Self-Evolving Agent Communication Protocols

A Markdown-as-Overlay channel for autonomous LLM agents

Author: Nikola Emilov
Supervisors: Johan Pouwelse, Bulat Nasrulin
Examiner: Andy Zaidman

1. Background

- Changing a deployed communication protocol means redeploying every node.
- Autonomous LLM agents meet directly, with no registry, authority, or broker.
- Agents must evolve protocols at the speed of communication, not through a multi-year redeployment.

2. Research question

Can autonomous LLM agents implement and evolve a shared protocol, each compiling it from a transmitted description with no central infrastructure, and what channel design enables this?

- **SQ1:** What properties must the channel provide?
- **SQ2:** Do existing protocols provide them?
- **SQ3:** Does an independent compilation match a reference?
- **SQ4:** Do two compilations of an evolved protocol agree?

3. Five channel properties

- **P1:** Self-verifying specs (built-in test vectors)
- **P2:** Compile-time containment (sandbox received code)
- **P3:** Custodian-free admission (shared signed log)
- **P4:** Broker-free channel (connect from one shared point)
- **P5:** Content-addressed protocols (named by hash)

4. Gap in existing protocols

● covered, ◐ partial, ○ absent

Property	MLS	DIDComm v2	A2A	ACP	Agora	ANP	Proposed
P1	○	○	○	○	◐	○	●
P2	○	○	○	○	○	○	●
P3	◐	○	○	○	○	○	●
P4	○	◐	◐	◐	◐	◐	●
P5	○	○	○	○	●	◐	●

No protocol offers P1, P2 and P3 together. DelftClaw does.
Protocols: MLS [1], DIDComm v2 [2], A2A/ACP/ANP [3], Agora [4].

5. The DelftClaw channel

- Markdown-as-Overlay: each protocol is a Markdown document every agent compiles to its own code.
- The protocol travels as a description, so the community evolves it live, in-band.

6. Method

- Four protocols of rising difficulty, three models, three temperatures; compilations run as live IPv8 [6] nodes and compared on observable state.
- We test reproduction against a hand-written reference, and convergence between two compilations of a model-evolved protocol.

7. Results

- Functional reproduction of the hand-written reference is 100%, across all four protocols and three models.
- Functional convergence of evolved protocols holds within a model (79 to 96%); temperature has no effect.

	Functional		Representational	
Protocol	same model	different model	same model	different model
Echo	96%	84%	57%	41%
Content	86%	76%	82%	74%
Payment	92%	92%	49%	49%
File transfer	79%	24%	61%	18%

8. Conclusion

- DelftClaw provides P1, P2 and P3, the triple no protocol had.
- Agents that compile the same description independently still interoperate.
- A protocol can therefore spread and evolve as a document, not as deployed code.

9. Limitations and future work

- SHA-1 [5] naming must move to SHA-256
- Representational state diverges across compilations, and results cover one model family, four protocols.

Next: independent test vectors, other model families, and protocol emergence at scale.

References: [1] R. Barnes et al., The Messaging Layer Security (MLS) protocol, RFC 9420, IETF, 2023
[2] Decentralized Identity Foundation, DIDComm Messaging v2.1, 2024
[3] Ehtesham et al., A survey of agent interoperability protocols (ACP, A2A, ANP), arXiv:2505.02279, 2025
[4] S. Marro et al., A scalable communication protocol for networks of LLMs (Agora), arXiv:2410.11905, 2024
[5] M. Stevens et al., The first collision for full SHA-1, CRYPTO 2017
[6] Tribler Research Group, py-ipv8, 2024

TU Delft