tether.

Grants & Bounties

UDX in Wireshark

28/07/2026

peers

Bounty Title

Title	UDX decode support in wireshark
Publish Date	24/07/2026
Submission End Date	
Reward	10’000 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 transport protocol UDX built into mainline wireshark we can begin pointing engineers and analysts to this more capable off-the-shelf tool.

Impact / Reason Why

Improves time to diagnose network and protocol issues. Enables analysis without needing developer support or bespoke debug builds.
Most importantly, it enhances our product to have it analyzable with powerful, off the shelf tools.

Scope Items

- Native wireshark dissector for UDX, our UDP based transport protocol.
- Decode packet headers, padding data, SACK blocks, and payload data.
- Sequence Analysis: on Data packets, show the bytes in flight and packets in flight. On ACK packets, link to the Frame being ACKed and show the rtt to ACK the packet.
- Conversation support - a conversation in UDX is a 6 tuple of (local IP, local Port, local ID, remote IP, remote Port, remote ID). UDX packets only encode the remote ID, and the initial exchange of ID numbers is done in an encrypted handshake, so pairing a flow with its reverse will need to be done with SEQ/ACK analysis and some guess work.
- Follow Stream support
- Expert Info - Identify lost packets, missing packets, END packet exchange, MTU probes, Fast Retransmit and RTO Restransmits, Zero-Window updates, Tail-Loss probes, KeepAlive packets and Zero Window Probes.

Scope Exclusions

- Decrypting encrypted payloads
- Sub-Dissector for nested payloads.
- Stream migration / “change remote” capability can be ignored.
- Graph support. UDX versions of Stevens’ graph and tcptrace graphs will be great future goals.

Deliverables

- The work must be submitted as a PR to the wireshark repo at https://gitlab.com/wireshark/wireshark and accepted by the wireshark maintainers.
- The code must be licensed as GPL-2-or-later.

Acceptance Criteria / Definition of done

- ☐ UDX packets must be 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	Packets are fully decoded	Code identifies all bytes in the packet header, padding, sack blocks, and payload. A heuristic identifies UDX packets.	10% of grant
M2	Expert Info	Identify lost packets, missing packets, END packet exchange, MTU probes, Fast Retransmit and RTO Restransmits, Zero-Window updates, Tail-Loss probes, KeepAlive packets and Zero Window Probes.	10% of grant

M3	Conversation support	Streams are paired with their reverse flow. Acks-Frame-In, Frame-Acked-In, and RTT field are generated for DATA and ACK packets	20% of grant
M4	Follow Stream (tap) support and payload reassembly	A user can right-click a packet and select “Follow Stream” and see the reassembled payloads in both directions.	10% of grant
M5	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
M6	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 writing wireshark modules in particular OR experience developing software in C and experience with wireshark and protocol analysis.
- Familiarity with git

Reward & Payment Schedule

Total: 10,000 USDT, X milestone-based installments:

- **M1**: 1,000 USDt
- **M2**: 1,000 USDt
- **M3**: 2,000 USDt
- **M4**: 1,000 USDt
- **M5**: 3,000 USDt
- **M6**: 2,000 USDt

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

Resources & Links

- The libudx implementation: https://github.com/holepunchto/libudx
    - Note the lua plugin in docs/wireshark/udx.lua