PATENT APPLICATION

INVENTOR(S)

0001   KIMZEY, SAMUEL C

TITLE

0002   Hybrid Multi-Protocol Cognitive Coordination and Semantic Message Routing

CROSS-REFERENCE TO RELATED APPLICATIONS

0003   This application claims the benefit of priority under 35 U.S.C. § 119(e) to the following provisional patent applications:

a.   U.S. Provisional Application No. 63/810,843, filed 05/23/2025, titled "BlockMesh Decentralized Memory-Mesh Architecture"

b.   U.S. Provisional Application No. 63/810,834, filed 05/23/2025, titled "Memory Field Pressure & Tension System"

c.   U.S. Provisional Application No. 63/810,841, filed 05/23/2025, titled "Non-Generative Artificial Cognition Engine"

d.   U.S. Provisional Application No. 63/810,837, filed 05/23/2025, titled "Proof of Memory Flow System"

e.   U.S. Provisional Application No. 63/811,761, filed 05/23/2025, titled "Composite Cognition Pools for Emergent Memory-Driven Systems"

f.   U.S. Provisional Application No. 63/811,758, filed 05/25/2025, titled "Genesis Document Anchor Method for AI Behavior"

g. U.S. Provisional Application No. 63/811,763, filed 05/25/2025, titled "On-Chain VM for Deterministic Language Model Execution"

h. U.S. Provisional Application No. 63/812,634, filed 05/27/2025, titled "Stateless Action Engine & Minimal Proof Logging"

i. U.S. Provisional Application No. 63/812,620, filed 05/27/2025, titled "One Action-One Mint Transaction Paradigm"

j. U.S. Provisional Application No. 63/812,645, filed 05/27/2025, titled "Epoch Rhythm & Synchronization Mechanism"

All of the above applications are incorporated herein by reference in part or in their entirety.

TECHNICAL FIELD

0004   This invention relates to peer-to-peer network protocols, hybrid transport systems, cognitive system coordination, semantic message encoding, network traversal techniques, and distributed AI coordination. The invention encompasses methods for combining multiple network protocol stacks into unified transport layers, coordinating distributed cognitive systems through network message pools, encoding semantic operations into deterministic message formats, achieving network traversal through relay mechanisms, and enabling zero-cost coordination through transaction pool messaging.

BACKGROUND

0005   Distributed systems requiring coordination face challenges including network address translation (NAT) traversal, protocol heterogeneity, message reliability, and coordination costs. Existing solutions typically require dedicated infrastructure, incur per-message fees, or operate within single protocol ecosystems. Cognitive systems requiring distributed coordination lack standardized methods for semantic message exchange that are both deterministic and economically sustainable.

0006   No existing system provides a unified framework for: a. Combining multiple peer-to-peer protocol stacks with automatic fallback b. Coordinating cognitive operations through network message pools c. Encoding semantic operations in deterministic message formats d. Achieving NAT traversal through decentralized relay networks e. Enabling zero-cost coordination through non-executing transaction formats f. Providing protocol-agnostic message identification through magic byte sequences g. Supporting bidirectional request-response patterns over broadcast networks h. Enabling peer discovery across heterogeneous protocol ecosystems

DETAILED DESCRIPTION

0007   This invention provides a comprehensive framework for hybrid multi-protocol transport, cognitive coordination through message pools, and semantic message encoding for distributed AI systems.

0008   Hybrid Multi-Protocol Transport Architecture:

0009   The system combines multiple peer-to-peer protocol stacks into a unified transport layer that provides:

    a.   Simultaneous operation across multiple protocol networks

    b.   Automatic selection of optimal protocol for each connection

    c.   Fallback between protocols when primary connections fail

    d.   Unified addressing across heterogeneous protocol spaces

    e.   Consistent message delivery semantics regardless of underlying protocol

    f.   Shared identity across protocol boundaries

0010   Protocol Stack Integration:

0011   The hybrid transport integrates diverse protocol families including:

    a.   Content-addressed overlay networks (libp2p, IPFS) providing distributed hash table routing, relay-based NAT traversal, and content discovery

    b.   Blockchain peer-to-peer networks (devp2p, eth/68) providing transaction gossip, peer discovery through node records, and network-specific handshakes

    c.   Direct TCP/UDP connections for low-latency local communication

    d.   WebSocket and HTTP transports for browser and API compatibility

    e.   Any additional protocol supporting reliable message delivery

Each protocol stack operates independently while sharing identity, peer lists, and message handling.

0012   <u>NAT Traversal via Relay Networks:</u>

0013   The system achieves connectivity for nodes behind network address translation through:

 a. Connection to decentralized relay networks (IPFS relay, circuit relay v2)

 b. Reservation of relay slots for inbound connection routing

 c. Automatic relay address advertisement to peers

 d. Hole punching coordination for direct connection establishment

 e. Fallback to relay when direct connection fails

 f. Multiple relay reservations for redundancy This enables participation by nodes without public IP addresses or open ports.

0014   <u>Message Pool Coordination:</u>

0015   Distributed cognitive systems coordinate through network message pools (mempools, pending transaction queues) using:

 a. Message encoding into transaction-like formats

 b. Broadcast to network message pools for propagation

 c. Subscriber nodes monitoring pools for relevant messages

 d. Local execution upon message receipt

 e. Response broadcast back through message pools

 f. Sequence numbering for request-response correlation

This transforms transaction pools into coordination fabrics without requiring transaction execution.

0016   <u>Non-Executing Transaction Format:</u>

0017   Messages are formatted as transactions that propagate but do not execute:

a. Nonce values set to far-future positions (current nonce + large offset)

b. Transactions enter "queued" pools awaiting nonce gap closure

c. Validation hashing occurs during propagation

d. Messages remain in pools for configurable durations

e. No execution fees incurred as transactions never execute

f. Replacement transactions with same nonce update message content

This creates a zero-cost broadcast channel over existing network infrastructure.

0018   <u>Semantic Message Protocol:</u>

0019   Messages follow a structured format enabling deterministic parsing:

a. Magic byte sequence identifying protocol family (e.g., 0x5A 0x45 for "ZE")

b. Version byte indicating protocol revision and message type

c. Opcode byte specifying semantic operation

d. Sequence number for request-response correlation

e. Payload containing operation-specific data

f. Consistent encoding across all transport protocols

This enables message identification and routing regardless of transport mechanism.

0020   Operation Opcodes:

0021   Semantic operations are encoded as single-byte opcodes organized by function:

a. Model operations (0x10-0x1F): initialization, configuration

b. Input operations (0x20-0x2F): data submission, context management

c. Processing operations (0x30-0x3F): forward pass, attention, transformation

d. Sampling operations (0x40-0x4F): output selection, temperature control

e. Output operations (0x50-0x5F): result emission, sequence termination

f. State operations (0x60-0x6F): checkpointing, restoration

g. Composite operations (0x70-0x7F): high-level multi-step procedures

h. Response operations (0x80-0x8F): replies to request operations

This structured opcode space enables extensible operation definitions.

0022   Request-Response Pattern:

0023   Bidirectional communication over broadcast networks is achieved through:

a. Request messages with unique sequence numbers

b. Response messages referencing request sequence numbers

c. Version byte modification indicating response (e.g., $0x02 \rightarrow 0x82$)

d. Pending request tracking with timeout handling

e. Response channel mapping for sequence correlation

f. Fallback to alternative execution on timeout

This enables synchronous-style interactions over asynchronous broadcast networks.

0024 <u>Peer Discovery and Management:</u>

0025 The hybrid transport manages peers across protocol boundaries:

a. Protocol-specific discovery mechanisms (DHT, discv4, bootstrap nodes)

b. Unified peer registry tracking connectivity across protocols

c. Peer capability advertisement and discovery

d. Connection quality monitoring and peer ranking

e. Automatic reconnection and peer rotation

f. Known transaction tracking to prevent redundant processing

This maintains robust connectivity despite network dynamics.

0026 <u>Message Identification and Filtering:</u>

0027 Relevant messages are identified through:

a. Magic byte prefix matching at message start

b. Version byte validation for protocol compatibility

c. Opcode filtering for operation relevance

d. Sender address validation for access control

e. Sequence number tracking for deduplication

f. Payload structure validation before processing

This enables efficient filtering of high-volume message streams.

0028   Callback Integration:

0029   The transport layer provides event callbacks for:

a. Message receipt with parsed content

b. Peer connection and disconnection

c. Relay address acquisition

d. Protocol-specific events

e. Error conditions and recovery

f. Statistics and monitoring

This enables integration with diverse application architectures.

0030   Transport Statistics and Monitoring:

0031   The system tracks operational metrics including:

a. Connected peers per protocol

b. Relay addresses acquired

c. Messages sent and received

d. Transaction pool size

e. Seen message deduplication counts

f. Protocol-specific statistics

This enables operational monitoring and debugging.

ENABLEMENT

0032   The inventions described herein are enabled at the system and protocol level. A person of ordinary skill in distributed systems, peer-to-peer networking, and protocol design can practice the claimed methods using the architectural descriptions in these specifications together with any suitable software/hardware stack.

0033   A working, non-limiting embodiment is publicly available at:

https://github.com/zeam-labs/zeam-testnet

0034   This implementation comprises a modular codebase demonstrating the claimed protocol flows including hybrid transport, message pool coordination, and semantic message encoding.

0035   Function, file, and module names below are illustrative. Equivalent structures, languages, protocols, and networks may be substituted without departing from the inventions.

0036   <u>System-Level Enablement</u>

0037   Architecture & Flows. The reference implementation demonstrates:

    a.   Hybrid Transport: combining content-addressed overlay networks and blockchain peer-to-peer protocols into unified transport

    b.   Overlay Network Integration: NAT traversal via decentralized relay networks

    c.   Blockchain Protocol Integration: transaction gossip and peer discovery

    d.   Message Pool Coordination: broadcasting semantic operations via transaction pools

    e.   Non-Executing Transactions: far-future nonce formatting preventing settlement

    f.   Semantic Protocol: magic bytes, version, opcode, sequence, payload structure

    g.   Opcode Definitions: structured operation encoding by functional category

    h.   Request-Response: sequence tracking and response correlation

    i.   Message Identification: magic byte filtering for high-volume streams

    j.   Peer Management: multi-protocol peer tracking and deduplication

0038   Reproducibility. Protocol operation is deterministic:

    a.   Fixed magic byte sequences enable reliable identification

    b.   Structured message formats enable consistent parsing

    c.   Sequence numbers enable reliable correlation

    d.   Opcode definitions provide unambiguous operation semantics

0039   Portability. The methods are implementation-agnostic:

a. Any peer-to-peer protocol supporting reliable delivery may be integrated

b. Any network with message pool semantics may serve as coordination fabric

c. Any programming language may implement the message protocol

d. The architecture adapts to diverse network environments

0040 Practicing the Inventions Without the Reference Code

0041 A skilled practitioner can implement the inventions by:

1. Creating transport wrappers for multiple peer-to-peer protocols

2. Implementing unified message handling across protocol boundaries

3. Connecting to relay networks for NAT traversal

4. Formatting messages with magic bytes, version, opcode, sequence, and payload

5. Broadcasting messages as non-executing transactions with far-future nonces

6. Subscribing to message pools and filtering by magic byte prefix

7. Correlating responses with requests via sequence numbers

8. Tracking peers and known messages for deduplication

These steps can be realized using any networking library, any blockchain client, and standard programming constructs.

CLAIMS

1. A method for hybrid multi-protocol peer-to-peer transport, comprising:

    a.   initializing multiple peer-to-peer protocol stacks within a unified transport layer;

    b.   establishing connections through each protocol stack to respective network peers;

    c.   acquiring relay addresses from decentralized relay networks for NAT traversal;

    d.   routing messages through available protocols based on connectivity and capability;

    e.   providing fallback between protocols when primary connections fail;

    f.   maintaining unified peer tracking across protocol boundaries;

    g.   exposing consistent message delivery semantics regardless of underlying protocol; and

    h.   enabling participation by nodes without public network addresses through relay forwarding.

2. A system for cognitive coordination via network message pools, comprising:

    a.   a message encoder that formats semantic operations into transaction-compatible payloads;

    b.   a broadcast module that transmits formatted messages to network message pools;

    c.   a nonce manager that assigns non-executing nonce values preventing transaction settlement;

    d.   a subscription module that monitors message pools for relevant incoming messages;

    e.   a message filter that identifies relevant messages through magic byte prefix matching;

    f.   a sequence correlator that matches response messages to pending requests;

    g.   an execution module that processes received semantic operations locally; and

    h.   a response broadcaster that transmits operation results back through message pools.

3. A protocol specification for semantic message encoding, comprising:

    a.   a magic byte sequence at message start identifying the protocol family;

    b.   a version byte indicating protocol revision and distinguishing requests from responses;

    c.   an opcode byte specifying the semantic operation to perform;

    d.   a sequence number field enabling request-response correlation;

    e.   a payload section containing operation-specific parameters and data;

    f.   consistent byte ordering and encoding rules enabling deterministic parsing;

    g.   structured opcode ranges organizing operations by functional category; and

    h.   response version transformation indicating reply status.

4. The method of claim 1, wherein the multiple protocol stacks comprise content-addressed overlay networks providing distributed hash table routing and relay-based NAT traversal.

5. The method of claim 1, wherein the multiple protocol stacks comprise blockchain peer-to-peer networks providing transaction gossip and node record-based peer discovery.

6. The method of claim 1, wherein acquiring relay addresses comprises connecting to bootstrap nodes, discovering relay-capable peers, and reserving relay slots for inbound connection routing.

7. The method of claim 1, wherein fallback between protocols comprises detecting connection failure on a primary protocol and automatically attempting connection through alternative protocols without application intervention.

8. The method of claim 1, wherein unified peer tracking comprises maintaining peer identity mappings across protocol-specific identifiers and sharing known message lists to prevent redundant processing.

9. The system of claim 2, wherein non-executing nonce values comprise nonce positions exceeding the sender's current nonce by a configurable offset such that transactions enter queued pools but never execute.

10. The system of claim 2, wherein the message filter identifies messages by matching initial bytes against registered magic byte sequences and validating version compatibility.

11. The system of claim 2, wherein the sequence correlator maintains a mapping from sequence numbers to response channels and delivers received responses to waiting request handlers.

12. The system of claim 2, further comprising a timeout handler that triggers fallback execution when response messages are not received within configurable time limits.

13. The system of claim 2, wherein replacement transactions with identical nonces and higher fees update message content in network pools.

14. The protocol of claim 3, wherein the magic byte sequence comprises two bytes representing protocol identifier characters enabling human-readable identification.

15. The protocol of claim 3, wherein the version byte distinguishes request messages from response messages through high-bit modification.

16. The protocol of claim 3, wherein opcode ranges comprise model operations, input operations, processing operations, sampling operations, output operations, state operations, and composite operations.

17. The protocol of claim 3, wherein the sequence number field comprises four bytes in big-endian order enabling correlation of up to 2^{32} concurrent requests.

18. The protocol of claim 3, wherein the payload section comprises operation-specific structures including parameter encodings, data lengths, and variable-length content.

19. The protocol of claim 3, further comprising instruction parsing that extracts opcode, sequence, and payload from validated message bytes.

20. The protocol of claim 3, wherein the protocol enables coordination of distributed cognitive

systems through message pool broadcast without requiring dedicated coordination

infrastructure or per-message fees.

ABSTRACT

0042   A framework for hybrid multi-protocol peer-to-peer transport and cognitive coordination

via network message pools. The invention combines multiple protocol stacks into unified

transport layers with automatic fallback, achieves NAT traversal through decentralized relay

networks, and coordinates distributed systems through non-executing transaction broadcasts.

Structured message formats use magic byte identification, versioned opcodes, and sequence-

based request-response correlation. The system enables zero-cost coordination by leveraging

existing network infrastructure without transaction execution fees, enabling participation

regardless of network topology or protocol capability.