

SteelHead TCP General Setting Service v1.0

Copyright © Riverbed Technology Inc. 2024

Created Jan 16, 2024 at 02:01 PM

Resource: cong_ctrl_options_available

TCP congestion control mode selections: Retrieve applicable congestion control mode selections: When the SkipWare license is present, it returns [{'algo': 'standard', 'for_scps': true}, {'algo': 'auto-detect', 'for_scps': false}, {'algo': 'highspeed', 'for_scps': true}, {'algo': 'bandwidth-estimation', 'for_scps': true}, {'algo': 'rtt-invariant', 'for_scps': false}, {'algo': 'skipware-per-connection', 'for_scps': true}, {'algo': 'skipware-error-tolerant', 'for_scps': true}] When the SkipWare license is absent, it returns [{'algo': 'standard', 'for_scps': true}, {'algo': 'auto-detect', 'for_scps': false}, {'algo': 'highspeed', 'for_scps': true}, {'algo': 'bandwidth-estimation', 'for_scps': true}, {'algo': 'rtt-invariant', 'for_scps': false}]

http://{device}/api/sh.tcp/1.0/cong_ctrl_options_available

JSON

```
{
  "items": [ cong\_ctrl\_option ]
}
```

Property Name	Type	Description	Notes
<i>cong_ctrl_options_available</i>	<object>	TCP congestion control mode selections: Retrieve applicable congestion control mode selections: When the SkipWare license is present, it returns [{'algo': 'standard', 'for_scps': true}, {'algo': 'auto-detect', 'for_scps': false}, {'algo': 'highspeed', 'for_scps': true}, {'algo': 'bandwidth-estimation', 'for_scps': true}, {'algo': 'rtt-invariant', 'for_scps': false}, {'algo': 'skipware-per-connection', 'for_scps': true}, {'algo': 'skipware-error-tolerant', 'for_scps': true}] When the SkipWare license is absent, it returns [{'algo': 'standard', 'for_scps': true}, {'algo': 'auto-detect', 'for_scps': false}, {'algo': 'highspeed', 'for_scps': true}, {'algo': 'bandwidth-estimation', 'for_scps': true}, {'algo': 'rtt-invariant', 'for_scps': false}]	
<i>cong_ctrl_options_available.items</i>	<array of <cong_ctrl_option>>		Optional;
<i>cong_ctrl_options_available.items[items]</i>	<cong_ctrl_option>		

Links

cong_ctrl_options_available: get

GET http://{device}/api/sh.tcp/1.0/cong_ctrl_options_available

Response Body

Returns a [cong_ctrl_options_available](#) data object.

Resource: cong_ctrl

TCP congestion control settings

http://{device}/api/sh.tcp/1.0/cong_ctrl

JSON

```
{
  "mode": cong\_ctrl\_algo
}
```

Property Name	Type	Description	Notes
<i>cong_ctrl</i>	<object>	TCP congestion control settings	

<p><code>cong_ctrl.mode</code></p>	<p><code><cong_ctrl algo></code></p>	<p>'standard' - Standard TCP optimization (RFC compliant): It is a standard-based implementation of TCP(RFC 793) and is the default settings in the SteelHead appliance. 'auto-detect' - TCP automatic detection: Automatically detects the optimal TCP configuration by using the same mode as the peer SteelHead appliance for inner connections. When both appliances select this mode the standard TCP optimization is used. 'highspeed' - HighSpeed TCP optimization: It is efficient in long fat networks(LFNs) with large WAN circuits (50 Mbps and above) over long distances. 'bandwidth-estimation' - TCP bandwidth estimation optimization: It uses an intelligent bandwidth estimation algorithm along with a modified slow-start algorithm to optimize performance in long lossy networks. 'rtt-invariant' - TCP RTT invariant optimization(TCP Hybla): It copes with longer round trip(RTTs) in heterogenous networks which incorporate a terrestrial or satellite radio link. It aims to solve RTT disparity problem by modifying the standard rules for the congestion window increase according to an analytical study. 'skipware-per-connection' - SkipWare per-connection TCP optimization: This feature requires a SkipWare license. It applies TCP congestion control to SCPS-capable connections. This control uses a modified slow-start algorithm and a modified congestion-avoidance approach while remaining reasonably fair and friendly to other traffic. This is a high-performance option for satellite networks. 'skipware-error-tolerant' - SkipWare error tolerant TCP optimization: This feature requires a SkipWare license. It enables SkipWare optimization with the error-rate detection and recovery mechanism on the SteelHead appliance. It allows the per-connection congestion control to tolerate some loss due to corrupted packets without redoing the throughput. This is a high-performance option for lossy satellite networks.</p>	<p>Default is standard; Values: standard, auto-detect, highspeed, bandwidth-estimation, rtt-invariant, skipware-per-connection, skipware-error-tolerant;</p>
------------------------------------	--	--	--

Links

cong_ctrl: get

Retrieve the TCP congestion control setting instance

```
GET http://{device}/api/sh.tcp/1.0/cong_ctrl
```

Response Body

Returns a [cong_ctrl](#) data object.

cong_ctrl: set

Update the TCP congestion control setting instance

```
PUT http://{device}/api/sh.tcp/1.0/cong_ctrl
```

Request Body

Provide a [cong_ctrl](#) data object.

Response Body

Returns a [cong_ctrl](#) data object.

Resource: rate_cap

Impose a global data-transmit limit on the link rate for all SCPS connections between peer SteelHeads, or on the link rate for a SteelHead paired with a third-party device running TCP performance-enhancing proxies (TCP-PEP).

```
http://{device}/api/sh.tcp/1.0/rate_cap
```

JSON

```
{
  "enabled": boolean
}
```

Property Name	Type	Description	Notes
---------------	------	-------------	-------

<code>rate_cap</code>	<code><object></code>	Impose a global data-transmit limit on the link rate for all SPCS connections between peer SteelHeads, or on the link rate for a SteelHead paired with a third-party device running TCP performance-enhancing proxies (TCP-PEP).	
<code>rate_cap.enabled</code>	<code><boolean></code>		Optional;

Links

rate_cap: get

GET `http://{device}/api/sh.tcp/1.0/rate_cap`

Response Body

Returns a [rate_cap](#) data object.

rate_cap: set

PUT `http://{device}/api/sh.tcp/1.0/rate_cap`

Request Body

Provide a [rate_cap](#) data object.

Response Body

Returns a [rate_cap](#) data object.

Resource: satellite_option

Satellite communication option settings

`http://{device}/api/sh.tcp/1.0/satellite_option`

JSON

```
{
  "sei_enabled": boolean,
  "legacy_compression_enabled": boolean
}
```

Property Name	Type	Description	Notes
<code>satellite_option</code>	<code><object></code>	Satellite communication option settings	
<code>satellite_option.sei_enabled</code>	<code><boolean></code>	Enable/Disable transport optimization for single-end interception connections with no SteelHead peer.	Optional; Default is False;
<code>satellite_option.legacy_compression_enabled</code>	<code><boolean></code>	Enable/Disable SkipWare legacy compression	Optional; Default is False;

Links

satellite_option: get

GET `http://{device}/api/sh.tcp/1.0/satellite_option`

Response Body

Returns a [satellite_option](#) data object.

satellite_option: set

PUT `http://{device}/api/sh.tcp/1.0/satellite_option`

Request Body

Provide a [satellite_option](#) data object.

Response Body

Returns a [satellite_option](#) data object.

Type: cong_ctrl_algo

'standard' - Standard TCP optimization (RFC compliant): It is a standard-based implementation of TCP(RFC 793) and is the default settings in the SteelHead appliance. 'auto-detect' - TCP automatic detection: Automatically detects the optimal TCP configuration by using the same mode as the peer SteelHead appliance for inner connections. When both appliances select this mode the standard TCP optimization is used. 'highspeed' - HighSpeed TCP optimization: It is efficient in long fat networks(LFNs) with large WAN circuits (50 Mbps and above) over long distances. 'bandwidth-estimation' - TCP bandwidth estimation optimization: It uses an intelligent bandwidth estimation algorithm along with a modified slow-start algorithm to optimize performance in long lossy networks. 'rtt-invariant' - TCP RTT invariant optimization(TCP Hybla): It copes with longer round trip(RTTs) in heterogenous networks which incorporate a terrestrial or satellite radio link. It aims to solve RTT disparity problem by modifying the standard rules for the congestion window increase according to an analytical study. 'skipware-per-connection' - SkipWare per-connection TCP optimization: This feature requires a SkipWare license. It applies TCP congestion control to SCPS-capable connections. This control uses a modified slow-start algorithm and a modified congestion-avoidance approach while remaining reasonably fair and friendly to other traffic. This is a high-performance option for satellite networks. 'skipware-error-tolerant' - SkipWare error tolerant TCP optimization: This feature requires a SkipWare license. It enables SkipWare optimization with the error-rate detection and recovery mechanism on the SteelHead appliance. It allows the per-connection congestion control to tolerate some loss due to corrupted packets without redoing the throughput. This is a high-performance option for lossy satellite networks.

JSON

```
string
```

Property Name	Type	Description	Notes
<i>cong_ctrl_algo</i>	<i><string></i>	'standard' - Standard TCP optimization (RFC compliant): It is a standard-based implementation of TCP(RFC 793) and is the default settings in the SteelHead appliance. 'auto-detect' - TCP automatic detection: Automatically detects the optimal TCP configuration by using the same mode as the peer SteelHead appliance for inner connections. When both appliances select this mode the standard TCP optimization is used. 'highspeed' - HighSpeed TCP optimization: It is efficient in long fat networks(LFNs) with large WAN circuits (50 Mbps and above) over long distances. 'bandwidth-estimation' - TCP bandwidth estimation optimization: It uses an intelligent bandwidth estimation algorithm along with a modified slow-start algorithm to optimize performance in long lossy networks. 'rtt-invariant' - TCP RTT invariant optimization(TCP Hybla): It copes with longer round trip(RTTs) in heterogenous networks which incorporate a terrestrial or satellite radio link. It aims to solve RTT disparity problem by modifying the standard rules for the congestion window increase according to an analytical study. 'skipware-per-connection' - SkipWare per-connection TCP optimization: This feature requires a SkipWare license. It applies TCP congestion control to SCPS-capable connections. This control uses a modified slow-start algorithm and a modified congestion-avoidance approach while remaining reasonably fair and friendly to other traffic. This is a high-performance option for satellite networks. 'skipware-error-tolerant' - SkipWare error tolerant TCP optimization: This feature requires a SkipWare license. It enables SkipWare optimization with the error-rate detection and recovery mechanism on the SteelHead appliance. It allows the per-connection congestion control to tolerate some loss due to corrupted packets without redoing the throughput. This is a high-performance option for lossy satellite networks.	Default is standard; Values: standard, auto-detect, highspeed, bandwidth-estimation, rtt-invariant, skipware-per-connection, skipware-error-tolerant;

Type: cong_ctrl_option

JSON

```
{
  "algo": string,
  "for_scps": boolean
}
```

Property Name	Type	Description	Notes
---------------	------	-------------	-------

<i>cong_ctrl_option</i>	<object>		
<i>algo</i>	<string>	Congestion Control Algorithm Option	Default is standard; Values: standard, auto-detect, highspeed, bandwidth-estimation, rtt-invariant, skipware-per-connection, skipware-error-tolerant;
<i>cong_ctrl_option.for_scps</i>	<boolean>	This flag indicates an option applicable for SCPS. Currently, the following set of options are applicable: standard, highspeed, bandwidth-estimation, skipware-per-connection, skipware-error-tolerant	Optional;