Technical Guide

Link Health Monitoring for Switches

Feature Overview and Configuration Guide

Introduction

Link health monitoring is a feature that allows a network manager to continually monitor the health of their network. It does this by gathering health metrics comprising latency, jitter, and probe loss on an on-going basis. The health metrics can optionally be recorded in a history buffer for later review and analysis. Link monitoring can also be used with the AlliedWare Plus Trigger facility to automatically change device configuration in response to changes in the health of a monitored link.

Contents

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ switches that support link health monitoring, running version **5.5.1-0.1** or later.

To see whether your product supports link health monitoring, see the following documents:

- The product's Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

Overview of link health monitoring

Link health monitoring on switches can be used to periodically monitor the health of a specific link or routing path over time by sending probes at a fixed interval. By recording the jitter and latency experienced by the probe, as well as the rate of probe loss, the health quality of the link can be determined.

Link health monitoring probes can be sent to fixed IPv4 or IPv6 addresses, or a fully-qualified domain name (FQDN) could be specified as the destination which the device will attempt to resolve to an IP address. Link health monitoring can be configured to use either ICMP echo (ping) packets or HTTP GET requests depending on the type of server the probe is being sent to.

The probe metrics can be recorded into a history buffer at multiple levels of granularity, allowing the network manager to collect performance data about their network that they can analyze to help diagnose network performance issues and outages. Triggers can be associated with link health monitoring probes and set to activate when a link is judged to be good, bad, unreachable or upon any state change, allowing for device configuration to be dynamically modified in response to changing network conditions.

Link health monitoring is similar to and can be considered to be a more advanced version of the Ping Polling feature. Below are some of the advantage that link health monitoring probes offer compared to Ping Polling:

- Latency and jitter information can be used in determining if a link is good or bad, not simply whether the probe times out
- Gathered metric information can be recorded for future review and analysis
- Link health monitoring supports an SNMP read-only MIB, so external network management tools can collect and analyse network performance data
- The interval can be set as low as 100ms between probes
- The destination can be an FQDN rather than a fixed IP address
- Source IP and egress interface can be manually configured
- The DSCP field and size of the ICMP probe can be configured, allowing the QOS configuration within the network to be tested and monitored
- HTTP probes can be used to assess reachability of remote web servers which may have ICMP access denied

One feature that ping polling has that link health monitoring does not, is a user-configurable timeout interval, after which point a probe is considered to be lost; link health monitoring has a fixed interval of 2 seconds after which it will consider a probe to be lost.

Link health monitoring supports stacking, where probes sent and received on any network ports within the stack are managed by the stack master.

On AlliedWare Plus Routers such as the AR4050S, link health monitoring is a key component in conjunction with Policy-Based Routing (PBR) and Deep Packet Inspection (DPI) to create a Software Defined Wide Area Network (SD-WAN) solution. On AlliedWare Plus switch platforms, DPI is not available and link health monitoring does not directly integrate with the hardware PBR feature, so SD-WAN functionality is not supported.

How does link health monitoring work

Link health monitoring works by you configuring a link health monitoring probe. At minimum, this consists of a destination IP address and the interval at which to send the probes. The probes are either ICMP echo requests (pings), or HTTP GET requests for a specific URL.

When the probe is enabled, it will send a probe packet of the selected type at the specified interval to the specified destination IP address. When a successful response is received from the probe's destination, the device calculates the round-trip latency of the message and records this as a successful probe. When a second probe is successful and its latency is calculated, the jitter (difference in latency between responses) can be calculated.

If a response for a probe is not received within 2 seconds, the probe is marked as lost; this is a hard cut-off, and even if the probe subsequently arrives after 2.1 seconds, it will still be marked as lost. The jitter can only be calculated for consecutive successful probe results; if every second probe is counted as lost (perhaps the destination device has an ICMP rate-limit that is discarding the probes), the jitter will be considered to be infinite for metric calculation and comparison purposes.

The current metrics for a probe are visible in the output of the **show linkmon probe** command:

- The **latency** and **jitter** current metrics are calculated based on the sample-size parameter for the probe. For ICMP probes, this defaults to 5 and has a configurable range of 1 to 100. For HTTP probes, this is fixed at 2.
- The purpose of **sample size** is to help smooth out one-off spurious results. For example, if the latency of a probe to a remote site is normally 110ms, a single probe that has a result of 160ms shouldn't by itself be an indication that the latency has worsened. But if probes consistently begin to take longer than 110ms to arrive, then the performance of the link should be considered to have worsened.
- The **loss rate** metric is the percentage of the last (up to) 100 probes that the device has sent where the reply was not received within 2 seconds of the probe being sent. This calculation always uses the 100 most recent probe results, regardless of probe interval. So for an ICMP probe with a configured interval of 10 seconds, it will take 1,000 seconds to send 100 probes. This means the loss rate metric can take quite a while to 'age out' lost probes.
- The **consecutive success/loss** metric is the number of probes in a row that have been successful or lost, up to a maximum of 100.

Optional probe-history collections can be created to store aggregations of the current metrics. The probe-history collections are associated with a probe. They have a number of buckets, which is the amount of historical data to be held in memory. They also have a collection interval, during which any new metrics that are received by link health monitoring are summed together and the result saved into the bucket.

Once all buckets have had data saved into them, the oldest buckets will be re-used and overwritten with the latest data, in the style of a circular buffer. It is possible to configure multiple collection histories and associate them with a single probe.

For example, you could configure 3 histories for a particular probe:

- the first has 3,600 buckets and a 1 second interval
- the second has 10,080 buckets and a 1 minute interval
- the third has 4,320 buckets and a 10 minute interval.

Such histories would be able to store second-by-second averages of the metrics for 1 hour (60 seconds x 60 minutes), minute-by-minute metric averages for 7 days (60 x 24 x 7), and 10-minute averages of the metrics for 30 days (6 x 24 x 30).

The **show linkmon probe-history** command will display a high-level summary of the data captured by the probe history. The full detailed content of the buckets is accessible via the SNMP MIB.

Note: Each bucket uses 28 bytes of memory, so a history collection instance with 65,535 buckets will consume 1.83MB of system memory. Creating too many histories with too many buckets could exhaust the device's available memory and cause it to restart. The memory is reserved when the command is entered.

Link health monitoring triggers can also be configured. These monitor the status of the associated probe, and can be configured to activate the trigger's scripts. Triggers can be set to activate when the associated probe is marked as good, bad, unreachable, or when the state changes between any of these. The states of good, bad, and unreachable are determined by a link health monitoring profile that must be associated with a link health monitoring trigger.

Link health monitoring profiles specify the threshold at which the current metrics for latency and jitter (taking a rolling average using the sample size) are considered to be 'bad' or 'good'. Consecutive probe loss also has a threshold for when a probe should be considered 'unreachable'. The other condition for a probe being considered 'unreachable' is when there is 100% loss rate of probes.

Link health monitoring probes

```
linkmon probe name <NAME> [type <icmp-ping|http-get>]
```

Link health monitoring probe configuration commands and options:

```
destination <A.B.C.D|X:X::X:X|FQDN>
dscp <0-63>
egress interface <interface>
enable
interval <probe-interval>
ip-version <4|6>
sample-size <1-100>
size <64-1500>
source <interface|IPv4-address>
```

A link health monitoring probe is used to determine latency, jitter, and consecutive probe loss for a given destination. The **linkmon probe** command is modal and is used to create and configure the link health monitoring probe options.

The **name** parameter is used to create a name used to identify the probe.

The **type** parameter lets you specify whether you want to create an ICMP or HTTP probe. By default, an ICMP probe will be created. For more information on creating an HTTP probe, refer to "HTTP probes" on page 7.

Once in configuration mode, the following configuration commands can be used:

The **destination** describes the target of the probe. The destination can be an **IPv4** or **IPv6** address, or alternatively, it can be a fully qualified domain name (FQDN). If an **FQDN** is configured, the device will periodically use DNS to resolve the FQDN to an IP address, and send the ICMP probe to this address. This is useful if you want to probe a particular end-point whose IP address is known to change periodically, for example a remote-office site that may not have a fixed IP address from its ISP.

FQDN resolution begins when the probe destination is initially configured, and will recur when the lowest TTL within the returned A/AAAA records has expired.

The **dscp** field allows you to define DSCP field in the IP header of the probes. The DSCP field is optional. By default, probes are sent with a DSCP of “0”.

The **egress** option allows you to configure the interface the probe will be sent out. For IPv4 ICMP probes, only point-to-point interfaces can be specified. IPv6 ICMP probes also support multi-access interfaces.

The **interval** field is used to define the rate at which probes are sent in milliseconds. **interval** is an optional field. By default, probes are sent with an interval of “1000” (one probe per second).

The **ip-version** option is used to specify the IP version used in the probe (IPv4/IPv6).

The **sample-size** field is used to define the sample size used in latency and jitter calculations. **sample-size** is an optional field, with the default value being “5”. The sample size in effect represents the sensitivity of probes to changes in the health metrics of the network. A low sample size means changes in the metrics can quickly propagate through to triggers executing, whereas a larger sample size means changes will propagate more slowly.

The **size** field allows the user to define the packet size of the probes. **size** is an optional field, with probes by default being sent with a size of “100”.

The **source** option allows you specify the source of the probe. Either an interface or an IPv4/IPv6 address can be specified.

The maximum number of link-health monitoring probes that can be configured is restricted to 100.

Note: Link health monitoring probes are disabled by default, and must be enabled per-probe using the **enable** command.

HTTP probes

```
linkmon probe name <NAME> type http-get
```

Link health monitoring probe HTTP configuration commands and options:

```
egress interface <interface>
enable
interval <probe-interval>
ip-version <4|6>
url <url>
```

Link health monitoring also supports HTTP header request probing. These probes are intended for use when you want to monitor the health of a remote HTTP server.

The link health monitoring probe configuration mode takes an optional type parameter. By default, if the type is not specified, it will create an ICMP probe. If you set the type parameter to **http-get** it will create an HTTP probe. All configuration commands relating to HTTP probes are under the HTTP probe configuration mode.

To configure an HTTP probe, use the following commands:

```
linkmon probe name MyProbe type http-get
url http://www.alliedtelesis.com/
enable
```

Because HTTP probes are much more resource intensive for a switch to carry out, as well as to avoid the device being identified as performing a TCP DOS or flood attack, the default interval for HTTP probes is 60 seconds rather than 1 second for ICMP. The minimum configurable interval is 30 seconds rather than 100ms. The maximum configurable interval is 3600 seconds rather than 10 seconds.

HTTP probes consist of performing an HTTP HEAD request to the probe destination. This entails performing a DNS lookup (if required), then opening up a TCP connection to the web-server and performing a HEAD request. The durations of each stage of this process are summed together and use as a metric for latency for accessing the website. Jitter is also recorded; however, given the long interval between probes and the fact that HTTP uses TCP, jitter is not a particularly useful metric for HTTP probing. Profiles can still be configured to take jitter into account with HTTP probes if desired.

It is important to configure an HTTP probe using the **url** command, which specifies the website URL that the HTTP HEAD request will be performed against. The URL must use ASCII characters and conform to the URL syntax in RFC 3986, with HTTP or HTTPS protocol at the start and an optional port number on the end, such as :80, :443 or :8080. Some examples of supported URLs:

- http://www.alliedtelesis.com/
- https://www.facebook.com/
- http://intranet.acme.com:8080

HTTP probes are supported for historical metric gathering.

For example, to create a probe named “probe0” and set the URL to “http://www.alliedtelesis.co.nz/” use the following commands:

```
awplus# config terminal
awplus(config)# linkmon probe name probe0 type http-get
awplus(config-linkmon-http-probe)# url http://www.alliedtelesis.co.nz/
```

Link health monitoring profiles

```
linkmon profile <name>
```

Link health monitoring profile configuration commands:

```
latency bad-above <1-2000>
latency good-below <1-2000>
jitter bad-above <1-1000>
jitter good-below <1-1000>
```

A link health monitoring profile is used to define what are acceptable metrics for a given probe and determine when a trigger should be executed. The **linkmon profile** command is used to create a link health monitoring profile, or to enter link health monitoring profile configuration mode, where the profile metrics can be configured. The **name** parameter is used to create a name used to identify the profile.

The quality of a link is based any combination of the metrics: latency, jitter, and consecutive probe loss:

- **Latency** is defined as the average round trip time for the last x number of probes, where x is the defined sample size (by default 5).
- **Jitter** is the average difference of latency for the last x number of probes, where x is the defined sample size (5 by default). For example, if the last five probes had latency of 5,10,5,10,5 then the difference between the probes is 5,5,5,5 and so the average jitter is (5+5+5+5)/4=5.
- **Consecutive probe loss** behaves a little differently, and is described in the section "Consecutive probe loss" on page 9.

For each of the metrics there are two definable thresholds, the **bad-above** threshold and the **good-below** threshold. If none of the metrics have **bad-above** thresholds defined, and the link health monitoring profile is attached to a link health monitoring trigger, then the trigger will not function as the probe will never be judged as either good or bad.

Link health monitoring triggers are used to associate profiles and probes together, and then take scripted actions when the trigger state is met, as determined by the profile's judgment against the probe's latest health results. If a metric for a probe exceeds any of the defined bad-above thresholds in the link health monitoring profile, then the trigger(s) for that probe are considered bad. If the last x number of packets have been lost, where x is the defined sample size for latency or jitter (by default 5), then the trigger is also considered bad.

The good-below threshold is designed to add some hysteresis to probe health changes. If a good-below value is not defined a trigger is considered good when the current metric drops below the defined bad-above threshold. For example, picture a profile with a link health monitoring bad-above value of 300 and a good-below value of 200. If the probe currently has a latency fluctuating around 305, the trigger will only be marked good when the latency drops below 200. This means that if the latency is fluctuating around 300 then any associated trigger matching on a 'good' status will not execute. For the good-below field to have any functional impact, the bad-above command for the associated metric must also be configured and good-below must be set to a value below the defined bad-above command. If the defined value for the good-below command is greater than the bad-above, then the good-below value is ignored and the system behaves as if only bad-above is configured.

A probe is marked “down” if its egress interface is administratively or electrically disabled, or if it has 100% packet loss.

Consecutive probe loss

Link health monitoring profile configuration commands:

```
consecutive-probe-loss bad-when <1-100>
consecutive-probe-loss good-when <1-100>
consecutive-probe-loss unreachable-when <1-100>
```

There is also an alternative metric that behaves slightly differently, consecutive probe loss:

- **Consecutive probe loss** measures the number of probes that have been lost in a row.

Profiles can specify how many probe packets need to be lost before the trigger will be considered bad or unreachable via the **bad-when** and **unreachable-when** thresholds. The **good-when** threshold applies for when a trigger is already marked as bad or unreachable. It determines how many consecutive probes must succeed before the trigger will be considered good.

Consecutive probe loss replicates the functionality (lost pings) provided by the ping-polling feature as part of the link health monitoring feature, for use with link health monitoring triggers. For more information on link health monitoring triggers, refer to "Link health monitoring triggers" on page 10.

Link health monitoring probe-history

```
linkmon probe-history [<1-65535>] probe <NAME> interval <1-2678400>
buckets <1-65535>
```

This command allows you to create a historic data capture for link health monitoring probes. A custom interval can be configured and a number of buckets, allowing metric data to be flexibly recorded for hours, days, weeks, or months, with resolutions as low as 1 second or as high as days.

Configuring link health monitoring probe history is optional. However, it is useful in diagnosing network outages or poor performance.

Link health monitoring triggers

You can configure triggers that respond to link health monitoring events, produced by a combination of a link health monitoring probe and profile. This configuration acts as a more powerful alternative to ping-poll triggers, since you can use the jitter and latency results from the probe to determine if the path to the destination IP address is bad or good, in addition to determining if the destination is up or down.

This is useful if you want the monitor the health of the link and use trigger scripts to alter the device configuration in response to the link health changing, for example to alter the administrative distance of default routes if a destination becomes unreachable. These triggers require a link health monitoring probe to be configured, as well as a profile that can be used to judge whether the probe results are good, bad, or if the destination is unreachable.

Use the **show linkmon trigger** command to see the currently configured link health monitoring triggers, and the probes and profiles associated with them. To see general trigger configuration information such as the trigger's scripts and activation times, use the **show trigger** command.

Figure 1: Example **show linkmon trigger** output

```
awplus#show linkmon trigger
Trigger 1
-------------------------------------------------------------------------------
Match State:           good
Change Count:          11
Last Change:
  Current State:       bad
  Previous State:      good
  Change Time:         16 Feb 2021 09:26:09
  Cause:               Rx probe 'ftp-server', consecutive probe loss (4>=4)

Probe:                 ftp-server
Enabled:               Yes
Latest Metrics
 Latency             : -
 Jitter              : -
 Loss Rate           : 6.0%
 Consecutive Loss    : 6

Profile:               LAN-resources
  latency bad above                 :  20 ms
  latency good below                :  - ms
  jitter bad above                  :  50 ms
  jitter good below                 :  - ms
  Consecutive success good when     :  3
  Consecutive loss bad when         :  4
  Consecutive loss unreachable when :  -
```

- **match state** indicates under which condition this trigger will execute the associated trigger script.
- **change count** is the number of times the state for the trigger has changed based on the latest probe metrics and configured profile. Note that this is the total number of times the health state of the probe monitored by the trigger has changed, as depending on the configuration not all state changes will result in trigger activations. For that value, see the output of **show trigger**.

- The **last change** information shows the current judgement state for the trigger and previous judgement, the time at which it changed and a short description of why the change occurred.
- The **probe** information shows the configured probe, whether it is enabled and the latest metrics collected for that probe.
- The **profile** information shows the configured profile and the thresholds for good, bad and unreachable judgement within it.

For more information about ping-poll triggers, refer to the Ping Polling Feature Overview and Configuration Guide.

Link health monitoring trigger configuration

The first example below shows how to create a ping-poll trigger that will be triggered when the link health changes.

Figure 2: Example ping-poll trigger configuration

```
ping-poll 1
 ip 192.168.1.1
 up-count 3
 fail-count 4
 active
!
trigger 1
 type ping-poll 1 down
```

The second example shows how to create the equivalent configuration using link health monitoring probes and profiles, and a link health monitoring trigger.

Figure 3: Example link health monitoring trigger configuration

```
linkmon probe name MyProbe type icmp-ping
 destination 192.168.1.1
 interval 1000
 enable
!
linkmon profile MyProfile
 consecutive-probe-loss good-when 3
 consecutive-probe-loss unreachable-when 4
trigger 2
 type linkmon-probe MyProbe MyProfile unreachable
```

Configuration example - link health monitoring

This example configuration shows enabling link health monitoring, configuring a probe and profile, and related triggers.

Figure 4: Example configuration for link health monitoring

```
service linkmon
!
linkmon probe name my-probe type icmp-ping
 destination 183.24.62.5
 interval 5000
 enable
!
linkmon profile unreach
 consecutive-probe-loss good-when 3
 consecutive-probe-loss unreachable-when 4
!
trigger 1
 type linkmon-probe my-probe unreach unreachable
 script 1 failover-on.scp
trigger 2
 type linkmon-probe my-probe unreach good
 script 1 failover-off.scp
```

Show commands

show linkmon probe

Command

```
show linkmon probe
```

This command is used to display output for one or all link health monitoring probes.

Figure 5: Example output for the show linkmon probe command

```
awplus#show linkmon probe
Probe Name              : my-probe
  Status                : enabled
  Type                  : ICMP PING
  IP version            : IPv4
  Destination           : 198.51.100.1
  Egress Int            : -
  Source                : -
  DSCP                  : -
  Packet Size           : 100 bytes
  Interval              : 1000ms
  Sample Size           : 5
Latest Metrics
  Latency               : 1001ms
  Jitter                : 0ms
  Packet Loss           : 0.0%
Probe Details
  Probes Sent           : 3154
  Last Probe Sent       : 23 Mar 2018 03:36:00
  Last Probe Received : 23 Mar 2018 03:36:00
```

Figure 6: Output parameters for the show linkmon probe command

PARAMETER	MEANING
Name	The name of the probe.
Status	Whether the probe is enabled or disabled. If it is enabled, then the device will attempt to send probes if the link is up. If it is disabled, then no probes will be sent.
IP version	The IP version being used, IPv4 or IPv6.
Destination	The destination IP address that the probes are sent to.
Egress Interface	The interface that the probe packets should egress.
Source	The source IP address or interface.
DSCP	The DSCP value to use when sending the packet.
Packet Size	The size of a probe packet.
Interval	The number of milliseconds between sending out each probe.
Sample Size	The number of probe results to use when calculating the latency and jitter metrics.
Latency	The average latency based on the last sample size samples.
Jitter	The average jitter based on the last sample size samples.
Packet Loss	The percentage of packets lost based on the last 100 probes.
Probes Sent	The number of probe packets that have been sent.
Last Probe Sent	The time that the last probe packet was sent.
Last Probe Received	The time that the device last successfully received a probe packet.

show linkmon probe history

Command `show linkmon probe-history [<1-65535>|probe <probe-name>]`

This command is used to display the history for one or all link health monitoring probes.

Figure 7: Example output for the show linkmon probe history command

```
awplus#show linkmon probe-history
------------------------------------------------------------------
ID   Interval (s) Buckets  Latency (ms): Min        Max        Avg
Probe                       Jitter (ms): Min        Max        Avg
                                Packets:  Tx         Rx   Loss (%)
------------------------------------------------------------------
10             1  300/300                 94        105         99
PROBE1                                     2         11          6
                                        2978       2978       0.00
20             5  300/300                 97        102         99
PROBE1                                     4          9          6
                                       14892      14892       0.00
30            10  300/300                 98        101        100
PROBE1                                     5          8          6
                                       29785      29785       0.00
```

Figure 8: Output parameters for the show linkmon probe history command

PARAMETER	MEANING
ID	The ID of the Link Health Monitoring probe-history.
Probe	The name of the probe that this history is for.
Interval	The amount of time between each history sample (in seconds).
Buckets	The total number of samples that are stored.
Latency min	The minimum latency that is in the history.
Latency max	The maximum latency that is in the history.
Latency avg	The average latency of the samples stored in the history.
Jitter min	The minimum jitter that is in the history.
Jitter max	The maximum jitter that is in the history.
Jitter avg	The average jitter of the samples stored in the history.
Packets Tx	The total number of packets transmitted in this history.
Packets Rx	The total number of packets received in this history.
Packets Loss	The percentage of packets lost in the history.

show linkmon trigger

Command

```
show linkmon trigger [<1-250>]
```

This command is used to display the status of one or all link health monitoring triggers.

Figure 9: Example output for the show linkmon trigger command

```
awplus#show linkmon trigger
Trigger 1
------------------------------------------------------------------
Match State:             bad
Change Count:            5
Last Change:
  Current State:         good
  Previous State:        unknown
  Change Time:           09 Feb 2021 21:41:15
  Cause:                 Rx probe 'my_probe', consecutive probe success (3>=3)

Probe:                   my_probe
Enabled:                 Yes
Latest Metrics
 Latency               : 1ms
 Jitter                : 1ms
 Loss Rate             : 22.6%
 Consecutive Success : 7

Profile:                 my_profile
  latency bad above                  :  - ms
  latency good below                 :  - ms
  jitter bad above                   :  - ms
  jitter good below                  :  - ms
  Consecutive success good when      :  3
  Consecutive loss bad when          :  2
  Consecutive loss unreachable when :  4
```

Figure 10: Output parameters for the show linkmon trigger command

PARAMETER	MEANING
Match State	If the state of the probe is the same as the match state, the trigger will activate.
Change Count	The number of time the probe has changed state.
Current State	The current state of the probe.
Previous State	The previous state of the probe.
Change Time	The timestamp when the probe last changed state.
Cause	The reason why the probe last changed state.
Probe	The name of the probe.
Enabled	Whether the probe is enabled.
Latency	The last latency metric of the probe.
Jitter	The last jitter metric of the probe.

PARAMETER	MEANING
Loss Rate	The loss rate of the probe, up to 100 metric results calculated.
Consecutive Success	The number of probes that have consecutively succeeded.
Profile	The name of the profile the probe results are being compared to. The profile determines the acceptable jitter, latency, and probe loss.

Allied Telesis™ NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com