

# Package ‘pingers’

August 21, 2018

**Type** Package

**Title** Identify, Ping, and Log ISP Connection Data

**Description** The goal of pingers is to assist you with troubleshooting ISP connection issues and assist isolating packet loss. It does this by allowing you to retrieve the top traceroute destinations your ISP uses, and recursively ping each server a series of time capturing the results. Each iteration it goes and queries the destinations again, before shuffling the sequence to ensure the analysis is unbiased and consistent across each trace route.

**Version** 0.1.0

**Maintainer** Jesse Vent <cryptopackage@icloud.com>

**URL** <https://github.com/JesseVent/pinger>

**BugReports** <https://github.com/JesseVent/pinger/issues>

**Depends** R (>= 3.4.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, stringr, tibble, tictoc, tidyselect, data.table

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Author** Jesse Vent [aut, cre]

## R topics documented:

capture_logs . . . . .	2
get_destinations . . . . .	2
ping_capture . . . . .	3
shuffle . . . . .	4
<b>Index</b>	<b>5</b>

---

capture_logs	<i>Capture ISP network logs</i>
--------------	---------------------------------

---

**Description**

Repeat capturing network logs with parameters you specify from [ping\\_capture](#) and [get\\_destinations](#). This will output a csv file with your ping results displaying packet loss and average ping across the defined periods.

**Usage**

```
capture_logs(destinations = 9, pings = 50, log_path = NULL,
             sleep = NULL)
```

**Arguments**

destinations	Retrieve the first n addresses in your ISP destinations
pings	Number of times to ping server
log_path	Optional: The path and filename to save the result set
sleep	Optional: Seconds to sleep for throughout iterations

**Value**

csv file with captured network log information

**Note**

If the log\_path parameter is not provided, it will default to saving a csv file in the current working directory called network\_logs.csv prefixed with the current timestamp in the format ' '

**Examples**

```
## Not run:
capture_logs(destinations = 3, pings = 10, log_path = log, sleep = 20)

## End(Not run)
```

---

get_destinations	<i>Get ISP destinations</i>
------------------	-----------------------------

---

**Description**

Traceroute google and grab the top n servers to assist isolating issues with individual nodes for your ISP.

**Usage**

```
get_destinations(keyword = NULL, top_n = NULL)
```

**Arguments**

keyword	Keyword to search for i.e. 'AAT'
top_n	Retrieve the first n addresses

**Value**

dataframe with server and IP range

**Examples**

```
{  
  dest <- get_destinations(top_n = 1)  
}
```

---

ping_capture	<i>Ping Server</i>
--------------	--------------------

---

**Description**

Ping a server to capture response details

**Usage**

```
ping_capture(server, count)
```

**Arguments**

server	IP address or URL of server
count	Number of times to ping server

**Value**

dataframe with ping results

**Examples**

```
{  
  dest      <- get_destinations(top_n = 1)  
  ping_res <- ping_capture(dest$ip[1], 1)  
}
```

---

`shuffle`*Shuffle dataframe rows randomly*

---

**Description**

Randomly reorder the rows of a dataframe

**Usage**

```
shuffle(data)
```

**Arguments**

`data`                      dataframe to shuffle

**Value**

reordered dataframe

**Examples**

```
{
  ordered_df <- tibble::tibble(V1=1:26,V2=letters)
  shuffled_df <- shuffle(ordered_df)
}
```

# Index

capture\_logs, [2](#)

get\_destinations, [2](#), [2](#)

ping\_capture, [2](#), [3](#)

shuffle, [4](#)