

The Story of the PING Program

By Mike Muuss

Yes, it's true! I'm the author of **ping** for UNIX. Ping is a little thousand-line hack that I wrote in an evening which practically *everyone* seems to know about. :-)

I named it after the sound that a sonar makes, inspired by the whole principle of echo-location. In college I'd done a lot of modeling of sonar and radar systems, so the "Cyberspace" analogy seemed very apt. It's exactly the same paradigm applied to a new problem domain: ping uses timed IP/ICMP ECHO_REQUEST and ECHO_REPLY packets to probe the "distance" to the target machine.

My original impetus for writing PING for 4.2a BSD UNIX came from an offhand remark in July 1983 by Dr. Dave Mills while we were attending a DARPA meeting in Norway, in which he described some work that he had done on his "Fuzzball" LSI-11 systems to measure path latency using timed ICMP Echo packets.

In December of 1983 I encountered some odd behavior of the IP network at BRL. Recalling Dr. Mills' comments, I quickly coded up the PING program, which revolved around opening an ICMP style SOCK_RAW AF_INET Berkeley-style socket(). The code compiled just fine, but it didn't work -- there was no kernel support for raw ICMP sockets! Incensed, I coded up the kernel support and had everything working well before sunrise. Not surprisingly, Chuck Kennedy (aka "Kermit") had found and fixed the network hardware before I was able to launch my very first "ping" packet. But I've used it a few times since then. *grin* If I'd known then that it would be my most famous accomplishment in life, I might have worked on it another day or two and added some more options.

The folks at Berkeley eagerly took back my kernel modifications and the PING source code, and it's been a standard part of Berkeley UNIX ever since. Since it's free, it has been ported to many systems since then, including Microsoft Windows95 and WindowsNT. You can identify it by the distinctive messages that it prints, which look like this:

```
PING vapor.arl.army.mil (128.63.240.80): 56 data bytes
64 bytes from 128.63.240.80: icmp_seq=0 time=16 ms
64 bytes from 128.63.240.80: icmp_seq=1 time=9 ms
64 bytes from 128.63.240.80: icmp_seq=2 time=9 ms
64 bytes from 128.63.240.80: icmp_seq=3 time=8 ms
64 bytes from 128.63.240.80: icmp_seq=4 time=8 ms
^C
----vapor.arl.army.mil PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 8/10/16
```

In 1993, ten years after I wrote PING, the USENIX association presented me with a handsome scroll, pronouncing me a Joint recipient of The USENIX Association 1993 **Lifetime Achievement Award** presented to the Computer Systems Research Group, University of California at Berkeley 1979-1993. ``Presented to honor profound intellectual achievement and unparalleled service to our Community. At the behest of CSRG principals we hereby recognize the following individuals and organizations as CSRG participants, contributors and supporters." Wow!

Want to see the source code ? (40k)

From my point of view PING is *not* an acronym standing for Packet InterNet Grouper, it's a sonar analogy. However, I've heard second-hand that Dave Mills offered this expansion of the name, so perhaps we're both right. Sheesh, and I thought the government was bad about expanding acronyms! :-)

The Story of the PING Program

By Mike Muuss

Phil Dykstra added ICMP Record Route support to PING, but in those early days few routers processed them, making this feature almost useless. The limitation on the number of hops that could be recorded in the IP header precluded this from measuring very long paths.

I was insanely jealous when Van Jacobson of LBL used my kernel ICMP support to write TRACEROUTE, by realizing that he could get ICMP Time-to-Live Exceeded messages when pinging by modulating the IP time to life (TTL) field. I wish I had thought of that! :-). Of course, the real traceroute uses UDP datagrams because routers aren't supposed to generate ICMP error messages for ICMP messages.

The best ping story I've ever heard was told to me at a USENIX conference, where a network administrator with an intermittent Ethernet had linked the ping program to his vocoder program, in essence writing:

```
ping goodhost | sed -e 's/./ping/' | vocoder
```

He wired the vocoder's output into his office stereo and turned up the volume as loud as he could stand. The computer sat there shouting "Ping, ping, ping..." once a second, and he wandered through the building wiggling Ethernet connectors until the sound stopped. And that's how he found the intermittent failure.