Neural Semantic Communication Between Semantic-Cognitive Virtual Machines in Hyper-Aether

Detailed Architecture and Component Explanations

Hirofumi Inomata

Abstract

This paper proposes a semantic-native communication architecture for distributed semantic-cognitive virtual machines (VMs) within the Hyper-Aether framework.

Unlike conventional systems where communication transfers packets, symbols, or byte streams, Hyper-Aether transfers semantic-cognitive states.

The architecture introduces:

- semantic-cognitive virtual machines,
- neural semantic communication,
- neural serializer/deserializer communication,
- semantic routing,
- semantic memory,
- semantic transactions,
- and distributed cognition synchronization.

This paper additionally provides detailed explanations for each architectural component and explains how cognition, communication, memory, and learning interact inside Hyper-Aether.

1 Introduction

Modern distributed systems fundamentally assume:

$$communication = data\ transfer$$

However intelligent systems require:

$$meaning = f(data, context, memory, intention)$$

Therefore:

$$communication \neq meaning$$

Hyper-Aether proposes:

$$communication = semantic\ state\ synchronization$$

In this architecture:

- VMs behave as semantic-cognitive entities,
- communication transfers latent semantic states,

- routing is driven by semantic similarity,
- memory forms a semantic topology,
- and intelligence emerges from distributed semantic interaction.

2 Overall Architecture

The complete Hyper-Aether system is represented as:

$$QH = (C, V, R, M, F, H)$$

where:

- C : cognitive field,
- V : VM network,
- R : semantic routing fabric,
- M : semantic memory system,
- F : communication fabric,
- H : semantic Hilbert space.

The architecture integrates:

- cognition,
- communication,
- memory,
- routing,
- learning,
- and semantic synchronization

into a unified semantic operating system.

3 Semantic-Cognitive Virtual Machines

Each VM is modeled as:

$$VM_i = (S_i, N_i, I_i, M_i, R_i, F_i)$$

where:

- S_i : semantic state,
- N_i : neural engine,
- I_i : communication interface,
- M_i : semantic memory,

- R_i : routing intelligence,
- F_i : learning/adaptation system.

Unlike conventional VMs, these entities:

- understand meaning,
- communicate semantically,
- learn continuously,
- and cooperate collectively.

4 Detailed VM Internal Architecture

4.1 Input Interface Layer

The input interface receives:

- semantic messages,
- multimodal inputs,
- sensor streams,
- serialized neural states,
- and external events.

Its role is to transform incoming signals into internal semantic representations.
Input encoding is represented as:

$$z_i = E(x_i, c_i)$$

where:

- x_i : input signal,
- c_i : context,
- E : semantic encoder.

4.2 Semantic Memory Layer

The semantic memory stores:

- graph memory,
- vector memory,
- episodic memory,
- associative memory,
- semantic history.

Unlike conventional storage systems, memory is organized by semantic similarity. Memory forms a semantic topology:

$$Memory = (K, E)$$

where:

- K : semantic knowledge nodes,
- E : semantic relationships.

4.3 Neural Engine

The neural engine executes:

- reasoning,
- generation,
- prediction,
- planning,
- and semantic interpretation.

The neural state is represented as:

$$h_i = N_i(z_i, s_i, m_i)$$

where:

- z_i : encoded input,
- s_i : semantic state,
- m_i : memory state.

Large Language Models (LLMs) typically function inside this layer.

4.4 Communication Interface

The communication interface performs:

- semantic messaging,
- context propagation,
- latent-state serialization,
- semantic synchronization,
- semantic feedback exchange.

This layer is implemented using:

$$Neural\ Serializer/Deserializer$$

architectures.

4.5 Routing Intelligence

The routing system determines:

- which VM should receive messages,
- semantic multicast targets,
- resonance clusters,
- semantic affinity groups.

Routing probability is:

$$P_{ij}(t) = \sigma(Q_i K_j^T + B_{ij}) e^{-D_H(i,j)}$$

where:

- $D_H(i, j)$: semantic distance,
- Q_i : semantic query,
- K_j : semantic key.

4.6 Learning and Adaptation Layer

This layer performs:

- online learning,
- reinforcement learning,
- adaptation,
- self-optimization,
- semantic correction.

Learning updates:

$$\theta_{t+1} = \theta_t - \eta \nabla L$$

where:

- θ : model parameters,
- L : semantic loss,
- η : learning rate.

5 Neural Semantic Communication

5.1 Concept

Traditional systems transfer:

$$bytes$$

Hyper-Aether transfers:

$$semantic\ cognitive\ states$$

The communication unit becomes:

$$message = (payload, context, memory, attention, feedback)$$

5.2 Communication Flow

The complete communication flow is:

1. Neural state generation,
2. Semantic serialization,
3. Serial semantic transmission,
4. Deserialization,
5. Semantic reconstruction,
6. Semantic alignment evaluation.

6 Neural Serializer / Deserializer

6.1 Purpose

Neural networks internally operate on:

$$\textit{high dimensional parallel states}$$

However communication channels are limited.
Thus:

$$\textit{parallel state} \rightarrow \textit{serial stream}$$

conversion is required.

6.2 Serialization

The sender state:

$$h_A \in \mathbb{R}^N$$

is serialized as:

$$S(h_A) \rightarrow \{s_t\}$$

where:

$$\{s_t\}$$

is a semantic time-series stream.

6.3 Deserialization

The receiver reconstructs:

$$D(\{s_t\}) \rightarrow h_B$$

where:

$$h_B \approx h_A$$

Importantly:

$$\textit{semantic equivalence}$$

is more important than bit-perfect equality.

7 Context Propagation

7.1 Importance of Context

Meaning depends on context.

For example:

- "apple" may represent:
  - fruit,
  - company,
  - symbol,
  - memory.

Thus payload-only communication is insufficient.

7.2 Types of Context

Hyper-Aether propagates:

- semantic context,
- temporal context,
- memory context,
- routing context,
- intention,
- attention states,
- importance weights.

8 Semantic Memory System

The memory system integrates:

- graph memory,
- vector memory,
- episodic memory,
- semantic indexing.

Memory retrieval is based on:

$$\textit{semantic similarity}$$

rather than addresses.

This enables:

- associative recall,
- semantic navigation,
- memory resonance,
- contextual retrieval.

9 Semantic Transactions

9.1 Traditional Transactions

Conventional systems use:

- ACID properties,
- packet acknowledgement,
- byte consistency.

9.2 Semantic Transactions

Hyper-Aether introduces:

$$Semantic\ Transaction$$

defined as:

$$T_s = (S_i, C_i, M_i, R_i)$$

where:

- S_i : semantic payload,
- C_i : context,
- M_i : memory association,
- R_i : reinforcement feedback.

9.3 Semantic Commit

Commit occurs when:

$$D_H(A, B) < \epsilon$$

where:

- D_H : semantic distance,
- ϵ : semantic threshold.

Thus:

$$commit = semantic\ convergence$$

9.4 Semantic Rollback

Rollback occurs when:

$$semantic\ divergence$$

is detected.
Recovery mechanisms include:

- context resend,

- attention correction,
- semantic reinforcement,
- memory synchronization.

10 Distributed Cognition

VMs collectively form:

distributed cognition

rather than isolated processes.
VMs exchange:

- latent states,
- intermediate reasoning,
- semantic memories,
- contextual states.

This enables:

- collaborative intelligence,
- emergent reasoning,
- distributed planning,
- collective adaptation.

11 LLM Integration

LLMs function inside:

- the neural engine,
- semantic reasoning layers,
- semantic communication layers.

Possible deployment models include:

- local per-VM LLM,
- shared semantic LLM services,
- hierarchical LLM systems,
- swarm LLM architectures.

LLMs provide:

- language understanding,
- reasoning,
- planning,
- tool usage,
- semantic interpretation.

12 Hardware Architecture

Hyper-Aether hardware consists of:

- neural compute tiles,
- semantic memory fabrics,
- Neural SerDes engines,
- semantic routing accelerators,
- optical semantic interconnects,
- distributed vector memory.

Communication may use:

- optical links,
- CXL,
- photonic fabrics,
- semantic-aware SmartNICs.

13 Advantages

The architecture provides:

- semantic-native communication,
- distributed cognition,
- semantic transactions,
- context-aware routing,
- latent-state synchronization,
- semantic memory organization,
- collective intelligence.

14 Challenges

Major challenges include:

- semantic distance measurement,
- synchronization scalability,
- semantic drift,
- hardware complexity,
- semantic verification,
- distributed stability.

15 Conclusion

This paper proposed a semantic-native communication framework for Hyper-Aether semantic-cognitive virtual machines.

The architecture integrates:

- cognition,
- communication,
- memory,
- routing,
- learning,
- and semantic synchronization

into a unified semantic operating system.

Unlike conventional distributed systems, Hyper-Aether transfers:

$$\textit{semantic cognitive states}$$

rather than packets or bytes.

Communication becomes:

$$\textit{semantic resonance}$$

and transactions become:

$$\textit{semantic convergence}$$

This framework provides a possible foundation for:

- AI-native operating systems,
- distributed AGI infrastructures,
- semantic computing platforms,
- and semantic cognition networks.

References

[1] John von Neumann, *Mathematical Foundations of Quantum Mechanics*, Princeton University Press, 1955.

[2] Claude Shannon, *A Mathematical Theory of Communication*, Bell System Technical Journal, 1948.

[3] Ashish Vaswani et al., *Attention Is All You Need*, NeurIPS, 2017.

[4] Richard Feynman, *Space-Time Approach to Quantum Mechanics*, 1948.

[5] Hirofumi Inomata, *Quantum Hyper-Aether: Semantic Geometry and Distributed Cognitive Virtual Machines*, 2026.