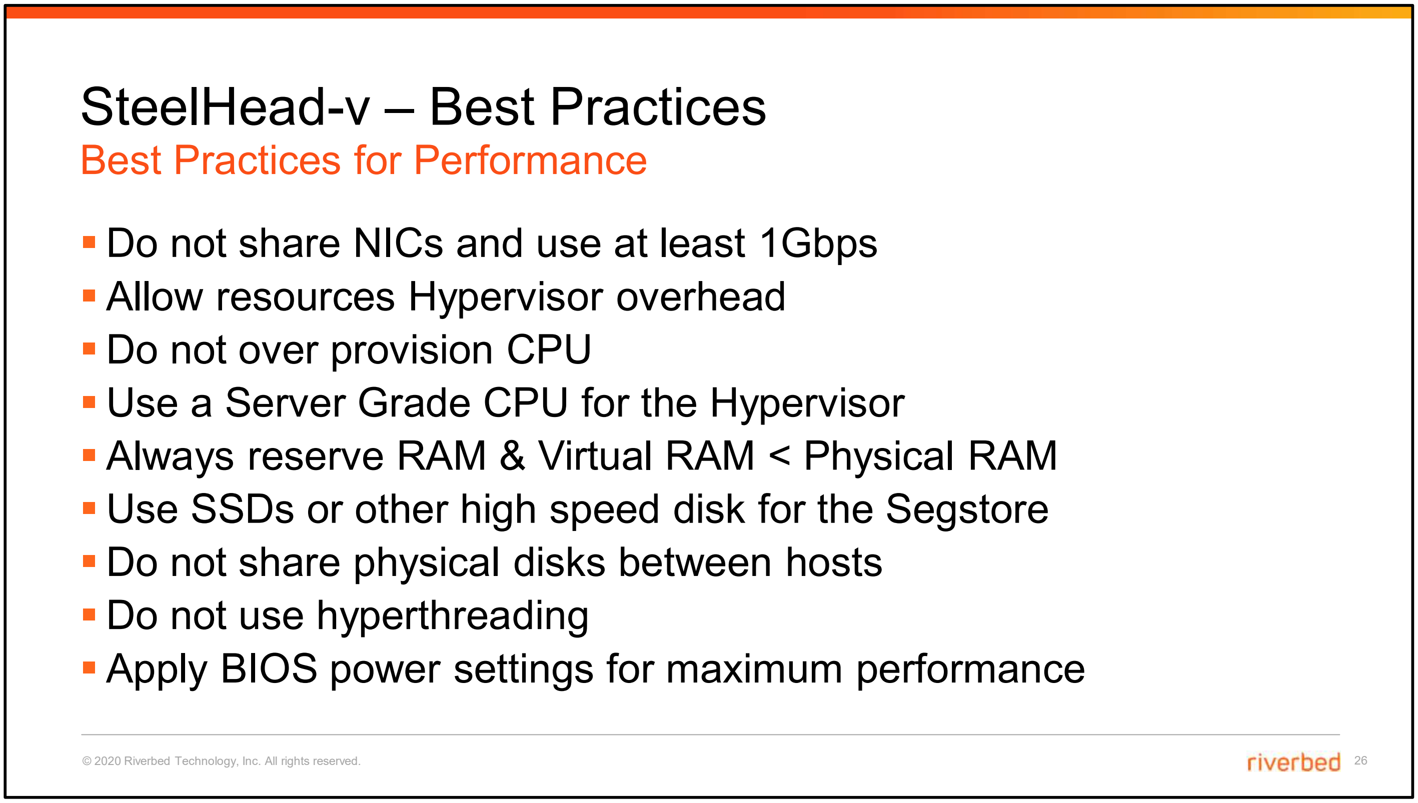
Software Description	Models	Release	Downloads
ESXi driver for Virtual Steelhead bypass support	N/A	Mar 5, 2012	Software (46 kB) Checksum

Bypass support (fail-to-wire) using the VMware Direct Path feature. This feature allows SteelHead-v to directly control the physical bypass card. The procedure for configuring bypass support is documented in the *Network and Storage Card Installation Guide*.

The in-path pair limit is four (four LAN and four WAN interfaces), including bypass cards. Each SteelHead-v requires a primary and auxiliary interface, which are the first two interfaces added. If you add additional interface pairs to the VM, they are added as in-path optimization interfaces. Total bandwidth and connection limits still apply.

Riverbed NICs provide hardware-based fail-to-wire and fail-to-block capabilities for SteelHead-v. The configured failure mode is triggered if the host loses power or is unable to run the SteelHead-v guest, if the SteelHead-v guest is powered off, or if the SteelHead-v guest experiences a significant fault (using the same logic as the physical SteelHead).

Some SteelHead-v models support a 2-port 10-GbE Multimode Fiber NIC (direct I/O only). You must use a Riverbed-branded NIC. Using pass-through devices requires that a memory reservation be made for the full amount of allocated memory. This reservation is done automatically initially, but if a model upgrade requires more memory, you must manually increase the reservation before powering on the VM.

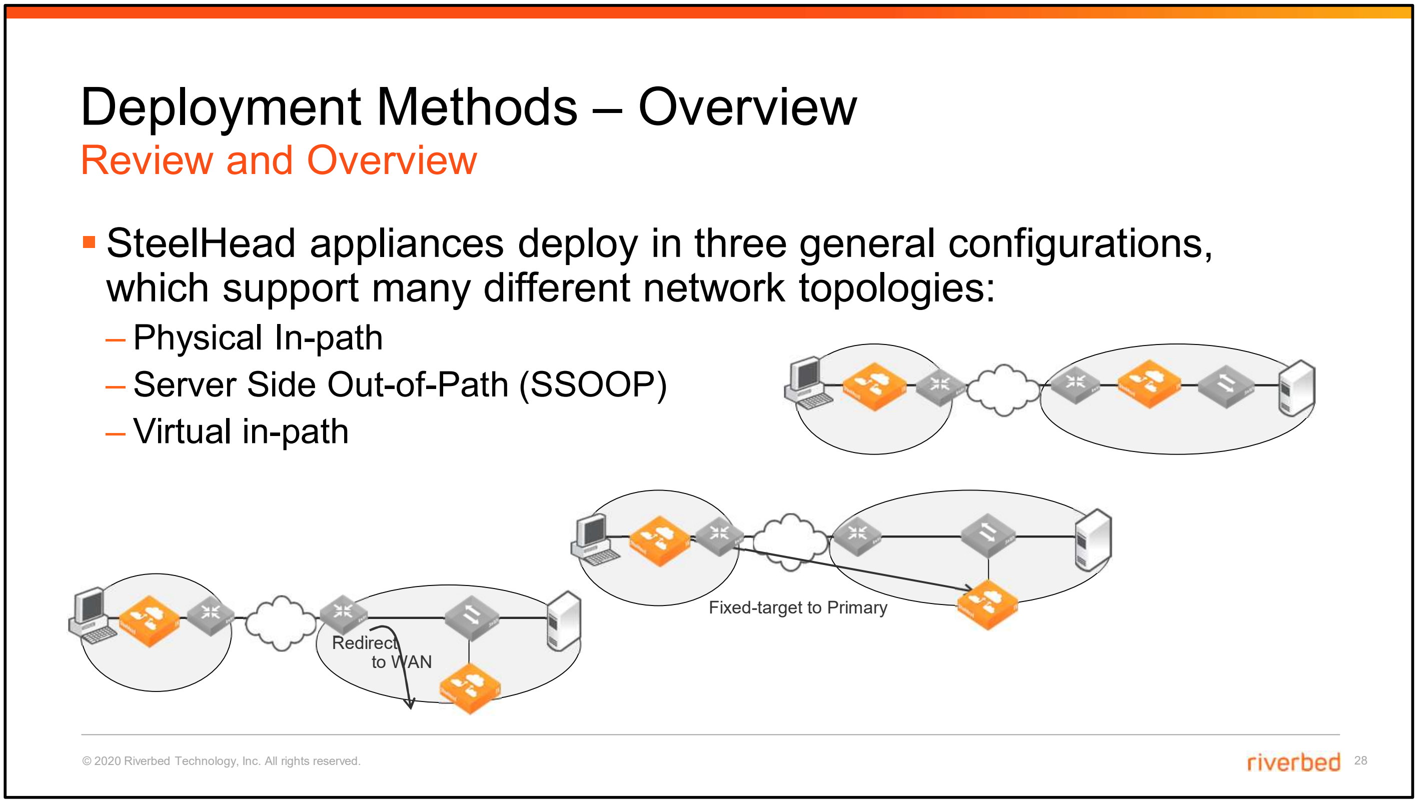


SteelHead-v – Best Practices

Best Practices for Performance

- Do not share NICs and use at least 1Gbps
- Allow resources Hypervisor overhead
- Do not over provision CPU
- Use a Server Grade CPU for the Hypervisor
- Always reserve RAM & Virtual RAM < Physical RAM
- Use SSDs or other high speed disk for the Segstore
- Do not share physical disks between hosts
- Do not use hyperthreading
- Apply BIOS power settings for maximum performance

Compare the Deployment Options at a High-level



Deployment Methods – Overview

Review and Overview

- SteelHead appliances deploy in three general configurations, which support many different network topologies:
  - Physical In-path
  - Server Side Out-of-Path (SSOOP)
  - Virtual in-path

Physical In-Path – The SteelHead appliance is *physically* in the direct path between the client and the server. The clients and servers continue to see client and server IP addresses. Physical in-path configurations are suitable for any location where the total bandwidth is within the limits of the installed SteelHead appliance.

Virtual In-Path – The SteelHead appliance is *virtually* in the path between the client and the server. This differs from a physical in-path in that a packet redirection mechanism is used to direct packets to SteelHead appliances that are not in the physical path. Redirection mechanisms include WCCP, Layer-4 switches, and PBR. In this configuration, clients and servers continue to see client and server IP addresses.

Out-of-Path – The SteelHead appliance is not in the direct path between the client and the server. Servers see the IP address of the server-side SteelHead appliance rather than the client IP address, which might impact security policies. An out-of-path configuration is suitable for datacenter locations where physically in-path or logically in-path configurations are not possible.

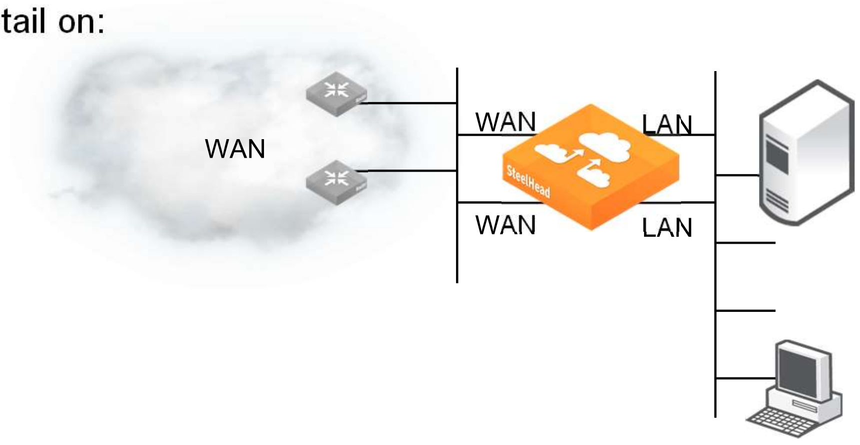
One critical requirement for any SteelHead appliance deployment is that the two SteelHead appliances involved in optimization of any given connection must see all packets for that connection in both directions (from client to server, and server to client).

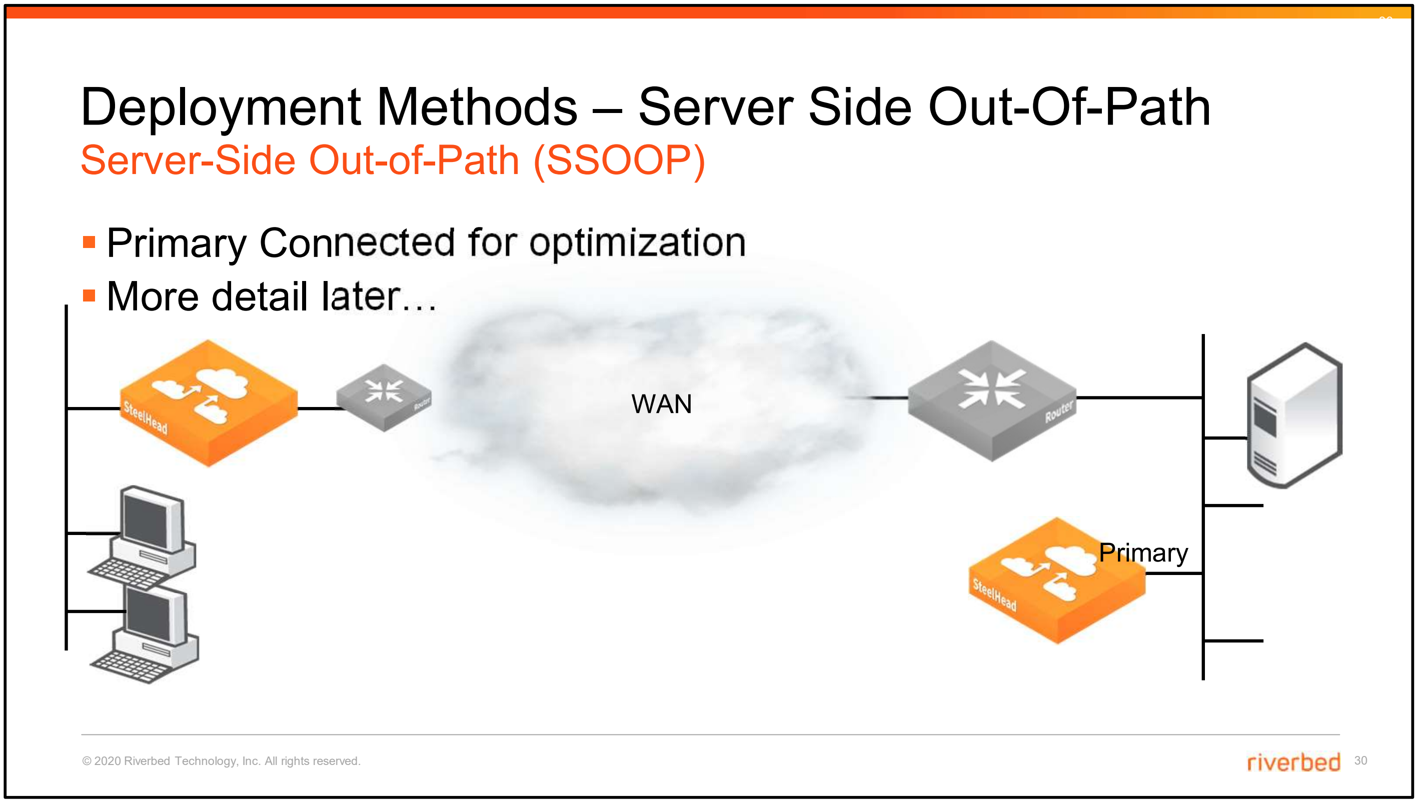
You can configure SteelHead appliances for in-path or virtual in-path deployment for TCP-over-IPv6 traffic. Riverbed also supports server-side out-of-path deployments. Considerations for the deployment type are the same as the considerations for optimizing IPv4 connections. Network integration features such as fail-to-wire, link state propagation, parallel deployments, firewalls, and so on continue to be relevant for optimization of TCP-over-IPv6 traffic. IPv4 connections can co-exist with TCP-over-IPv6 traffic.

Deployment Methods – Physical In-Path

In-path Options

- LAN & WAN Connected
- We will look into more detail on:
  - Serial clusters
  - Parallel clusters
  - Simplified Routing
  - Failover Scenarios





Deployment Methods – Server Side Out-Of-Path

Server-Side Out-of-Path (SSOOP)

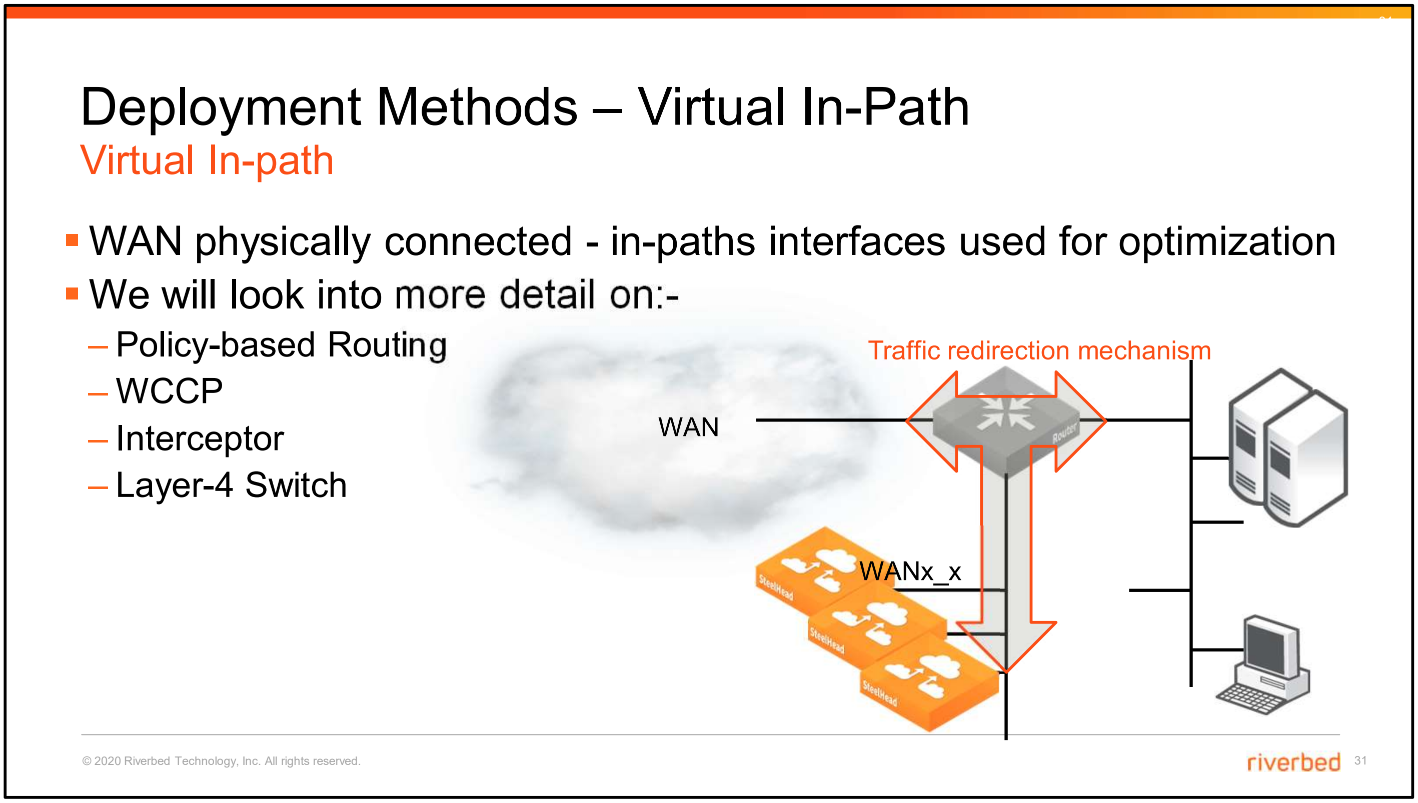
- Primary Connected for optimization
- More detail later…

riverbed 30

Server-side Out-of-Path – The SteelHead appliance is not in the direct path between the client and the server. Servers see the IP address of the server-side SteelHead appliance rather than the client IP address, which might impact security policies. An out-of-path configuration is suitable for datacenter locations where physically in-path or logically in-path configurations are not possible.

One critical requirement for any SteelHead appliance deployment is that the two SteelHead appliances involved in optimization of any given connection must see all packets for that connection in both directions (from client to server, and server to client).

You can configure SteelHead appliances for in-path or virtual in-path deployment for TCP-over-IPv6 traffic. Riverbed also supports server-side out-of-path deployments. Considerations for the deployment type are the same as the considerations for optimizing of IPv4 connections. Network integration features such as fail-to-wire, link state propagation, parallel deployments, firewalls, and so on continue to be relevant for optimization of TCP-over-IPv6 traffic. IPv4 connections can co-exist with TCP-over-IPv6 traffic.



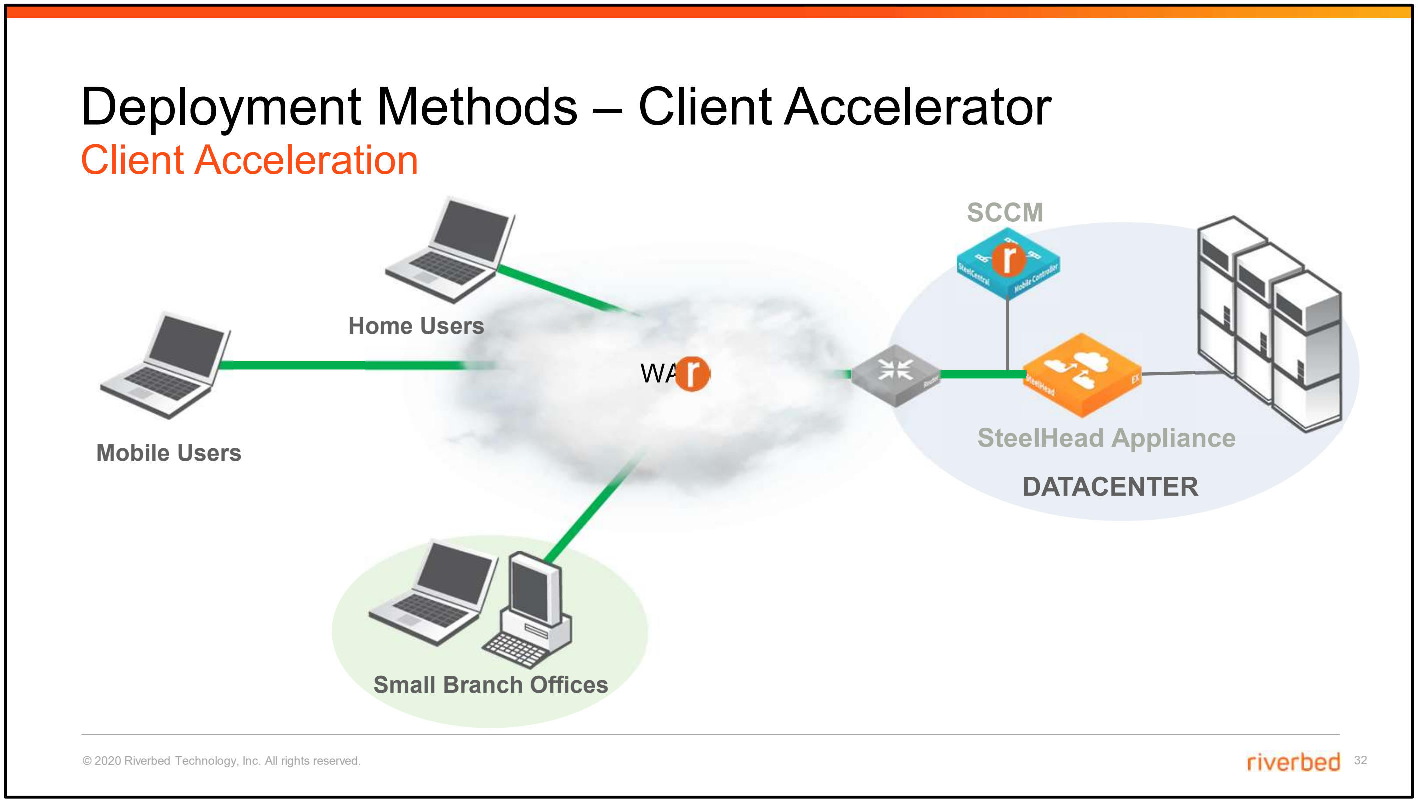
Deployment Methods – Virtual In-Path

Virtual In-path

- WAN physically connected - in-paths interfaces used for optimization
- We will look into more detail on:-
  - Policy-based Routing
  - WCCP
  - Interceptor
  - Layer-4 Switch

riverbed 31

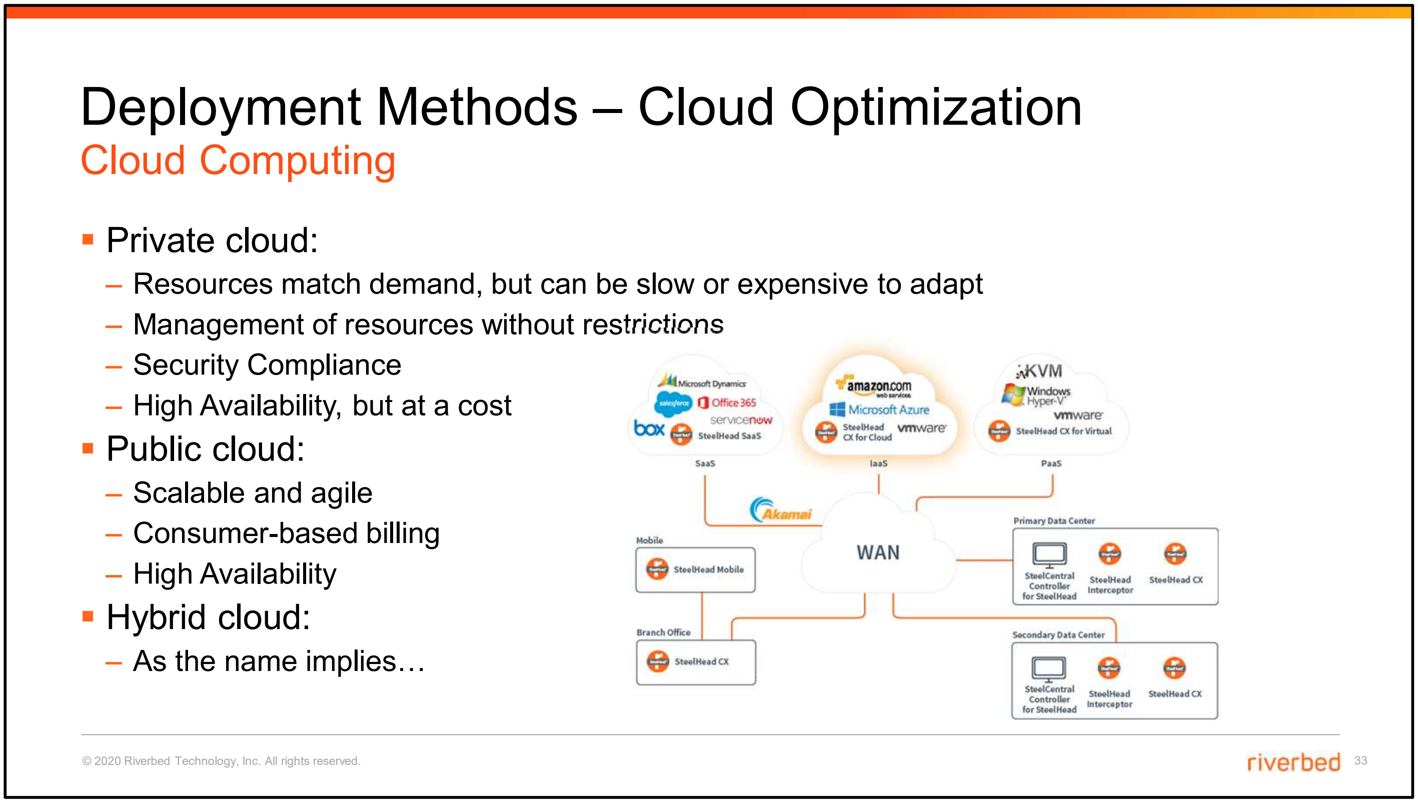
Virtual In-Path – The SteelHead appliance is virtually in the path between the client and the server. This differs from a physical in-path in that a packet redirection mechanism is used to direct packets to SteelHead appliances that are not in the physical path. Redirection mechanisms include WCCP, Layer-4 switches, and PBR. In this configuration, clients and servers continue to see client and server IP addresses.



Deployment Methods – Client Accelerator

Client Acceleration

- Delivers mobile optimization and application acceleration to mobile workers whether from a laptop or a desktop in the branch office.
- Provides mobile workers with access to corporate files and applications.
- Improves productivity for on-the-go workers and branch offices.
- Accelerates business-critical web applications up to 60x.
- Reduces bandwidth by up to 99%.
- Interacts directly with any SteelHead solution to optimize applications either on the DC or in the cloud.



Deployment Methods – Cloud Optimization
Cloud Computing

- Private cloud:
  - Resources match demand, but can be slow or expensive to adapt
  - Management of resources without restrictions
  - Security Compliance
  - High Availability, but at a cost
- Public cloud:
  - Scalable and agile
  - Consumer-based billing
  - High Availability
- Hybrid cloud:
  - As the name implies…

riverbed 33

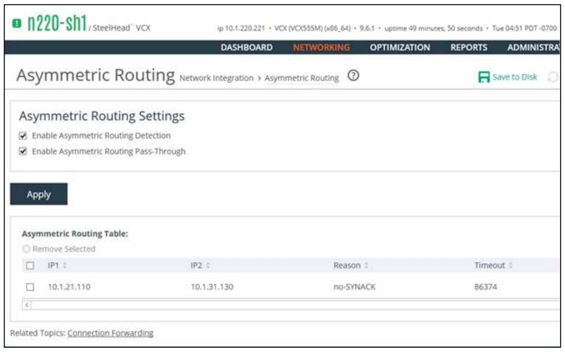
A private cloud hosting solution, also known as an internal or enterprise cloud, resides on a company's intranet or hosted datacenter where all data is protected behind a firewall. This is a sufficient solution for companies that have their own datacenters because they can use their current infrastructure. However, the main drawback with a private cloud is that all management, maintenance, and updating of datacenters is the responsibility of the company. And, over time, it is expected that the datacenter servers will need to be replaced, which can get very expensive. On the other hand, private clouds offer an increased level of security and they share very few, if any, resources with other organizations.

Deployment Methods & Network Asymmetry

Network Asymmetry Affects Optimization

Asymmetric Routing (AR) can be disruptive in optimized environments!

- SteelHeads need two-way traffic to optimize, regardless of topology:
  - In-path or virtual in-path
  - Virtual in-path can also be a solution, more later
- As a result:
  - The initial connection will take longer, or even fail
  - SteelHeads will pass through all IP pairs for *24 hours* when AR detected
    - This is to avoid reoccurring failures
    - The table can be flushed manually, but must be done on all affected SteelHeads
    - Remember *NOTHING* happens until we see a SYN message to start optimization
    - Consider manual reset or an auto-kickoff rule


n220-sh1 / SteelHead VCX
ip 10.1.220.221 • VCX (VCX555M) (x86_64) • 9.6.1 • uptime 49 minutes, 50 seconds • Tue 04:51 PDT -0700
DASHBOARD
NETWORKING
OPTIMIZATION
REPORTS
ADMINISTRA
Asymmetric Routing
Network Integration › Asymmetric Routing
Save to Disk
Apply
Remove Selected
Related Topics: Connection Forwarding

Asymmetric Routing Settings

☑ Enable Asymmetric Routing Detection

☑ Enable Asymmetric Routing Pass-Through

Asymmetric Routing Table:

□	IP1	IP2	Reason	Timeout
□	10.1.21.110	10.1.31.130	no-SYNACK	86374

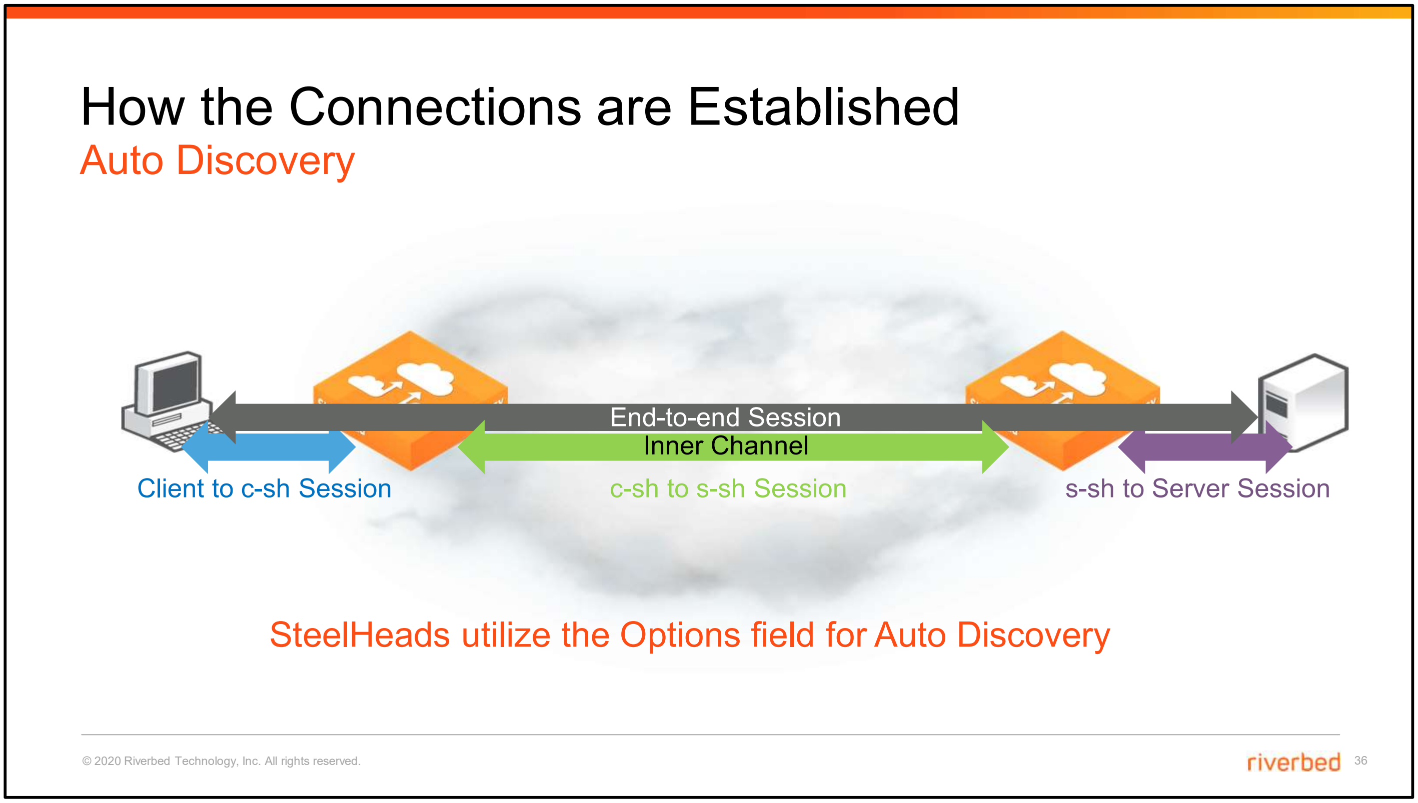
Asymmetric routing is when a packet takes one path to the destination and takes another path when returning to the source. Asymmetric routing is common within most networks; the larger the network, the more likely there's asymmetric routing in the network.

Asymmetric routing is undesirable for many network devices, including firewalls, VPNs, and SteelHeads. These devices all rely on seeing every packet to function properly. When SteelHeads are deployed in a network, all TCP traffic must flow through the same SteelHeads in the forward and reverse directions. If traffic flows through a SteelHead in one direction and not the other, then TCP clients are unable to make connections to TCP servers.

Asymmetric automatic detection enables SteelHeads to detect the presence of asymmetry within the network. Asymmetry is detected by the client-side SteelHeads. Once detected, the SteelHead passes through asymmetric traffic unoptimized allowing the TCP connections to continue to work. The first TCP connection for a pair of addresses might be dropped because during the detection process the SteelHeads have no way of knowing that the connection is asymmetric.

If asymmetric routing is detected, an entry is placed in the asymmetric routing table and any subsequent connections from that IP-address pair is passed through unoptimized. Further connections between these hosts aren't optimized until that particular asymmetric routing cache entry times out.

Describe Auto Discovery



How the Connections are Established

Auto Discovery

End-to-end Session

Inner Channel

Client to c-sh Session

c-sh to s-sh Session

s-sh to Server Session

SteelHeads utilize the Options field for Auto Discovery

riverbed 36

Note:
c-sh = client SteelHead
s-sh = server SteelHead

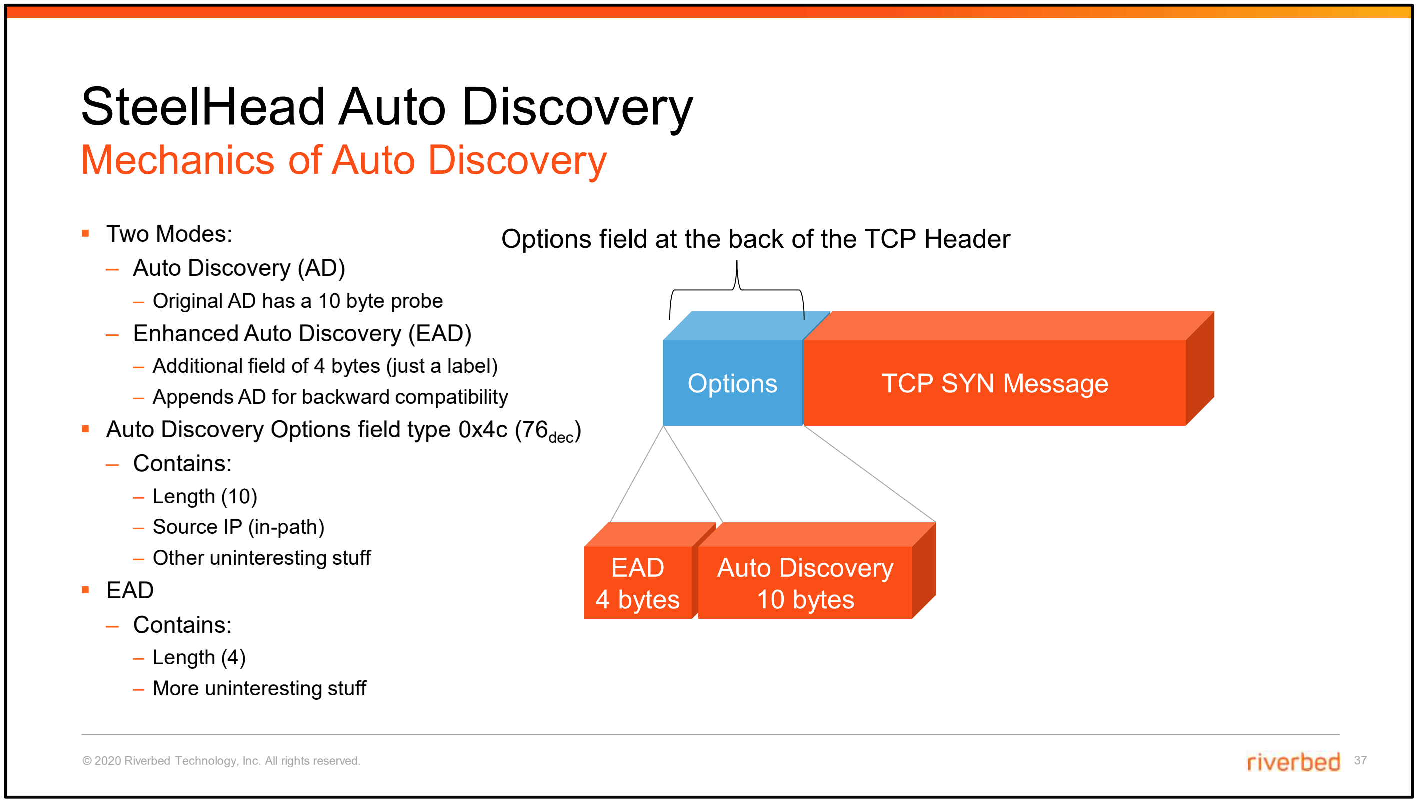
Regarding TCP Option Size

The RiOS probe is added to the SYN packet as a TCP option.
A TCP header can be at most 60 bytes and the usual TCP header info takes 20 bytes, leaving only 40 bytes for TCP options. Other TCP options exist even before we add our probe, reducing the 40 bytes we have to work with. For instance, a 20 byte option is not unusual on Linux so there are really only 20 bytes left for the probe.

Common TCP options include:

- 0: Eol (1 byte)
- 1: Nop (1 byte)
- 2: Max segment size (4 bytes)
- 3: Window scale (3 bytes)
- 4: Sack permitted (2 bytes)
- 8: Timestamp (10 bytes)

All options registered can be found at IANA (Internet Assigned Numbers Authority) - http://www.iana.org/

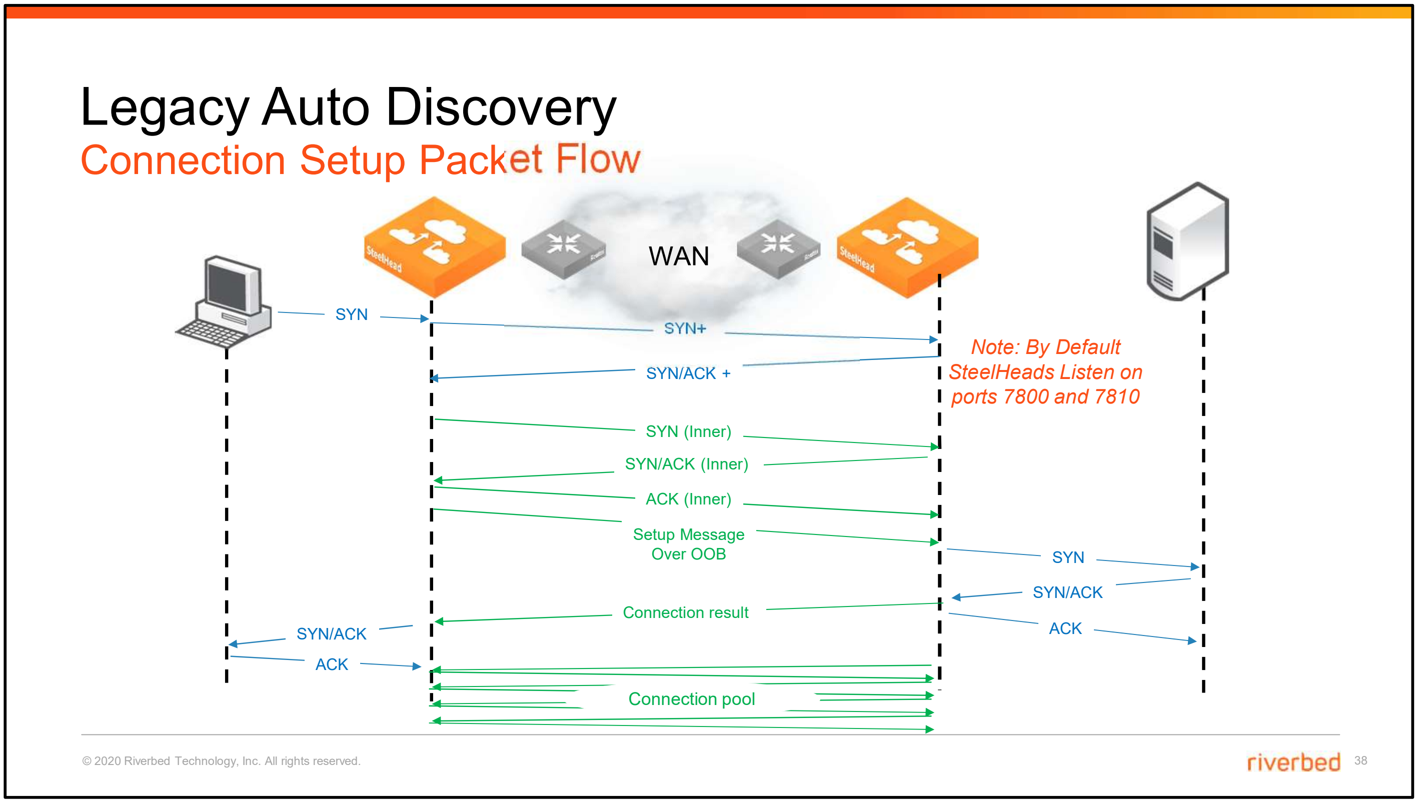


SteelHead Auto Discovery

Mechanics of Auto Discovery

- Two Modes:
  - Auto Discovery (AD)
    - Original AD has a 10 byte probe
  - Enhanced Auto Discovery (EAD)
    - Additional field of 4 bytes (just a label)
    - Appends AD for backward compatibility
- Auto Discovery Options field type 0x4c (76_{dec})
  - Contains:
    - Length (10)
    - Source IP (in-path)
    - Other uninteresting stuff
- EAD
  - Contains:
    - Length (4)
    - More uninteresting stuff

Options field at the back of the TCP Header



Legacy Auto Discovery

Connection Setup Packet Flow

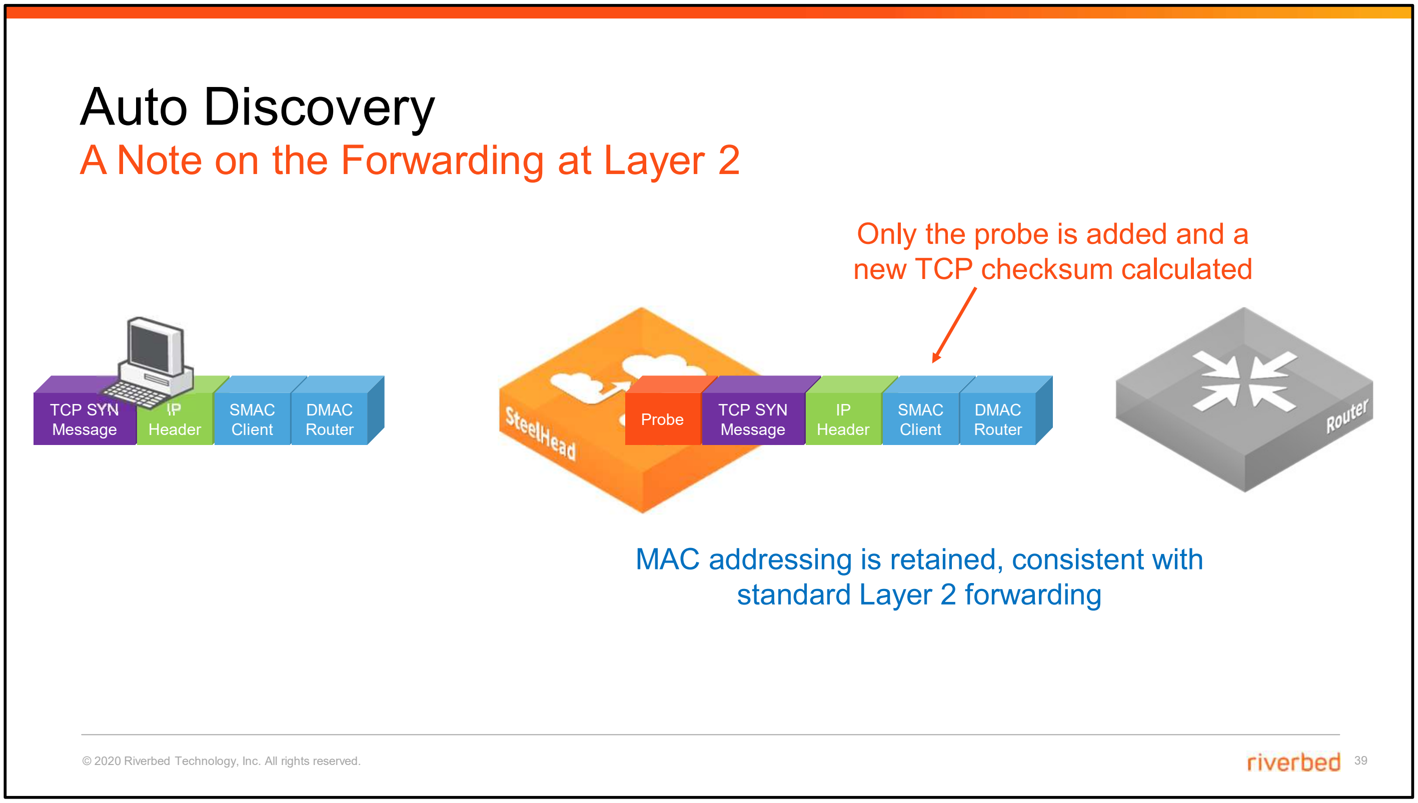
Note: By Default SteelHeads Listen on ports 7800 and 7810

riverbed 38

Legacy AD is having a bit of a renaissance due in no small part to SaaS backhaul deployments

This is the basic sequence diagram for auto-discovery. Note the out-of-band splice. There will always be one of these for every pair of optimizing SteelHead appliances. It communicates version and feature information but not user data.

The key point to remember here is that for each optimized connection there are, in fact, three separate TCP connections: One between the client and the client-side SteelHead, one between the server and server-side SteelHead, and a third, the *inner connection*, between the two SteelHead devices themselves. These connections are quite separate, with their own set of sequence numbers, and hold the key to how we can optimize applications. Because the outer- and inner- connections are separate, the number of turns does not need to be equal. So, for chatty applications, we can have the turns and round trips on the LAN, where there is no problem with latency, but keep the optimized, *inner* connection free from turns to speed up our performance. The way we do this is through a technique called *Transaction Prediction* or *Application Streamlining,* different for the different protocols, but identical in their aim to keep transactions local.



Auto Discovery

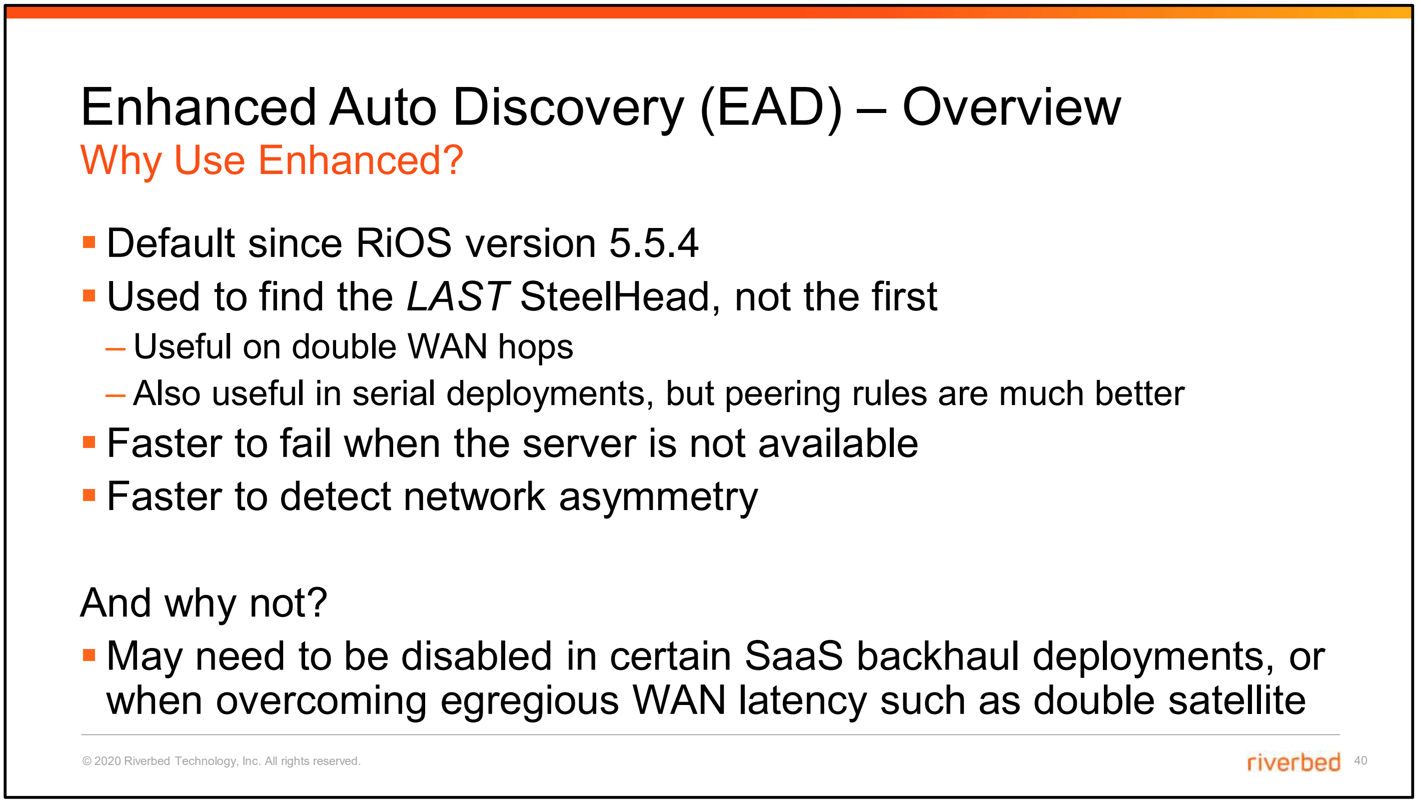
A Note on the Forwarding at Layer 2

Only the probe is added and a new TCP checksum calculated

MAC addressing is retained, consistent with standard Layer 2 forwarding

riverbed 39

The LAN and WAN interfaces are bridged together at layer two. You will cause a broadcast storm by putting the in the same VLAN. In fact if that is the case on a virtual SteelHead it will not even create the LAN, WAN or in-path interfaces if it detects the same VLAN on the LAN and WAN.



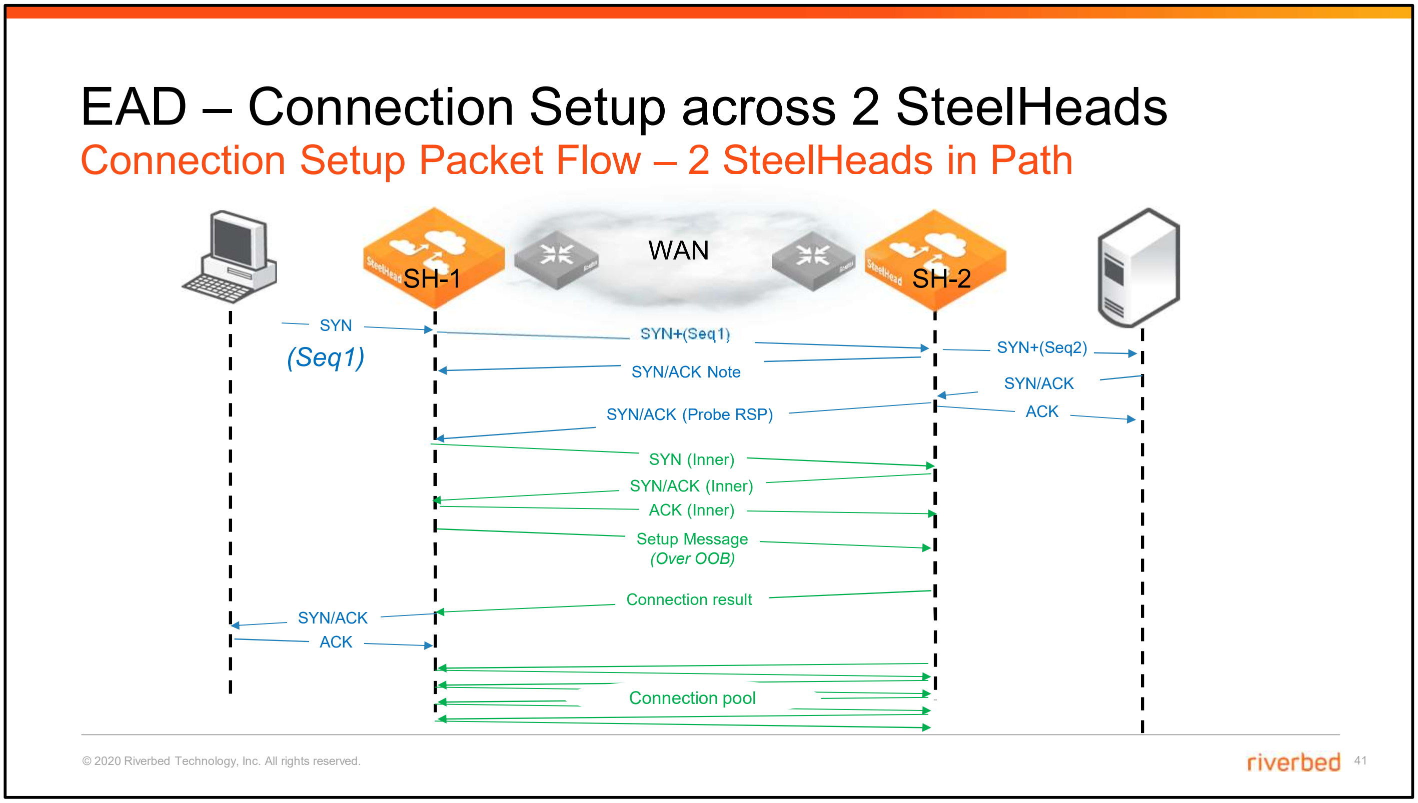
Enhanced Auto Discovery (EAD) – Overview

Why Use Enhanced?

- Default since RiOS version 5.5.4
- Used to find the *LAST* SteelHead, not the first
  - Useful on double WAN hops
  - Also useful in serial deployments, but peering rules are much better
- Faster to fail when the server is not available
- Faster to detect network asymmetry

And why not?

- May need to be disabled in certain SaaS backhaul deployments, or when overcoming egregious WAN latency such as double satellite

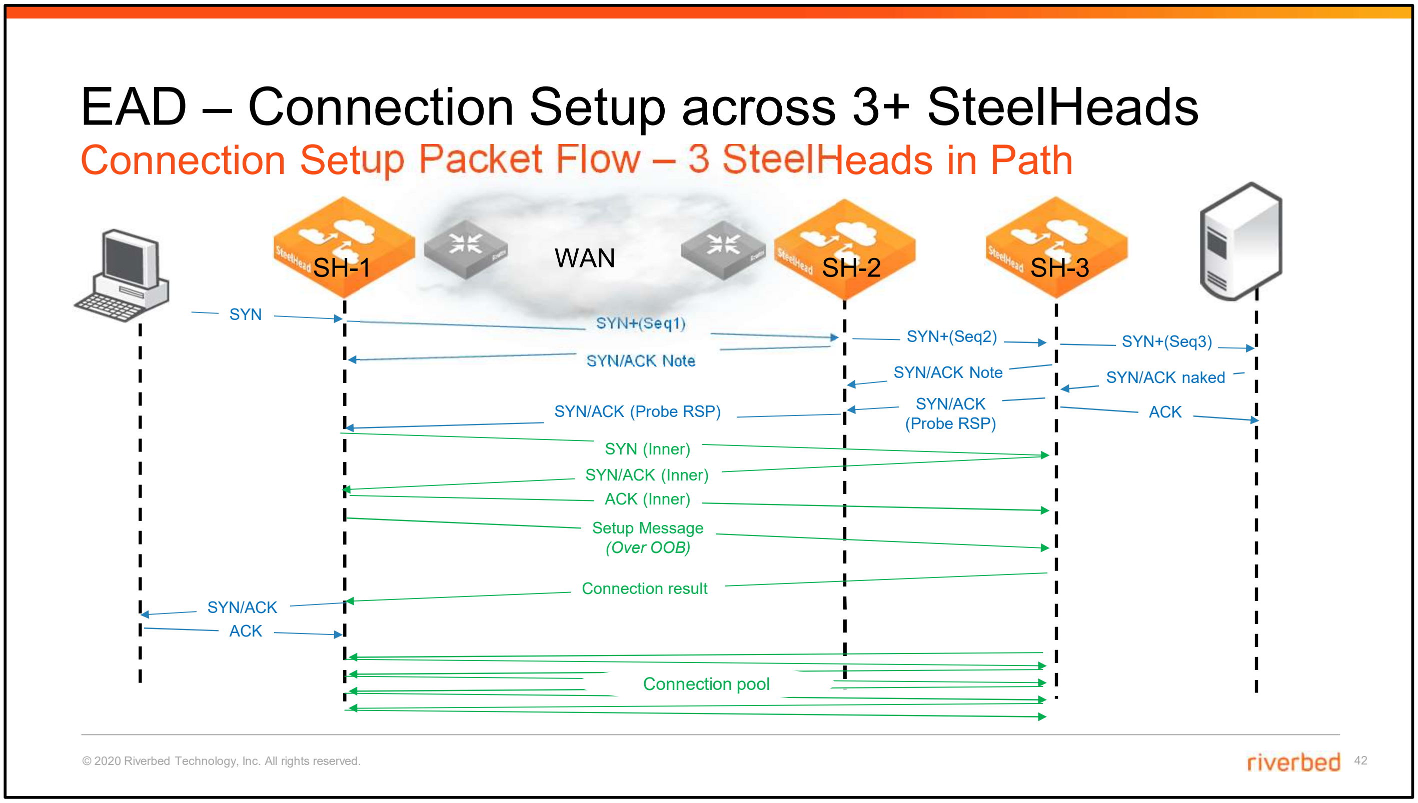


EAD – Connection Setup across 2 SteelHeads

Connection Setup Packet Flow – 2 SteelHeads in Path

riverbed 41

If the inner connection fails after the server-side SteelHead 2 has connected to the server and the auto-discovery succeeded, we need to tear down that server-side connection and let the connection pass through. We also need to resend the original SYN to establish the connection. During the time the client-side SteelHead is trying to connect to the server-side SteelHead, any SYN retransmit from the client is dropped.



EAD – Connection Setup across 3+ SteelHeads

Connection Setup Packet Flow – 3 SteelHeads in Path

SH-1
WAN
SH-2
SH-3

SYN
SYN+(Seq1)
SYN+(Seq2)
SYN+(Seq3)
SYN/ACK Note
SYN/ACK Note
SYN/ACK naked
SYN/ACK (Probe RSP)
SYN/ACK (Probe RSP)
ACK
SYN (Inner)
SYN/ACK (Inner)
ACK (Inner)
Setup Message *(Over OOB)*
Connection result
SYN/ACK
ACK
Connection pool

riverbed 42

If the inner connection fails after the server-side SteelHead 3 has connected to the server and the auto-discovery succeeded, we need to tear down that server-side connection and let the connection pass through. We also need to resend the original SYN to establish the connection. During the time the client-side SteelHead is trying to connect to the server-side SteelHead, any SYN retransmit from the client is dropped. Inner Channel Communication (SH1 – SH2 – SH3)

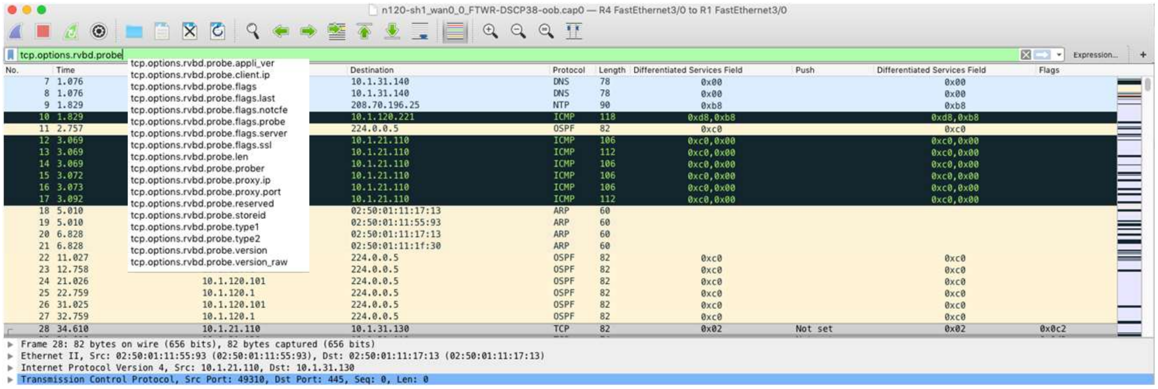
Note: SH3 doesn't send the probe response to SH1. SH3 sends the SYN/ACK probe response towards SH2, which sees the packet and responds to SH1.

In the process, you notice the change in sequence numbers occurring throughout the exchange. SH3 sends the probe response with the acknowledgement number (ACKNUM), which is different from the original SH1. The ACKNUM is responding to SH2. In turn, SH2 changes the ACKNUM to coincide to SH1 SEQ1. Therefore, if you had firewalls throughout the communication, it would see all corresponding sequence numbers SEQ1/ACK1 and SEQ2/ACK2.

Auto and Enhanced Auto Discovery – Wireshark filter

On the Wire

On Wireshark Filter by:
tcp.options.rvbd


n120-sh1_wan0_0_FTWR-DSCP38-oob.cap0 — R4 FastEthernet3/0 to R1 FastEthernet3/0
Expression...
+
tcp.options.rvbd.probe.appli_ver
tcp.options.rvbd.probe.client.ip
tcp.options.rvbd.probe.flags
tcp.options.rvbd.probe.flags.last
tcp.options.rvbd.probe.flags.notcfe
tcp.options.rvbd.probe.flags.probe
tcp.options.rvbd.probe.flags.server
tcp.options.rvbd.probe.flags.ssl
tcp.options.rvbd.probe.len
tcp.options.rvbd.probe.prober
tcp.options.rvbd.probe.proxy.ip
tcp.options.rvbd.probe.proxy.port
tcp.options.rvbd.probe.reserved
tcp.options.rvbd.probe.storeid
tcp.options.rvbd.probe.type1
tcp.options.rvbd.probe.type2
tcp.options.rvbd.probe.version
tcp.options.rvbd.probe.version_raw

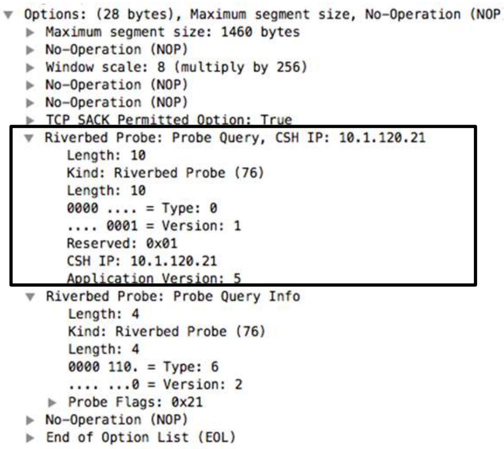
tcp.options.rvbd.probe

No.	Time		Destination	Protocol	Length	Differentiated Services Field	Push	Differentiated Services Field	Flags
7	1.076		10.1.31.140	DNS	78	0x00		0x00	
8	1.076		10.1.31.140	DNS	78	0x00		0x00	
9	1.829		208.70.196.25	NTP	90	0xb8		0xb8	
10	1.829		10.1.120.221	ICMP	118	0xd8,0xb8		0xd8,0xb8	
11	2.757		224.0.0.5	OSPF	82	0xc0		0xc0	
12	3.069		10.1.21.110	ICMP	106	0xc0,0x00		0xc0,0x00	
13	3.069		10.1.21.110	ICMP	112	0xc0,0x00		0xc0,0x00	
14	3.069		10.1.21.110	ICMP	106	0xc0,0x00		0xc0,0x00	
15	3.072		10.1.21.110	ICMP	106	0xc0,0x00		0xc0,0x00	
16	3.073		10.1.21.110	ICMP	106	0xc0,0x00		0xc0,0x00	
17	3.092		10.1.21.110	ICMP	112	0xc0,0x00		0xc0,0x00	
18	5.010		02:50:01:11:17:13	ARP	60				
19	5.010		02:50:01:11:55:93	ARP	60				
20	6.828		02:50:01:11:17:13	ARP	60				
21	6.828		02:50:01:11:1f:30	ARP	60				
22	11.027		224.0.0.5	OSPF	82	0xc0		0xc0	
23	12.758		224.0.0.5	OSPF	82	0xc0		0xc0	
24	21.026	10.1.120.101	224.0.0.5	OSPF	82	0xc0		0xc0	
25	22.759	10.1.120.1	224.0.0.5	OSPF	82	0xc0		0xc0	
26	31.025	10.1.120.101	224.0.0.5	OSPF	82	0xc0		0xc0	
27	32.759	10.1.120.1	224.0.0.5	OSPF	82	0xc0		0xc0	
28	34.610	10.1.21.110	10.1.31.130	TCP	82	0x02	Not set	0x02	0x0c2

```
▶ Frame 28: 82 bytes on wire (656 bits), 82 bytes captured (656 bits)
▶ Ethernet II, Src: 02:50:01:11:55:93 (02:50:01:11:55:93), Dst: 02:50:01:11:17:13 (02:50:01:11:17:13)
▶ Internet Protocol Version 4, Src: 10.1.21.110, Dst: 10.1.31.130
▶ Transmission Control Protocol, Src Port: 49310, Dst Port: 445, Seq: 0, Len: 0
```

Auto and Enhanced Auto Discovery – Discovery Probe

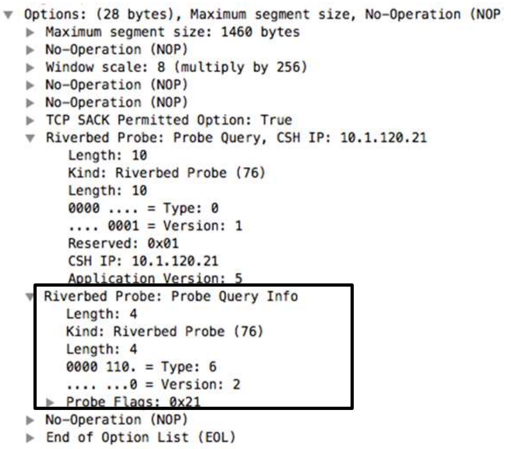
On the Wire



```
▼ Options: (28 bytes), Maximum segment size, No-Operation (NOP
  ▶ Maximum segment size: 1460 bytes
  ▶ No-Operation (NOP)
  ▶ Window scale: 8 (multiply by 256)
  ▶ No-Operation (NOP)
  ▶ No-Operation (NOP)
  ▶ TCP SACK Permitted Option: True
  ▼ Riverbed Probe: Probe Query, CSH IP: 10.1.120.21
      Length: 10
      Kind: Riverbed Probe (76)
      Length: 10
      0000 .... = Type: 0
      .... 0001 = Version: 1
      Reserved: 0x01
      CSH IP: 10.1.120.21
      Application Version: 5
  ▼ Riverbed Probe: Probe Query Info
      Length: 4
      Kind: Riverbed Probe (76)
      Length: 4
      0000 110. = Type: 6
      .... ...0 = Version: 2
    ▶ Probe Flags: 0x21
  ▶ No-Operation (NOP)
  ▶ End of Option List (EOL)
```

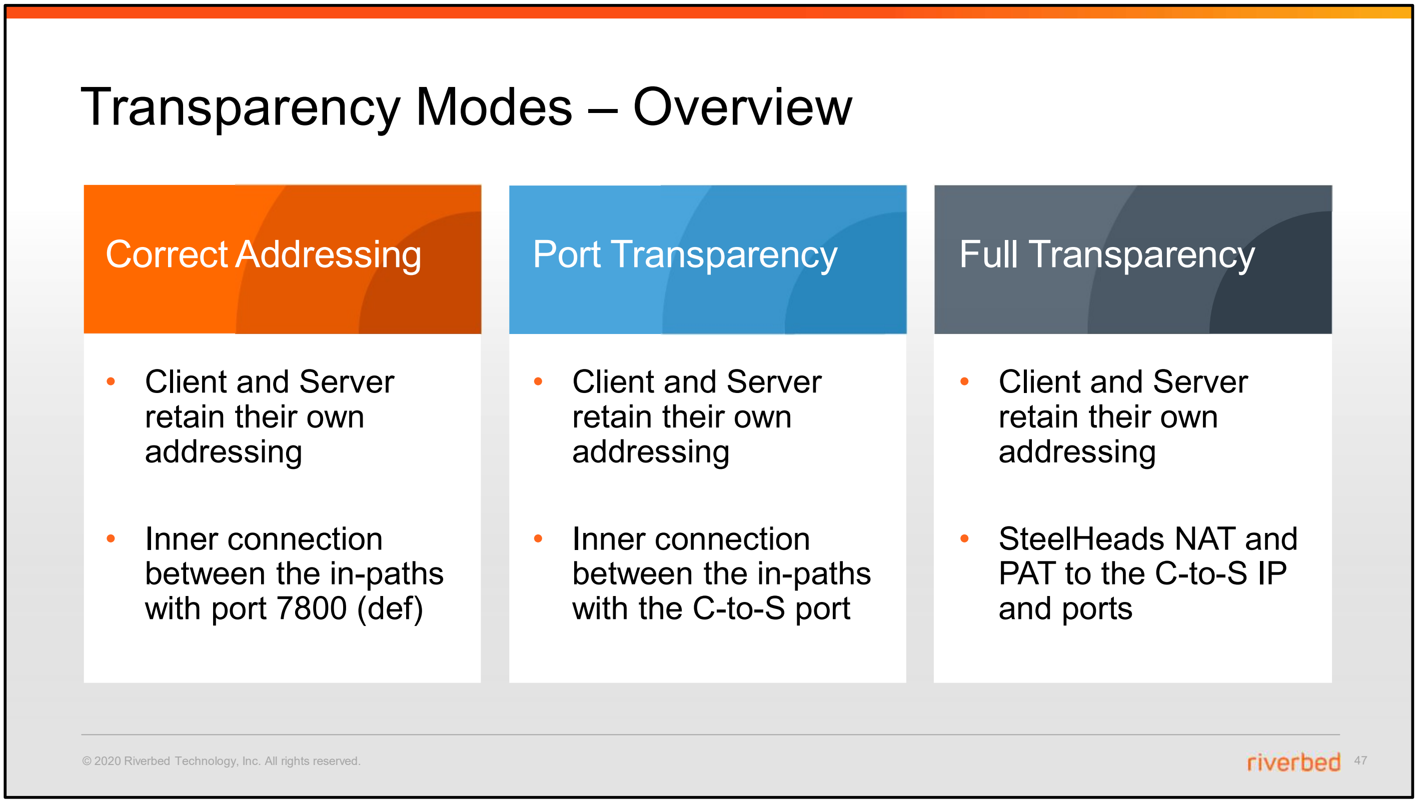
Auto and Enhanced Auto Discovery – Notification probe

On the Wire



```
▼ Options: (28 bytes), Maximum segment size, No-Operation (NOP
  ▶ Maximum segment size: 1460 bytes
  ▶ No-Operation (NOP)
  ▶ Window scale: 8 (multiply by 256)
  ▶ No-Operation (NOP)
  ▶ No-Operation (NOP)
  ▶ TCP SACK Permitted Option: True
  ▼ Riverbed Probe: Probe Query, CSH IP: 10.1.120.21
       Length: 10
       Kind: Riverbed Probe (76)
       Length: 10
       0000 .... = Type: 0
       .... 0001 = Version: 1
       Reserved: 0x01
       CSH IP: 10.1.120.21
       Application Version: 5
  ▼ Riverbed Probe: Probe Query Info
       Length: 4
       Kind: Riverbed Probe (76)
       Length: 4
       0000 110. = Type: 6
       .... ...0 = Version: 2
     ▶ Probe Flags: 0x21
  ▶ No-Operation (NOP)
  ▶ End of Option List (EOL)
```

Configure Transparency Modes



Transparency Modes – Overview

Correct Addressing

- Client and Server retain their own addressing
- Inner connection between the in-paths with port 7800 (def)

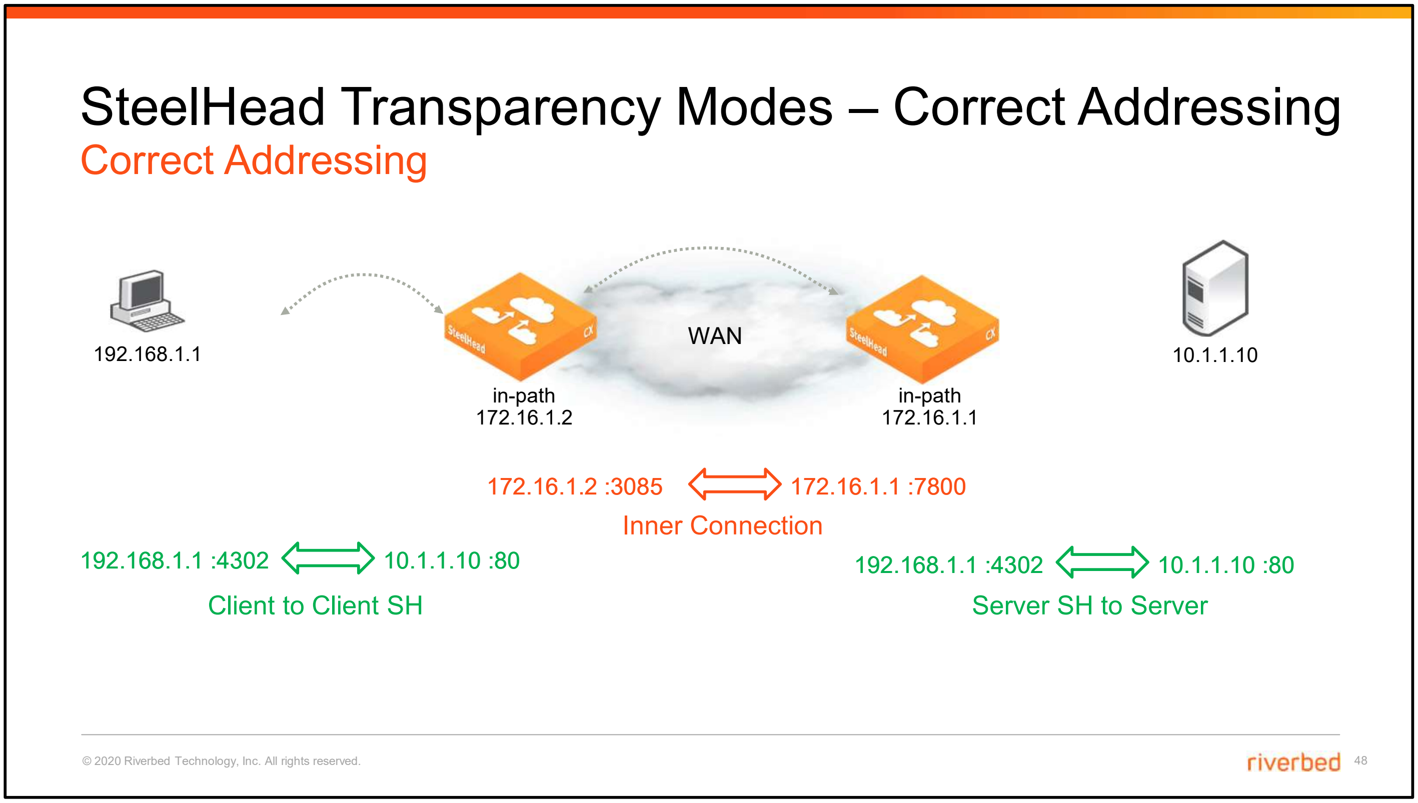
Port Transparency

- Client and Server retain their own addressing
- Inner connection between the in-paths with the C-to-S port

Full Transparency

- Client and Server retain their own addressing
- SteelHeads NAT and PAT to the C-to-S IP and ports

riverbed 47



SteelHead Transparency Modes – Correct Addressing

Correct Addressing

192.168.1.1

WAN

in-path
172.16.1.2

in-path
172.16.1.1

10.1.1.10

172.16.1.2 :3085 ⟺ 172.16.1.1 :7800

Inner Connection

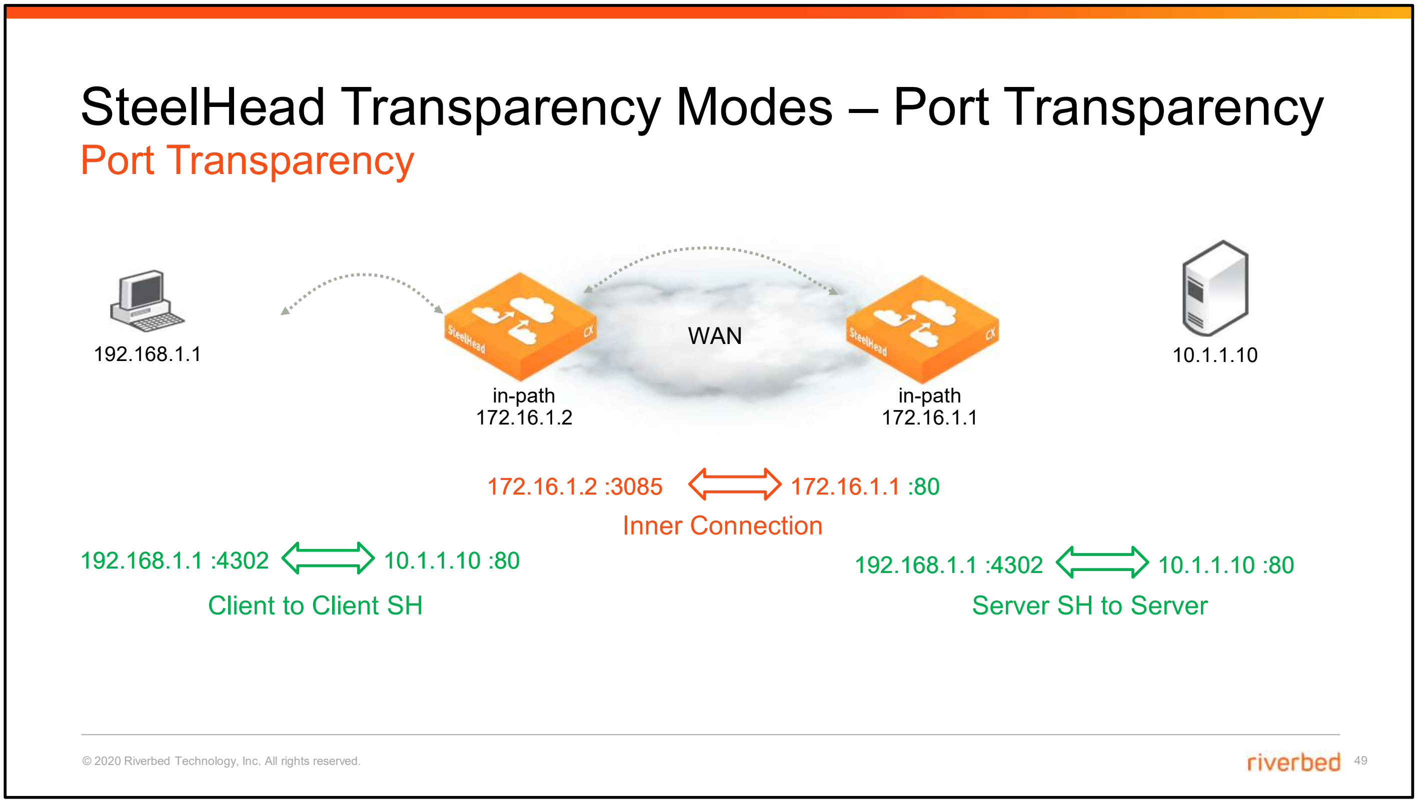
192.168.1.1 :4302 ⟺ 10.1.1.10 :80

Client to Client SH

192.168.1.1 :4302 ⟺ 10.1.1.10 :80

Server SH to Server

Correct Addressing – Turns WAN visibility off. Correct addressing uses SteelHead appliance IP addresses and port numbers in the TCP/IP packet header fields for optimized traffic in both directions across the WAN. This is the default setting.



SteelHead Transparency Modes – Port Transparency

Port Transparency

172.16.1.2 :3085 ⇔ 172.16.1.1 :80

Inner Connection

192.168.1.1 :4302 ⇔ 10.1.1.10 :80

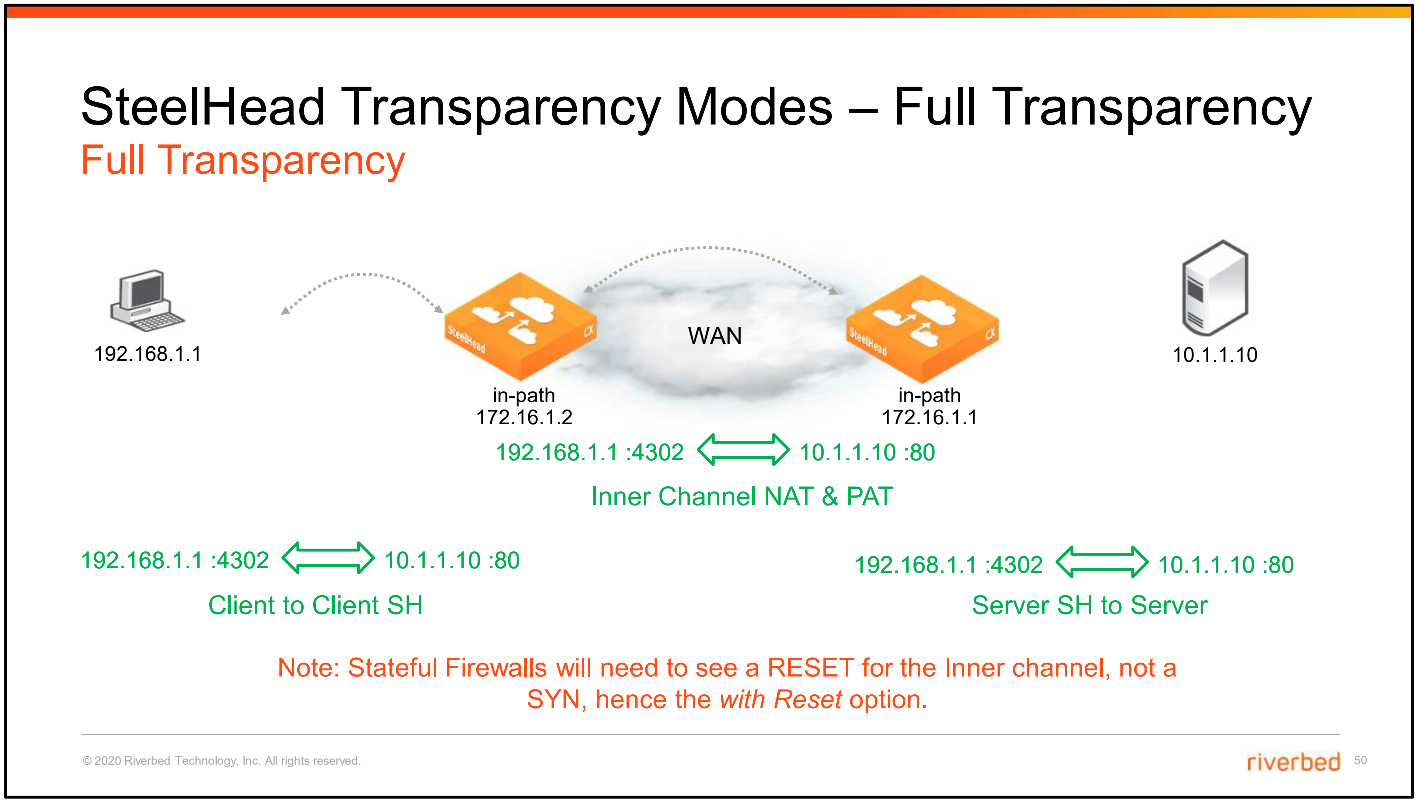
Client to Client SH

192.168.1.1 :4302 ⇔ 10.1.1.10 :80

Server SH to Server

riverbed 49

Port Transparency – Preserves your destination port numbers in the TCP/IP header fields for optimized traffic in both directions across the WAN. Traffic is optimized while the server port number in the TCP/IP header field appears to be unchanged. Routers and network monitoring devices deployed in the WAN segment between the communicating SteelHead appliances can view these preserved fields. Example use cases are when classifying traffic based on the destination port for QoS, Security ACLs, Policy Based Routing, etc.



SteelHead Transparency Modes – Full Transparency

Full Transparency

WAN

192.168.1.1

in-path
172.16.1.2

in-path
172.16.1.1

10.1.1.10

192.168.1.1 :4302 ⇔ 10.1.1.10 :80

Inner Channel NAT & PAT

192.168.1.1 :4302 ⇔ 10.1.1.10 :80

Client to Client SH

192.168.1.1 :4302 ⇔ 10.1.1.10 :80

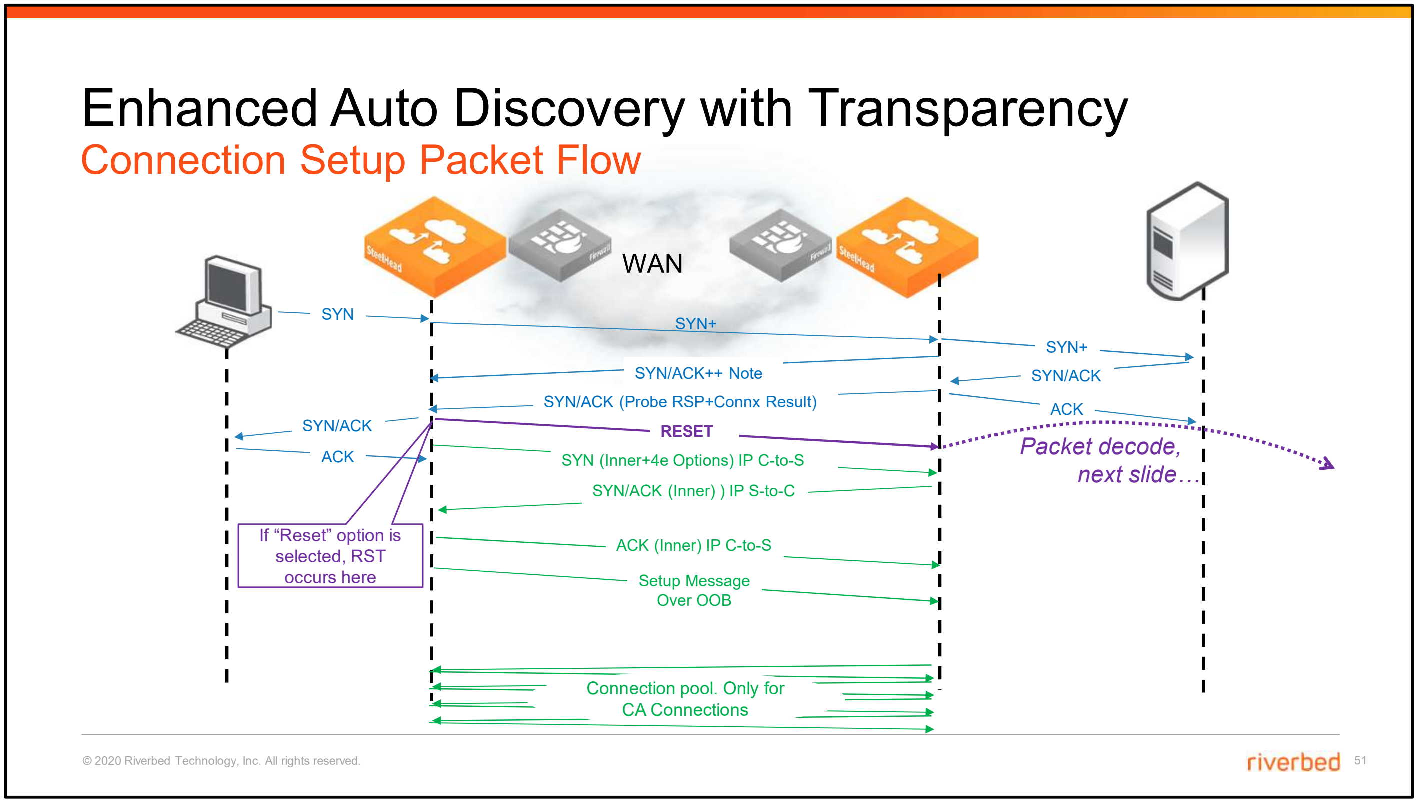
Server SH to Server

Note: Stateful Firewalls will need to see a RESET for the Inner channel, not a SYN, hence the *with Reset* option.

riverbed 50

Full Transparency – Preserves your client and server IP addresses and port numbers in the TCP/IP header fields for optimized traffic in both directions across the WAN. Traffic is optimized while these TCP/IP header fields appear to be unchanged. Routers and network monitoring devices deployed in the WAN segment between the communicating SteelHead appliances can view these preserved fields.

Full Transparency w/Reset – Enables full address and port transparency and also sends a forward reset between receiving the probe response and sending the transparent inner channel SYN. This ensures the firewall does not block inner transparent connections because of information stored in the probe connection. The forward reset is necessary because the probe connection and inner connection use the same IP addresses and ports and both map to the same firewall connection. The reset clears the probe connection created by the SteelHead appliance and allows for the full transparent inner connection to traverse the firewall.



Enhanced Auto Discovery with Transparency

Connection Setup Packet Flow

WAN
SYN
SYN+
SYN+
SYN/ACK++ Note
SYN/ACK
SYN/ACK (Probe RSP+Connx Result)
ACK
SYN/ACK
RESET
ACK
SYN (Inner+4e Options) IP C-to-S
Packet decode, next slide…
SYN/ACK (Inner)) IP S-to-C
If "Reset" option is selected, RST occurs here
ACK (Inner) IP C-to-S
Setup Message Over OOB
Connection pool. Only for CA Connections

riverbed 51

There are common networking problems inherent to transparent addressing.

Network Asymmetry – Enabling full address transparency increases the likelihood of problems inherent to asymmetric routing. If the router has a route to the client or server that does not pass through a SteelHead appliance, and it transmits the optimized packets on that route, the optimized and LAN-side connections might fail.

Firewalls Located Between SteelHead Appliances – Transparent addressing can have an impact on systems that monitor or alter the state of TCP connections on the WAN for reporting, security, or congestion control.

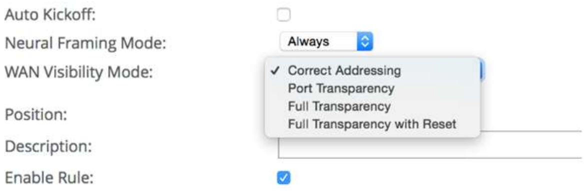
Misrouting Optimized Traffic – In an environment in which transparent addressing is used, if the server-side SteelHead appliance is not functioning, or if a packet is routed along an alternative network path, the packet might go from the client-side SteelHead appliance directly to the server. Because the server-side SteelHead appliance does not have an opportunity to convert the packet to its native format, the server cannot recognize it, and the connection fails. In most cases, the server is able to detect whether a packet contains invalid payload information or, in this case, has an unrecognizable format, and rejects the packet. Assuming that the server does detect that the format is unrecognizable, the server rejects the packet and resets the TCP connection. If the client TCP connection is reset, the client can reconnect to the server without any SteelHead appliance involvement. Of course, this type of traffic misrouting can occur in both directions across the WAN.

Important: Before enabling and using full address transparency, carefully consider the risks and exposures in the event that a server accepts and routes a packet that has an unrecognizable format.

Transparency Options Field

Carried in *EVERY* packet, after SYN and SYN/ACK

- Configured on in-path rules:
  - Can be with Fixed Target (on cli and in-path only)
  - Filter on Wireshark: tcp.options.rvbd.trpy
- 4e labels packets as “WAN side Optimized” for use by:
  - SteelHeads
  - Interceptors
  - SteelCentral devices
- Contains pseudo Src/Dest IP and ports
- Effectively NAT/PAT on egress of SH



Auto Kickoff: □

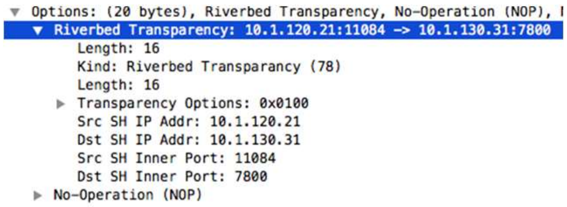
Neural Framing Mode: Always

WAN Visibility Mode: ✓ Correct Addressing

Position:

Description:

Enable Rule: ☑

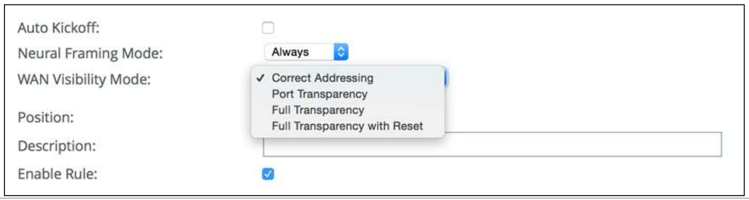


```
▼ Options: (20 bytes), Riverbed Transparency, No-Operation (NOP),
  ▼ Riverbed Transparency: 10.1.120.21:11084 -> 10.1.130.31:7800
        Length: 16
        Kind: Riverbed Transparancy (78)
        Length: 16
      ▶ Transparency Options: 0x0100
        Src SH IP Addr: 10.1.120.21
        Dst SH IP Addr: 10.1.130.31
        Src SH Inner Port: 11084
        Dst SH Inner Port: 7800
  ▶ No-Operation (NOP)
```

SteelHead Transparency Modes

The Need for “with Reset” on Full Transparency

- Stateful firewall monitor connection setups
- Sequence numbers changing on the fly are not allowed
- The initial SYN message carrying the probe will have one sequence number
- The inner channel will share the port and IP addressing but will have new sequencing
- RESET messages are the “legal” way to change sequence numbers
- This keeps stateful firewalls happy(er)



Auto Kickoff: □

Neural Framing Mode: Always

WAN Visibility Mode: ✓ Correct Addressing

Position:

Description:

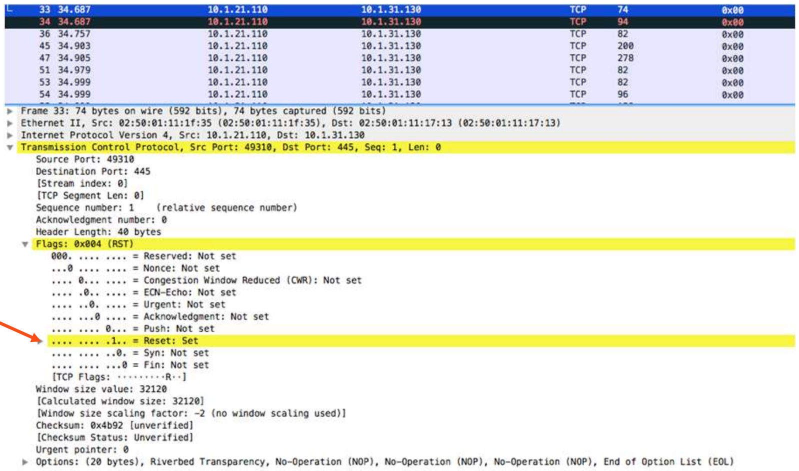
Enable Rule: ☑

riverbed

Full Transparency Establishment

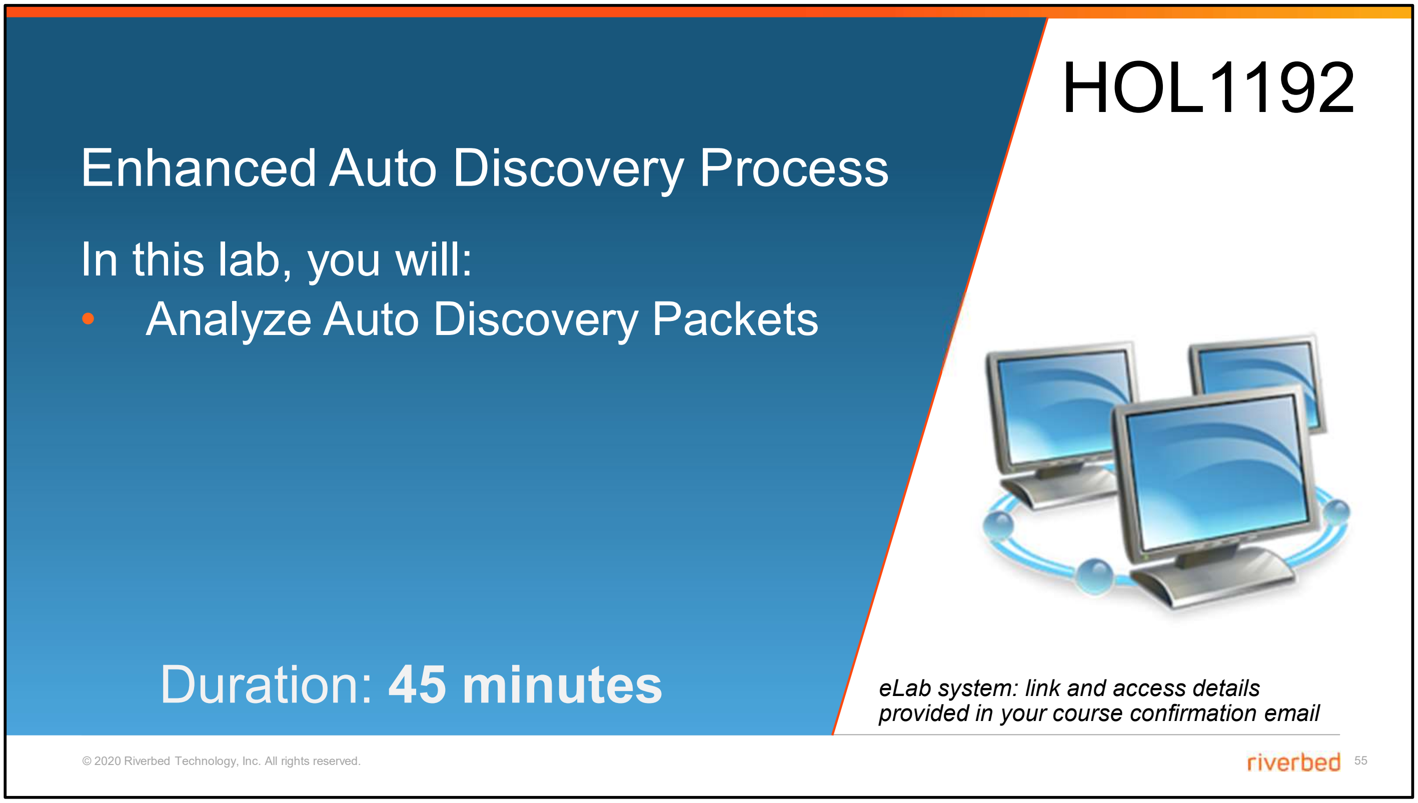
Decode from Wireshark

Here we can see that the Reset message follows the initial probe. As a side note, the syn that follows is the first message to carry the TRPY (4e) option.



```
  33 34.687       10.1.21.110     10.1.31.130      TCP   74     0x00
  34 34.687       10.1.21.110     10.1.31.130      TCP   94     0x00
  36 34.757       10.1.21.110     10.1.31.130      TCP   82     0x00
  45 34.903       10.1.21.110     10.1.31.130      TCP   200    0x00
  47 34.905       10.1.21.110     10.1.31.130      TCP   278    0x00
  51 34.979       10.1.21.110     10.1.31.130      TCP   82     0x00
  53 34.999       10.1.21.110     10.1.31.130      TCP   82     0x00
  54 34.999       10.1.21.110     10.1.31.130      TCP   96     0x00
▶ Frame 33: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▶ Ethernet II, Src: 02:50:01:11:1f:35 (02:50:01:11:1f:35), Dst: 02:50:01:11:17:13 (02:50:01:11:17:13)
▶ Internet Protocol Version 4, Src: 10.1.21.110, Dst: 10.1.31.130
▼ Transmission Control Protocol, Src Port: 49310, Dst Port: 445, Seq: 1, Len: 0
    Source Port: 49310
    Destination Port: 445
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 0
    Header Length: 40 bytes
  ▼ Flags: 0x004 (RST)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .1.. = Reset: Set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·········R··]
    Window size value: 32120
    [Calculated window size: 32120]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x4b92 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ Options: (20 bytes), Riverbed Transparency, No-Operation (NOP), No-Operation (NOP), No-Operation (NOP), End of Option List (EOL)
```

riverbed 54



HOL1192

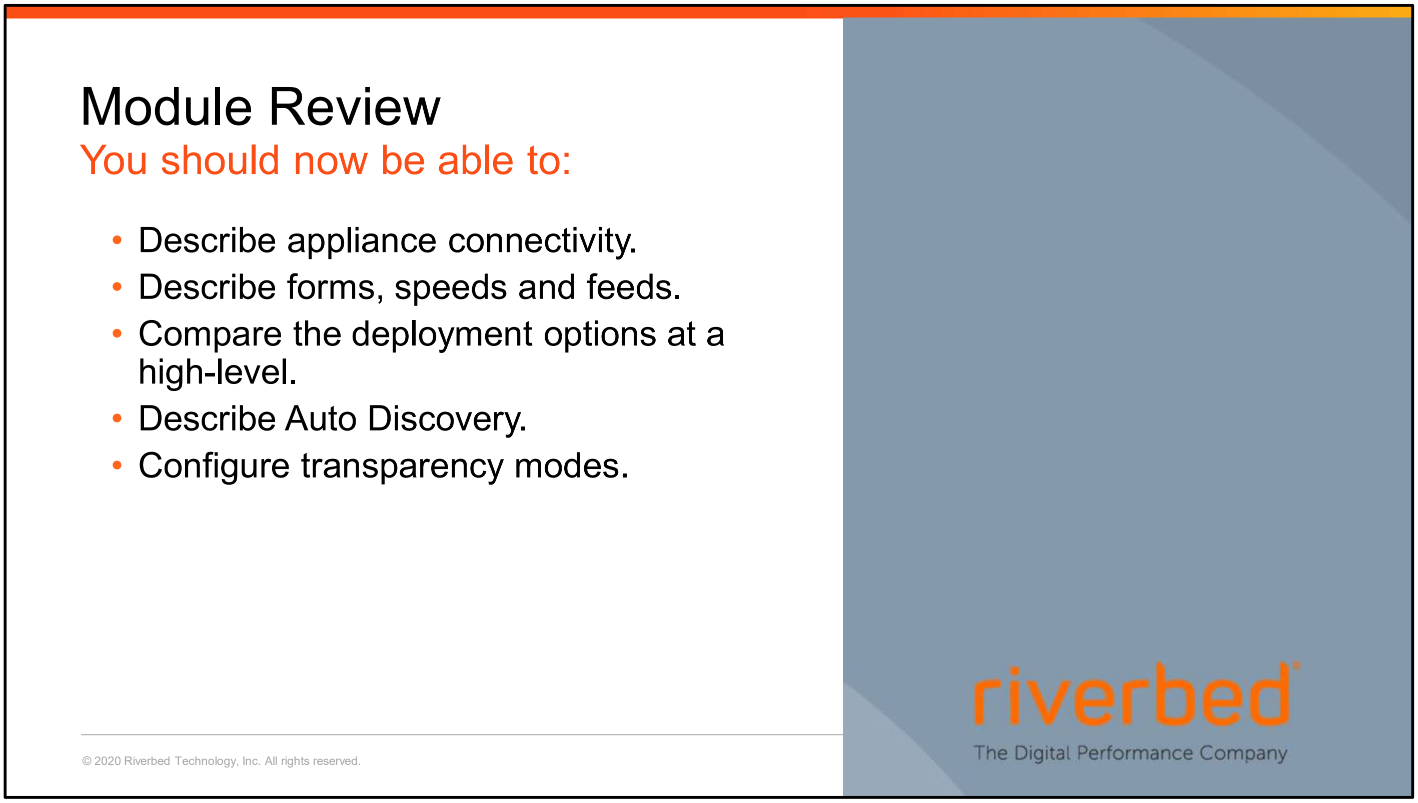
Enhanced Auto Discovery Process

In this lab, you will:

- Analyze Auto Discovery Packets

Duration: **45 minutes**

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



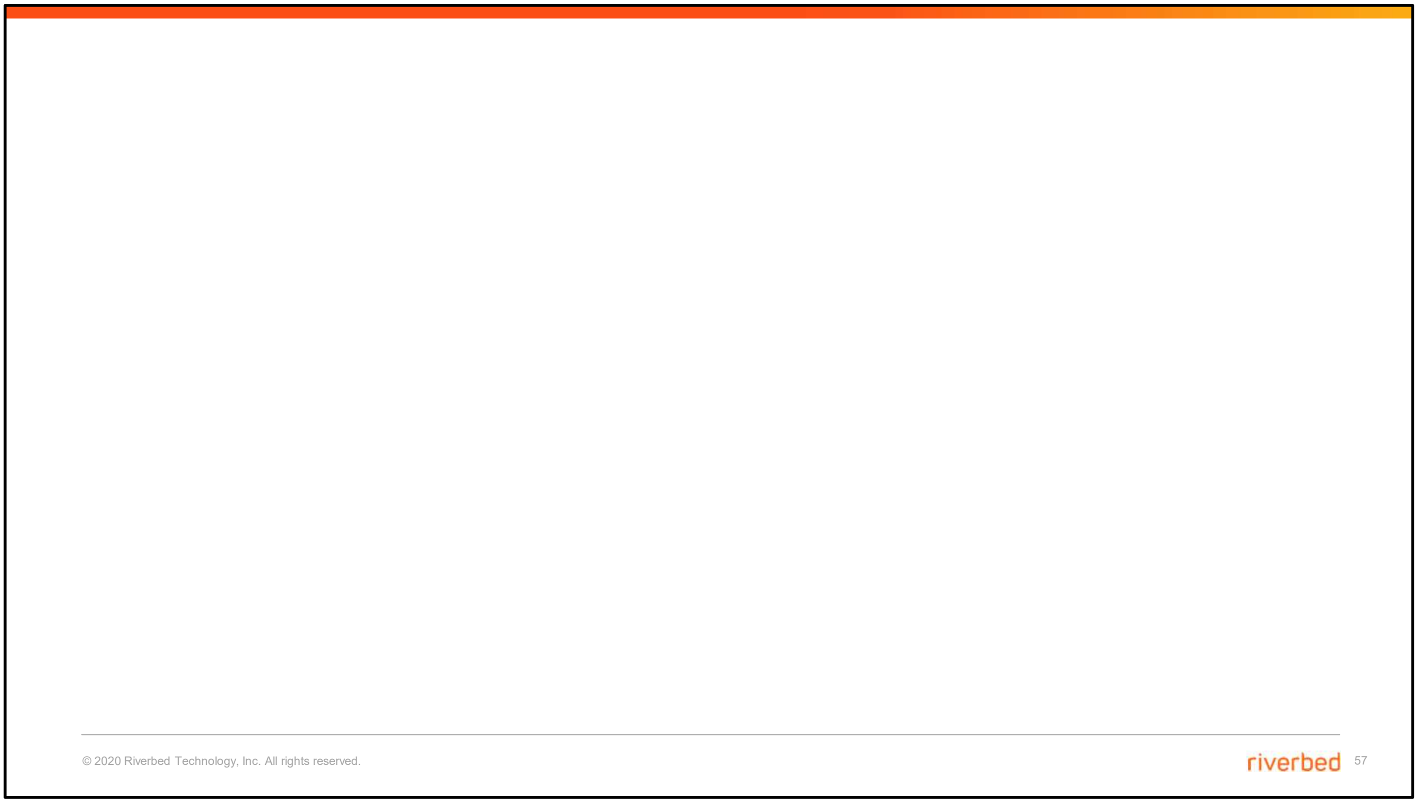
Module Review

You should now be able to:

- Describe appliance connectivity.
- Describe forms, speeds and feeds.
- Compare the deployment options at a high-level.
- Describe Auto Discovery.
- Configure transparency modes.

riverbed

The Digital Performance Company



This slide intentionally left blank