Clarifying PROBE (rfc8335)

Bill Fenner
IETF 119

Do we need clarifications?

A survey of existing responders and my opinion simply based on the spec

	A	B	D	Me
Extended checksum required?	✅	❌	✅	✅
Packet data echoed?	✅	✅	n/a	❌
Extra data allowed?	✅	✅	❌	❌

Do we need clarifications? (Clients)

A survey of existing clients and my opinion based on the spec

	A	C	Bill
Fill in extended checksum	✅	❌	✅
Include extra data	✅	✅	❌
Rely on responder echoing extra data	✅	✅	❌
Include padding in option length	✅	❌	✅

Do we need clarifications? (Users)

Is this a successful probe request?

```
user@Router> probe a.b.c.d by-name ae1
PROBE a.b.c.d (a.b.c.d):
27 bytes from a.b.c.d: icmp_seq=0 ttl=255 code=2 state=0
active=0 IPv4=0 IPv6=0 time=6.981 ms
27 bytes from a.b.c.d: icmp_seq=1 ttl=255 code=2 state=0
active=0 IPv4=0 IPv6=0 time=4.420 ms
27 bytes from a.b.c.d: icmp_seq=2 ttl=255 code=2 state=0
active=0 IPv4=0 IPv6=0 time=2.530 ms
```

Changes from RFC8335

Updated Figure 1 to reflect the presence of the ICMP Extension Object and additional data.

Updated Section 2 to mention the ICMP Extension Structure checksum, and extra verbosity about how the Extension Structure does not cover the rest of the packet.

Updated Figure 3 to reflect the presence of the ICMP Extension Structure and additional data.

Added a step in Section 4 about copying data from the request to the response.

Added a step in Section 4.1 about validating the ICMP Extension Structure checksum.

Added section Appendix A.1 to suggest human-readable display of PROBE responses

Clarified in Section 2.1 that the length of an ifName Object is adjusted when padding is added.

Non-normative Appendix A.1 text

For the PING application, the primary available piece of information is the fact that we received an ICMP Echo Reply. Therefore, the appropriate information to display is all of the available information about the received reply, e.g., size, ttl, etc. However, with PROBE, the primary piece of information is the reported status of the probed interface: the code, status, A, 4, and 6 fields. It's appropriate to convert the combination of the returned values into a "human-readable" response.

For example, an application may perform these steps:

If the code field is non-zero, print the code value as described in Section 3.

If the code field is zero, then if the L field sent is zero, print the state value as described in Section 3.

Otherwise, the L field sent is 1; print the state represented by the A, 4, and 6 bits. Sample textual translations for these bits are shown in Table 1.

Resources

https://github.com/aristanetworks/probe-tools contains an IPv4 client (to test a responder) and a synthetic response generator (to test a client)

Pull requests #1115 and #1131 at https://github.com/the-tcpdump-group/tcpdump/ implement PROBE printing for IPv4 and IPv6

https://github.com/aristanetworks/iputils contains:

- The list of issues from my review of the original code by Andreas Roeseler and Spencer Lang
- A rebase to the current iputils head of the original work, and fixes to many of the issues

https://fenner.github.io/probe-clarification/ for this document

What’s next?

Before publication as RFC, decide what to do about author count limitation

Interop testing?

WG adoption?