tether.

Grants & Bounties

DHT-RPC and HyperDHT in Wireshark

28/07/2026

peers

tether.

tether.

tether.
Grants & Bounties

Bounty Title

Title	DHT-RPC and HyperDHT in Wireshark
Publish Date	24/07/2026
Submission End Date	
Reward	5’000 USDt

Problem Summary

When debugging networking issues we tend to rely on debug builds, custom internal tools, or my own one-off wireshark build. If we can get our protocols - starting with DHT-RPC / HyperDHT - built into mainline wireshark we can begin pointing engineers and analysts to this more capable off-the-shelf tool.

Impact / Reason Why

Improve time to diagnose network and protocol issues.
Improve the speed and quality of analysis.
Improve user experience with the protocol - It adds value to our product to be able to be analyzed with off the shelf tools.

Scope Items

- Native wireshark dissector for HyperDHT and DHT-RPC. (This is one protocol from wireshark’s point of view - HyperDHT extends DHT-RPC.)
- Support all current Commands: PING, DELAYED_PING, PING_NAT, FIND_NODE, DOWN_HINT, PEER_HANDSHAKE, PEER_HOLEPUNCH, FIND_PEER, LOOKUP, ANNOUNCE, UNANNOUNCE, MUTABLE_PUT, MUTABLE_GET, IMMUTABLE_PUT, IMMUTABLE_GET, and their Response messages.
- Response Fields include the latency from Request to Response. Requests and Responses link to each other’s frames.
- Generate an Expert Item for unrecognized commands and trailing data.
- The response to PING_NAT is sent to the port encoded in the PING_NAT request, and is thus part of a different UDP conversation. The Response should link to the request in the previous conversation.

- A heuristic dissector that reliably identifies the protocol without misidentifying other common protocols.

Scope Exclusions

- Decryption of Encrypted Payloads, such as the Noise Payload and Holepunch Payload
- Verification of cryptographic signatures

Deliverables

- The source code, licensed as GPL-2-or-later. Most likely this will be a single C file. (PR to https://gitlab.com/wireshark/wireshark)
- Read access to the GitLab repository that the PR is submitted from.

Acceptance Criteria / Definition of done

- ☐ The required requests and responses (see: Scope) are fully decoded by the module.
- ☐ The dissector must be a built-in, native module, NOT a plugin or Lua module
- ☐ The source code must adhere to the coding style in the wireshark developer's guide.. https://gitlab.com/wireshark/wireshark/-/raw/master/doc/README.developer.
- ☐ The module must be submitted as a PR to wireshark team according to their process
- ☐ The grantee must incorporate any changes required by the wireshark developers to have the patch accepted into wireshark.

Milestones

ID	Description	Deliverables	Reward
M1	Initial Dissector submitted for PR. It must decode all required request and response messages.	A wireshark build that dissects all commands listed in the Acceptance Criteria. Read access to the GitLab repository the PR is submitted from.	50% of grant

M2	Make changes as required by wireshark maintainers	If the wireshark maintainers require changes, this milestone is completed when the first set of changes are accepted..	30% of grant
M3	PR is accepted.	This milestone is completed when the PR is accepted.	20% of grant

Success Indicators / Key Results

- ☐ Patch is accepted by the wireshark maintainers and merged into mainline wireshark.

Applicants Requirements

- Experience developing software in C
- Familiarity with wireshark
- Familiarity with git
- Familiarity with JavaScript may be helpful. The software that sends and receives these messages is written in JavaScript. Being able to read JavaScript and run NodeJS programs to capture the traffic will be very helpful.

Reward & Payment Schedule

Total: 6,000 USDT, X milestone-based installments:

- **M1**: 2,500 USDt
- **M2**: 1,500 USDt
- **M3**: 1,000 USDt

Each milestone is reviewed within 10 business days. Payment released after reviewer approval. No partial payouts.

Resources & Links

- The DHT-RPC implementation: https://github.com/holepunchto/dht-rpc
- The HyperDHT implementation: https://github.com/holepunchto/hyperdht
- Keet https://keet.io/

- A work in progress dissector that is nevertheless already useful, which may be used as a starting point: https://gitlab.com/jthomas43/wireshark/-/blob/hyperdht-dht-rpc/epan/dissectors/packet-hyperdht.c?ref_type=heads