**The sys admin's daily grind**

# Step Counter

**Charly takes the name of the "My Traceroute" tool very literally. The journey is the reward for this alternative TTL-measuring utility.** *By Charly Kühnast*

Been there, done that – there is no need for most network administrators to think twice about the command syntax for the *traceroute* command; the diagnostics tool is something you probably have a gut feeling for, and you can shake it out of your sleeve without too much human latency. On my permanent search for perfection – and because laziness is a virtue in network management – I finally stumbled across My Traceroute, or MTR for short [1]. I immediately liked MTR: it is proud to have reached version number 0.79 after just 13 years of development – no need to rush things, after all.

MTR transmits a bunch of pings (or ICMP echo request packets if you prefer) to the target server, successively increasing the packets' TTL (Time to Live). Packets have a defined lifespan to prevent them from whizzing around the network until the end of time. Any router that transports a packet will decrease its TTL by 1.

## Time to Hop

Once the packet has been decremented to 0, its sender receives an `ICMP Time exceeded` error message for IPv4 or a `Hop count exceeded` for IPv6. The difference is because IPv4 from 1981 [2] measures a packet's trip time in seconds, whereas IPv6 does away with this legacy feature. This also explains why the corresponding data field in the packet header is no longer called TTL, but Hop Limit [3].

The first ICMP packet transmitted by MTR is assigned a TTL of 1, which is immediately decremented to 0 when the packet reaches the next router. The next packet is given a TTL of 2, which takes it one hop further. This game of leap frog continues until the packet reaches the target host.

In contrast to traceroute, which would quit at this point, MTR restarts and continually updates the display. Figure 1 documents the route of a series of packets from Germany to Australia. The target server hosts Sydney's Internet site. The command line for MTR here was as follows::

```
mtr -i5 www.cityofsydney.nsw.gov.au
```

The `-i5` parameter defines an interval of 5 seconds between the individual ICMP packets, although the default is 1 second. The jumps in the latency values between the seventh and eighth hops are quite considerable here. It looks like these two routers are at opposite ends of the undersea cables.

## Round Trip to Australia

If I were mainly interested in the round-trip time for the packets, I would disable name resolution by setting the `-n` parameter. Reporting mode is useful, too. I can

```
mtr -r -c 10 target_address
```

to have the tool run 10 cycles, quit, and summarize. Setting the `-l` switch gives machine-parsable raw data for ongoing processing with Bash or Perl. The `-u` option tells MTR to follow traceroute's example and use UDP datagrams instead of ICMP. This is useful because many routers assign a lower priority to ICMP. ∎∎∎

## ■ INFO

[1] MTR: *http://www.bitwizard.nl/mtr/*

[2] RFC 792: *http://tools.ietf.org/html/rfc0792*

[3] RFC 2460: *http://tools.ietf.org/html/rfc2460*



**Figure 1:** The traceroute-style MTR tool jetting to Australia without bloating its CO2 footprint. The `-i5` parameter sets an interval of 5 seconds between the individual ICMP packets.

## ■ CHARLY KÜHNAST

**Charly Kühnast** is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.