

Smokeping_probes_IRTT(3)

SmokePing

Smokeping_probes_IRTT(3)

NAMESmokeping::probes::IRTT – a SmokePing Probe for IRTT <<https://github.com/peteheist/irtt>>**SYNOPSIS**

```

*** Probes ***

+IRTT

binary = /usr/local/bin/irtt # mandatory
forks = 5
offset = 50%
step = 300
timeout = 15
tmpdir = /tmp/smokeping-irtt

# The following variables can be overridden in each target section
dscp = 46
extraargs = --ttl=32
fill = rand
hmac = opensesame
interval = 1.5
ipversion = 6
length = 172
localaddr = 192.168.1.10:63814
metric = rtt
pings = 5
readfrom = irtt1
readfrompollinterval = 2
serverfill = rand
sleep = 0.5
writeto = irtt1

# [...]

*** Targets ***

probe = IRTT # if this should be the default probe

# [...]

+ mytarget
# probe = IRTT # if the default probe is something else
host = my.host
dscp = 46
extraargs = --ttl=32
fill = rand
hmac = opensesame
interval = 1.5
ipversion = 6
length = 172
localaddr = 192.168.1.10:63814
metric = rtt
pings = 5
readfrom = irtt1
readfrompollinterval = 2
serverfill = rand
sleep = 0.5
writeto = irtt1

```



DESCRIPTION

This SmokePing probe uses IRTT <<https://github.com/peteheist/irtt>> to record network round-trip time <https://en.wikipedia.org/wiki/Round-trip_delay_time>, one-way delay <https://en.wikipedia.org/wiki/End-to-end_delay> or IPDV <https://en.wikipedia.org/wiki/Packet_delay_variation> (jitter), based on the value of the **metric** variable.

Additionally, the probe provides a results sharing feature, which allows using results from a single IRTT run to record multiple metrics for a given host at the same time. One target is defined with the **writeto** variable set, which selects the name of a temporary file to save the IRTT output to. Additional targets are defined with the **readfrom** variable set to the same value, which, instead of running IRTT, wait for the main target's output to become available, then parse it to record the chosen metric from the same data. See the **writeto** and **readfrom** variables for more information.

WARNING

The results sharing feature (**writeto** and **readfrom** variables) requires the number of **forks** for the IRTT probe to be at least the total number of IRTT targets defined (regardless of whether they have **writeto** and **readfrom** set). Otherwise, there can be a deadlock while **readfrom** targets wait for their corresponding **writeto** target to complete, which may never start.

VARIABLES

Supported probe-specific variables:

binary

The location of your irtt binary.

Example value: /usr/local/bin/irtt

Default value: /usr/bin/irtt

This setting is mandatory.

forks

Run this many concurrent processes at maximum

Example value: 5

Default value: 5

offset

If you run many probes concurrently you may want to prevent them from hitting your network all at the same time. Using the probe-specific offset parameter you can change the point in time when each probe will be run. Offset is specified in % of total interval, or alternatively as 'random', and the offset from the 'General' section is used if nothing is specified here. Note that this does NOT influence the rrds itself, it is just a matter of when data acquisition is initiated. (This variable is only applicable if the variable 'concurrentprobes' is set in the 'General' section.)

Example value: 50%

step

Duration of the base interval that this probe should use, if different from the one specified in the 'Database' section. Note that the step in the RRD files is fixed when they are originally generated, and if you change the step parameter afterwards, you'll have to delete the old RRD files or somehow convert them. (This variable is only applicable if the variable 'concurrentprobes' is set in the 'General' section.)

Example value: 300

timeout

How long a single 'ping' takes at maximum

Example value: 15

Default value: 5

tmpdir

A temporary directory in which to place files for writeto/readfrom.

Default value: /tmp/smokeping-irtt



Supported target-specific variables:

dscp

The packet DSCP <https://en.wikipedia.org/wiki/Differentiated_services> value to use (`irtt client --dscp`). This is the same as the classic one byte IP ToS field, but on the modern Internet, typically only the lower 6 bits are used, and this is called the DSCP value. The upper two bits are reserved for ECN <https://en.wikipedia.org/wiki/Explicit_Congestion_Notification>. Hex may be used if prefixed by 0x.

Example value: 46

extraargs

Extra arguments to `irtt client` (see **irtt-client**(1)). **Be careful** with extra arguments, as some can corrupt the results.

Example value: `--ttl=32`

fill The fill to use in the payload for the client to server packet (`irtt client --fill`). The **length** variable must be large enough so there's a payload to fill. Use `rand` for random fill, or see **irtt-client**(1) for more options.

Example value: `rand`

hmac

The HMAC <https://en.wikipedia.org/wiki/Hash-based_message_authentication_code> key to use when sending packets to the server (`irtt client --hmac`).

Example value: `opensesame`

interval

The interval between successive requests, in seconds (`irtt client -i`, but the unit is always seconds (s)).

WARNING

If **interval** is increased to greater than 5 seconds, the **timeout** (which defaults to **pings** * 5 seconds + 1) must be modified so that SmokePing doesn't kill the probe prematurely. Additionally, **interval** must not be increased such that **pings** * **interval** is greater than **step**. For example, at **step**=300 and **pings**=20, the **interval** must not be greater than 15 seconds, but should preferably be less to account for handshake and packet wait times.

Example value: 1.5

Default value: 1

ipversion

The IP version to use for packets (4 or 6, corresponding to `irtt client -4` or `irtt client -6`). By default the IP version is chosen based on the supplied host variable.

Example value: 6

length

The length (size) of the packet (`irtt client -l`). The length includes IRTT headers, but not IP or UDP headers. The actual packet length is increased to accommodate the IRTT headers, if necessary. Header size as of IRTT 0.9.0 as used in SmokePing is 48 bytes when **writeto** is set (since both monotonic and wall clock values are requested) and 40 bytes otherwise.

Example value: 172

localaddr

The local address to bind to when sending packets (`irtt client --local`). See **irtt-client**(1) Host formats for valid syntax.

Example value: 192.168.1.10:63814

metric

The metric to record, one of:

- `rtt`: round-trip time <https://en.wikipedia.org/wiki/Round-trip_delay_time>



- `send`: one-way send delay <https://en.wikipedia.org/wiki/End-to-end_delay> (*requires external time synchronization*)
- `receive`: one-way receive delay <https://en.wikipedia.org/wiki/End-to-end_delay> (*requires external time synchronization*)
- `ipdv`: IPDV <https://en.wikipedia.org/wiki/Packet_delay_variation> (instantaneous packet delay variation, or jitter)
- `send_ipdv`: IPDV for sent packets
- `receive_ipdv`: IPDV for received packets

Note that the `send` and `receive` metrics require accurate external system clock synchronization, otherwise the values from one will be abnormally high and the other will be abnormally low or even negative, in which case the value 0 will be given SmokePing. It is recommended to install `ntp` on both the SmokePing client and IRTT server. Properly configured NTP may be able to synchronize time to within a few milliseconds, which is usually enough to provide useful results. PTP over a LAN may achieve microsecond-level accuracy. For best results between geographically remote hosts, GPS receivers may be used. Since `send_ipdv` and `receive_ipdv` measure the variation in times between successive packets, and since `rtt` and `ipdv` use monotonic clock values on the client side only, external time synchronization is not required for these metrics.

Default value: `rtt`

pings

How many pings should be sent to each target, if different from the global value specified in the Database section. Note that the number of pings in the RRD files is fixed when they are originally generated, and if you change this parameter afterwards, you'll have to delete the old RRD files or somehow convert them.

Example value: 5

readfrom

The name of a file to read results from, instead of running IRTT. Use in combination with **writeto** to use the results from one IRTT run to record multiple metrics. The value will become the name of a file in **tmpdir**, and must be the same as another target's setting for **writeto**. Multiple targets may use the same value for **readfrom**, but **writeto** and **readfrom** may not be both set for a given target. When **readfrom** is set, any variables that affect `irtt client` are ignored because IRTT is not being invoked, including: **dscp**, **extraargs**, **fill**, **hmac**, **interval**, **ipversion**, **length**, **localaddr** and **serverfill**. These values are only relevant in the corresponding **writeto** target.

Note that the **host** variable must still be defined for targets that define **readfrom**, otherwise the target won't be used.

When using this feature, be sure to have at least as many **forks** for the IRTT probe as you have total IRTT targets defined. See the "DESCRIPTION" section for more information.

Example value: `irtt1`

readfrompollinterval

The integer interval in seconds on which to poll for results when **readfrom** is set. Lower numbers will allow **readfrom** to see the results a bit sooner, at the cost of higher CPU usage. Polling does not begin until the soonest time at which the IRTT client could have terminated normally.

Example value: 2

Default value: 5

serverfill

The fill to use in the payload for the server to client packet (`irtt client --sfill`). The **length** variable must be large enough to accommodate a payload. Use `rand` for random fill, or see **irtt-client**(1) for more options.

Example value: `rand`



Smokeping_probes_IRTT(3)

SmokePing

Smokeping_probes_IRTT(3)

sleep

The amount of time to sleep before starting requests or processing results (a float in seconds). This may be used to avoid CPU spikes caused by invoking multiple instances of IRTT at the same time.

Example value: 0.5

writeto

The name of a file to write results to after running IRTT. Use in combination with **readfrom** to use the results from this IRTT run to record multiple metrics. The value will become the name of a file in **tmpdir**, and any targets with **readfrom** set to the same value will use this target's results. There must be only one target with **writeto** set for a given file, and **writeto** and **readfrom** may not be both set for a given target.

When using this feature, be sure to have at least as many **forks** for the IRTT probe as you have total IRTT targets defined. See the “DESCRIPTION” section for more information.

Example value: irtt1

AUTHORS

Pete Heist <pete AT heistp DOT net>

