

# Traceface

## traceiface

The `traceiface` tool was presented by John Kristoff at NANOG39. It uses an expanding-ring search technique to find not only the ingress point of each link traversed, but also the respective egress points. Although a nice hack, the utility of this isn't entirely clear. In addition, it is dubious how well this can work over IPv6, unless operators choose to assign link addresses from very small ranges. The prototype implementation is written in Perl based on [Net::Traceroute::PurePerl](#).

## Example Output

Here's some example output from the NANOG39 talk.

```
$ sudo ./traceiface.pl www.nanog.org
1 > 140.192.249.1 0.48 ms
2 > 140.192.9.181 0.52 ms
2 < 140.192.9.177 3.36 ms
3 > 140.192.9.148 0.48 ms
3 < 140.192.9.145 0.77 ms
4 > 140.192.9.154 1.78 ms
4 < 140.192.9.153 0.63 ms
5 > 192.122.182.17 6.81 ms
5 < 192.122.182.18 5.33 ms
6 > 198.108.1.50 7.52 ms
6 < 198.108.1.2 6.97 ms
```

There are two lines for each hop except the first. The `>` line is the "egress" (far end) of the link (actually the ingress interface of the next-hop router), the `<` line is the "ingress" (close end) of the link. Note that the destination (`www.nanog.org`) is the second-to-last line - the last line in the trace is actually the last-hop router to which the destination connects.

## References

- *traceiface*, J. Kristoff, February 2007, NANOG39 Lightning Talk ([PDF](#))
- `_traceiface` Web page, J. Kristoff. Includes source code. <http://condor.depaul.edu/~jkristof/traceiface/>

– Main.SimonLeinen - 07 Feb 2007