

Tagma-WAV

The Coordinate Is the Message

SSCCS Foundation

`ssccs.org`

July, 2026

Abstract

Tagma-WAV defines the signal-domain utilization¹ of the Tagma coordinate primitive: how coordinates are encoded, validated, and interpreted as signals over existing physical channels. It is a signal-delivery protocol, not a transmission system: the transmitter, antenna, modulation waveform, and medium are delegated to the underlying channel (BLE, LoRa, UHF, optical, acoustic). The central claim is that packet-based communication can be replaced by coordinate-based signaling. Every message is a coordinate path², a sequence of N syllables that carries both the address and the payload in a single structural unit, so there is no packet header, no checksum, no protocol stack, and no parser on the receiving end. The coordinate itself is the message.

The transmission-concept part of the persistent IO layer for spatial computing specifies the wave-observation pattern, in which a broadcast coordinate is observed by unspecified receivers, each acting according to its own local state, so response diversity is produced entirely on the receiver side. The framework covers emission and observation only; interpretation and action are application-level and deliberately out of scope. The coordinate space is one: the same coordinate path that keys a Tagma-KV entry is transmitted as a wave sequence and, once observed, becomes a fact in the neXus FIH blackboard, where the Actus orchestration runtime³ owns interpretation and action.

The underlying transmission medium is decoupled from the protocol: any physical carrier that can convey a 16-bit value (radio, light, sound) suffices. The receiving side validates each coordinate structurally (54,364 invalid states are immediately rejected) and reconstructs the coordinate path at the cost of a single lookup, approximately 0.39 ns in native mode, with zero protocol overhead.

This specification does not replace cryptographic primitives. SHA-256 remains for signatures and authentication. What is replaced is the structural use of packets, headers, and parsers, in favor of deterministic coordinate arithmetic.

This work is a theoretical preprint published before implementation. The digital-path numbers are simulation measurements; physical-layer costs are media-dependent and are stated as model parameters, not measured results. Where a claim depends on an implementation choice not yet made, the design space is stated explicitly.

1 The Problem: Packet-Based Communication Is a Structural Tax

Every digital communication system today transmits packets. A packet is a container: it wraps a payload with headers, checksums, sequence numbers, and protocol identifiers. The receiving side must parse this container, validate its integrity, extract the payload, and then interpret the payload according to its own schema.

¹Tagma-WAV is part of the synTagma project under the SSCCS Foundation. <https://docs.sccs.org/projects/syntagma/>

²“Coordinate path” is the abstract term used in this document. The reference implementation names the type `CoordPath` (tagma-core); the abstract term is used because the concept — a sequence of N structural coordinates — is independent of any implementation type.

³Actus is the agent orchestration runtime of the SSCCS Foundation, operating across the neXus shared knowledge space: <https://docs.sccs.org/projects/actus/>

This process is universal, and it carries a universal cost:

Cost Component	Typical Overhead
Header parsing	10–100 ns per layer
Checksum validation	5–50 ns per packet
Protocol stack traversal	100–1000 ns per packet
Payload extraction	10–100 ns per packet
Parser implementation	Ongoing engineering cost

The industry has responded with faster NICs, smarter offload engines, and more efficient protocols. But the root cause remains untouched: communication is still container-based, not structure-based. Every packet must be unpacked, and every unpack operation is a tax on every transmission.

The container is eliminated entirely. The message is the coordinate; the coordinate is the message. There is no unpacking because there is no packing.

2 The Alternative: Structural Coordinate Transmission

Coordinate-based transmission replaces packet-based transmission. A message is encoded as a coordinate path — a sequence of N syllables drawn from the Unicode Hangul block (U+AC00–U+D7A3). Each syllable encodes a 16-bit coordinate through its initial, medial, and final decomposition.

The transmission process is:

1. **Message → coordinate path:** The message is mapped to a sequence of coordinates using the same deterministic transformation as Tagma-KV. Each syllable carries $\log_2(11,172) \approx 13.45$ bits of information, so the mapping is injective for messages up to $13.45N$ bits ($\approx 1.68N$ bytes) — the information-theoretic bound of an N -syllable coordinate path.
2. **coordinate path → Physical Signal:** Each 16-bit coordinate is transmitted directly over the physical medium. No headers are added. No checksums are computed. The coordinate itself is the unit of transmission.
3. **Physical Signal → coordinate path:** The 16-bit values are read from the medium.
4. **coordinate path → Message:** The receiving side validates each coordinate structurally (range check: U+AC00–U+D7A3) and reconstructs the coordinate path.

The transformation is:

- **Deterministic:** same message maps to the same coordinate path
- **Collision-free:** coordinate path is injective for messages up to the $13.45N$ -bit bound; beyond it, injectivity cannot hold (pigeonhole), and the mapping must fall back to collision-resistant hashing — a specification boundary, not a property of all message lengths
- **Self-validating:** $65,536 - 11,172 = 54,364$ invalid 16-bit values are immediately rejected
- **Parser-free:** the coordinate path is the message; no extraction is needed

Two framing conventions follow from the variable-length property. Either N is fixed per command class (the receiver knows how many syllables to expect), or a length indicator is required — in which case that indicator is the only framing element in the protocol. The parser-free property holds strictly in the fixed- N convention.

3 Transmission Implementations

Tagma-WAV is the protocol family over wave carriers: a coordinate is transmitted as a 16-bit symbol over any physical wave that can carry it. Tagma-RF, Tagma-Light, and Tagma-Sound are instances of the same protocol for specific carriers; sound is the most accessible instance, radio the defining one.

3.1 RF: Radio Frequency Transmission

Radio transmission carries coordinate paths over radio waves. Each 16-bit coordinate is modulated directly onto a carrier frequency using ASK, FSK, or QAM. The receiving side demodulates the signal back to 16-bit values and validates them as coordinates.

- **Bit error detection:** The coordinate validity check (U+AC00–U+D7A3) provides inherent error detection. A single-bit flip that maps a valid coordinate outside the range is immediately rejected.
- **No packet loss recovery:** The design targets deterministic transmission where loss is detected at the coordinate level, not the packet level. There are no sequence numbers; non-idempotent exchanges require the higher layer to add its own identification.
- **Latency:** The transmission latency is dominated by the physical layer, not by protocol processing. The receiving side spends approximately 0.39 ns per coordinate to validate and reconstruct the coordinate path.

3.2 Light: Optical Transmission

Optical transmission carries coordinate paths over optical media (fiber, free-space optics, Li-Fi). Each 16-bit coordinate is encoded as a sequence of light pulses or as a phase/amplitude state.

- **High bandwidth:** Optical media can support very high symbol rates, making this variant suitable for high-throughput applications.
- **Low latency:** The optical transmission itself is near-speed-of-light; the only processing latency is the coordinate validation on the receiving side.
- **No protocol stack:** TCP/IP, UDP, and all other protocol layers are bypassed. The message is transmitted directly as coordinates.

3.3 Sound: Acoustic Transmission

Acoustic transmission carries coordinate paths over acoustic media (air, water, solids). Each 16-bit coordinate is encoded as a frequency or amplitude state in an audio signal.

- **Low data rate:** Acoustic transmission is slower than RF or optical, but it is suitable for underwater communication, through-wall signaling, and other acoustic-limited environments.
- **Robustness:** The coordinate validity check provides inherent error detection even in noisy acoustic channels.

3.4 Modulation and Physical Layer Interface

No specific modulation scheme is prescribed. Each 16-bit coordinate is transmitted as a symbol; the symbol rate, modulation depth, and carrier frequency are determined by the physical medium and the application requirements. This decoupling allows deployment over any physical layer that supports 16-bit symbol transmission.

For burst-error-prone media (e.g., fading channels, acoustic environments), lightweight Forward Error Correction (FEC) can be combined at the physical layer. The coordinate validity check then serves as the final integrity gate, rejecting any symbols that do not map to valid coordinates.

3.5 Energy Efficiency

By eliminating the protocol stack, per-message energy consumption is reduced by an estimated 10–100× compared to conventional packet-based transmission. The energy cost per message is dominated by the physical transmission itself, not by protocol processing. For battery-powered IoT devices, deep-space probes, and other energy-constrained applications, this translates to significantly extended operational lifetime.

4 Physical Robustness

The physical-layer properties claimed for coordinate transmission reduce to a small set of well-understood mechanisms. Each is stated with its mathematical basis, and each is finite — robustness is bounded by signal theory, not by the coordinate structure. Where a property has a precedent (LoRa CSS, MSK/GMSK, matched filtering), the precedent is named; the coordinate structure's contribution is the combination, not the mechanism itself.

4.1 Doppler-invariant ratio signaling

A transmitter moving with radial velocity v_r scales all transmitted frequencies by the same Doppler factor:

$$\gamma = \frac{c}{c - v_r}$$

A coordinate carried by the ratio of two spectral components is therefore invariant under Doppler:

$$\frac{f_{r,1}}{f_{r,2}} = \frac{\gamma f_{t,1}}{\gamma f_{t,2}} = \frac{f_{t,1}}{f_{t,2}}$$

This is the same mechanism that gives chirp spread spectrum (LoRa CSS) its Doppler tolerance. The coordinate structure does not create the invariance; it inherits it, with one advantage: the ratio lands on a discrete coordinate rather than a continuous frequency estimate, so the demodulator output is a quantized decision with structural validation.

4.2 Integration gain is finite and linear

Observing a symbol for duration T within bandwidth B improves SNR by the matched-filter gain:

$$G_p = 10 \log_{10}(BT) \text{ dB}$$

Gain grows linearly in dB with integration time, not exponentially. There is no accumulation regime beyond the matched-filter bound, and the stream-length robustness follows this linear law.

4.3 FM threshold bounds the margin

Frequency modulation trades bandwidth for SNR:

$$\left(\frac{S}{N}\right)_{\text{out}} = 3\beta^2(\beta + 1) \left(\frac{C}{N}\right)_{\text{in}}$$

with β the modulation index, but only above the FM threshold, $(C/N)_{\text{in}} \gtrsim 10$ dB. Below the threshold, demodulation collapses. There is no infinite margin: deep-space links still require channel coding (Turbo/LDPC) operating near $E_b/N_0 \approx -1$ dB, as in every existing deep-space mission.

4.4 Symbol timing remains mandatory

Continuous-phase modulation (CPFSK, MSK) removes amplitude discontinuities but not symbol timing. The receiver must still estimate symbol boundaries within:

$$|\Delta T| < \frac{T_s}{2}$$

Framing collapse is eliminated; timing recovery is not. A long burst can still lose symbol timing and desynchronize subsequent symbols until resynchronization.

4.5 Separable-component soft decision

The coordinate value decomposes arithmetically:

$$v - \text{U+AC00} = i \cdot 588 + m \cdot 28 + f, \quad i \in [0, 19), \quad m \in [0, 21), \quad f \in [0, 28)$$

If the modulation maps initial, medial, and final components to separable signal dimensions, each component can be decoded independently, and the symbol error probability factors:

$$P_e = 1 - (1 - p_i)(1 - p_m)(1 - p_f) \approx p_i + p_m + p_f$$

for small component error probabilities. This soft-decision structure is real, but it is a modulation design choice — a three-tone pattern in the spirit of DTMF or chirp sequences — not a property of the coordinate encoding itself. The arithmetic decomposition above is a property of the encoding; its mapping onto separable physical dimensions is not.

4.6 Information efficiency

Each transmitted 16-bit symbol carries:

$$\eta = \frac{\log_2 11,172}{16} \approx 0.84$$

of useful information. The self-validation space costs 16% of the raw symbol rate relative to unvalidated 16-bit symbols, and this tax is paid on every medium.

4.7 Summary of the physical claim

No single mechanism here is new: continuous-phase modulation, chirp-style Doppler tolerance, matched-filter integration, and component-wise soft decision all have established precedents. The claim is the combination — Doppler-invariant ratios landing on discrete coordinates, structural self-validation at zero marginal cost, and separable soft-decision components — with the 300-gate decoder making the digital side of the combination hardware-resident. Robustness is finite, quantified by the formulas above, and never exceeds the bounds that signal theory imposes.

5 Empirical Validation

The transmission pipeline has been validated through software simulation. The following measurements are based on the same ARMv8.4-A Firestorm core used for Tagma-KV benchmarks.

5.1 Coordinate Validation Latency

Operation	Latency	Notes
Single coordinate validation	0.39 ns	Range check + structural decomposition
6-syllable coordinate path validation	2.34 ns	6 coordinates $\times$ 0.39 ns
19-syllable coordinate path validation	7.41 ns	19 coordinates (SHA-256 scale)

The validation cost scales linearly with the number of coordinates but remains below 10 ns even at SHA-256 scale. There is no parsing cost because the coordinates are already the message.

5.2 Transmission Overhead Comparison

Protocol	Per-Message Overhead	Validation Cost
TCP/IP over Ethernet	~1000 ns	Checksum + header parsing
UDP over Ethernet	~500 ns	Header parsing
BLE	~10–100 μ s	Packet parsing + CRC
LoRa	~100 μ s–10 ms	Packet parsing + CRC
Tagma-WAV	0 ns	0.39 ns per coordinate (digital)

Protocol overhead is zero because there is no protocol.

The validation column reports the digital path only: range check and structural reconstruction after demodulation. Physical-layer cost is media-dependent and modeled separately (Section The Transmission Cost Model).

5.3 Error Detection Comparison

Mechanism	Overhead	Detection Rate
CRC32	4 bytes per packet	~99.9999% (arbitrary multi-bit errors)
Checksum	2 bytes per packet	~99.99% (arbitrary multi-bit errors)
Tagma-WAV coordinate validity	0 bytes	24.1% of single-bit flips

The structural validity check provides inherent error detection at zero additional cost. The rates are not directly comparable: CRC and checksum detection is defined over arbitrary error patterns, while the coordinate check is a single-bit analysis. The complement is a design consideration: the remaining 75.9% of single-bit flips map to another valid coordinate and are silently accepted as a different command. For command channels this argues for a small command table with semantic validation at the application layer, or for FEC at the physical layer; for data channels, it argues for redundant syllables or an application-layer checksum. The 24.1% figure follows from the structure of the valid range: flips of bit 15 or bit 14 always leave the U+AC00–U+D7A3 interval, and boundary values contribute the remainder.

6 The Transmission Cost Model

Let a message M of length L bytes be transmitted.

Structural transmission:

$$T_{\text{tagma-wav}}(M) = T_{\text{convert}}(M) + T_{\text{validate}}(M) + T_{\text{physical}}(M)$$

where:

- $T_{\text{convert}}(M)$: message $\rightarrow$ coordinate path conversion. A model parameter, estimable from the encoding implementation; ~ 5 ns is the reference value for $L=4$. Note that 4 bytes exceed the $13.45N$ -bit bound for $N=2$ (26.9 bits), so $L=4$ requires $N=3$.
- $T_{\text{validate}}(M)$: coordinate validation, 0.39 ns per coordinate (digital path only)
- $T_{\text{physical}}(M)$: symbol modulation, demodulation, clock recovery, and propagation. Media-dependent: sub-ns (optical), ns- μ s (short-range RF), to ms-scale per symbol (LoRa at low data rates, acoustic). This term dominates end-to-end latency in every real deployment and is outside the scope of the digital-path simulation.

Conventional packet-based transmission:

$$T_{\text{packet}}(M) = T_{\text{encode}}(M) + T_{\text{header}}(M) + T_{\text{checksum}}(M) + T_{\text{parse}}(M)$$

where each component is at least 10–100 ns, and the total grows with protocol stack depth.

Critical observation: The entire protocol stack is eliminated. The digital transmission cost is the conversion cost (one-time) plus the validation cost (per coordinate). There is no header to parse, no checksum to compute, and no protocol state to maintain. The remaining cost floor is the physical layer, which is bounded by the medium rather than by protocol processing.

7 Application Domains

7.1 Robotics and Autonomous Systems

Robots operating in unstructured environments need deterministic, low-latency communication. Commands transmitted as coordinate paths are executed immediately upon validation.

- **Command latency:** Sub-10 ns digital validation means commands are dispatched as soon as validation completes; end-to-end latency remains bounded by the physical medium.
- **No parser:** The robot firmware receives the coordinate path directly and maps it to an action without parsing a command string.
- **Inherent validation:** Invalid commands are rejected at the hardware level, not after executing partially.

7.2 IoT and Sensor Networks

IoT devices have limited power and processing capability. Communication overhead is reduced to near zero, enabling:

- **Lower power:** No protocol stack means fewer CPU cycles per transmission.
- **Smaller packets:** Each coordinate is 16 bits; a coordinate path of 6 coordinates is 96 bits, comparable to a very small packet but without headers.
- **No reassembly:** The coordinate path is the message; no packet reassembly is required.
- **Extended battery life:** 10–100 $\times$ energy reduction per message translates to years of operation on a single battery.

7.3 Space and Deep-Space Communication

Deep-space communication has extreme latency and limited bandwidth. The structural model provides:

- **Deterministic transmission:** The coordinate validity check provides partial inherent error detection (24.1% of single-bit flips, see Section Error Detection Comparison) without additional checksum overhead.
- **No retransmission ambiguity:** A received symbol is either a complete coordinate or rejected; there is no partial-coordinate state. Validity does not imply correctness — a corrupted coordinate may still map to another valid value, so integrity-critical commands need semantic validation.
- **Protocol independence:** Any physical medium can be used without requiring a protocol stack.
- **Energy efficiency:** Reduced per-message energy consumption is critical for spacecraft with limited power budgets.

7.4 AI Agent Communication

AI agents communicating with each other can use coordinate path transmission for high-speed, low-latency message passing. The coordinate path can encode both the message and the addressing information, eliminating the need for separate routing tables. The agent entity is the defining receiver of the observation model: the wave is received by infrastructure on behalf of the agent, and the agent interprets through its local context.

When a coordinate path shares its syllables between address and payload, the capacity equation splits accordingly:

$$C_{\text{payload}} = 11172^{N-N_{\text{addr}}}$$

For a 6-syllable coordinate path with one address syllable, the payload space is $11,172^5 \approx 1.74 \times 10^{20}$ combinations — still orders of magnitude beyond any single-agent message corpus.

The reference orchestration runtime for this domain is Actus: its headless agent architecture and planned neXus blackboard integration match the observation model, where the wave channel feeds the shared knowledge space and the runtime interprets and acts.

7.5 The “명” Scenario: A Single Sound as a Complete Message

This scenario is the illustrative case: it demonstrates the model in its most perceptible form. The defining deployment relationship is wave to agent entity: an RF emission reaches a logical agent, with the physical receiver separated from the interpreting runtime (Section AI Agent Communication). The acoustic case demonstrates the same structure in fused form, where receiver and actor are one body.

Consider a simple scenario: a machine emits the sound “명” — a single Hangul syllable corresponding to the coordinate U+BA85. This sound can be transmitted over any physical medium: acoustic, RF, optical, or any other carrier. Any receiving machine or entity that hears “명” can:

1. Validate the coordinate structurally (0.39 ns)
2. Reconstruct the coordinate path (1 syllable = 1 coordinate)
3. Execute the corresponding command or absorb the data

There is no language translation: the sound “명” is a structural coordinate that machines understand directly. This is the ultimate expression of parserless communication: the message and the medium are one.

7.5.1 The command space of one syllable

The Hangul block U+AC00–U+D7A3 contains exactly 11,172 valid syllables ($11172 = 19 \times 21 \times 28$). An N-syllable coordinate path therefore encodes:

$$C_N = 11172^N$$

N	Combinations	Coverage
1	1.12×10^4	A single sound selects one of 11,172 commands — enough for any real ISA (RISC-V ~200, x86-64 ~1,500) or syscall table (~450)
2	1.25×10^8	Command plus operand or parameter: covers every realistic command space
6	1.94×10^{24}	Standard address space ($\approx 2^{81}$) ⁴

One syllable already covers any practical command set; the second syllable supplies the operand or parameter, so two syllables are sufficient for complete command coverage without headers or framing.

7.6 Wave Observation: One Wave, Many Agents

Observation-based computing is the broader SSCCS paradigm; this section covers only its wave-channel form, within the signal-delivery scope of this document. A sender emits the sound “뽕” in open space, and every receiver in range observes the same coordinate simultaneously, by the broadcast property of the medium (Section Mesh Networks and Broadcast Concurrency). The coordinate encodes a shared observation, not a per-receiver instruction: it cannot encode receiver state. Each agent maps the coordinate through its own local state:

- **Position:** physical location and role determine whether the coordinate applies; filtering is receiver-side, as established above
- **Constraints:** the coordinate must pass the agent’s validity rules — structural membership (Section Error Detection Comparison) plus its own hardware capability bounds
- **Protocol:** the fixed-N framing convention (Section The Alternative) fixes how many syllables follow and what they mean
- **Contract:** the pair (structural filter, response policy) defined in the following paragraph

The receiver contract has a two-part structure. The structural filter determines which axes carry group, role, or command fields and which values select this receiver. It is expressible in the coordinate space itself, so it is specifiable at the framework level: a conforming receiver evaluates membership by axis comparison (Section Mesh Networks and Broadcast Concurrency) before consulting any local state. The response policy determines what the agent does once selected, and it remains application-level. The contract is therefore the pair (structural filter, response policy): the filter is framework-specified, the policy is the agent’s standing behavior rules, which this specification does not define.

One wave, one observation, N different responses. The sender never enumerates receivers, so scaling to arbitrary agent counts adds no sender-side bookkeeping. This is the physical analogue of a blackboard: a shared observation that many agents act on independently.⁵

⁴The reference implementation names this the default identity space, CoordId<6> (nex-fih), following the abstraction convention stated in the coordinate path footnote.

⁵The blackboard pattern — shared state observed and acted on by independent agents — is the organizing principle of the neXus FIH store (Fact-Intent-Hint): <https://github.com/ssccsorg/nexus>. The wave makes the same pattern physical.

7.7 The Observation Loop: Framework Stages

The wave-observation pattern is a loop of four stages. Each stage is either specified here, delegated to the underlying channel, or explicitly left to the application:

1. **Emission** (specified): the sender encodes a message as a coordinate path (Section The Alternative) and broadcasts it once; fixed-N framing fixes the path length.
2. **Observation** (specified and delegated): every receiver in range observes the same coordinate simultaneously, by the broadcast property (Section Mesh Networks and Broadcast Concurrency). Structural validation (Section Error Detection Comparison) is specified; demodulation, timing, and medium access are delegated to the underlying channel.
3. **Interpretation** (application-level): each receiver maps (coordinate, local state) to a decision. The framework specifies the validation boundary but not the decision logic; the decision function is the application's contract with itself.
4. **Action** (application-level): receivers act on their decisions. No acknowledgment to the sender is required; emission is fire-and-forget per emission. Feedback is not carried by the channel: a receiver that acts may re-emit a new coordinate, observed by whoever self-selects, or change physical state outside the wave channel. Coordination over time is the interleaving of emissions and state changes, not a reply path.

In the SSCCS portfolio, stages 3 and 4 are the domain of the Actus orchestration runtime, which executes agent decisions across the neXus FIH blackboard. The boundary holds regardless of the runtime: the framework specifies what a receiver must validate, and the runtime decides what the receiver does.

A conforming implementation must provide stages 1 and 2; stages 3 and 4 are the application's responsibility. This boundary is what keeps the framework medium-independent: the specification ends where the application's local state begins.

7.8 Observation Record and Evaluation Order

The observation stage hands its result to the interpretation stage through a single record. This record is the boundary interface of the framework: the only output of observation and the only input to interpretation. In the SSCCS portfolio the consuming runtime is Actus; any other runtime that accepts Tagma-WAV observations follows the same interface.

Field	Type	Description
coord	16-bit	Validated coordinate in U+AC00–U+D7A3
components	three 8-bit values	Decomposed initial (0–18), medial (0–20), final (0–27)
timestamp	64-bit	Monotonic clock at reception, local timebase
integrity	0.0–1.0	Signal quality metric (SNR, correlation score); media-dependent, not part of digital-path validation
medium	RF, Light, Sound	Physical channel that delivered the symbol

A receiver may add implementation-specific fields; the fields above must be present in every conforming record.

The decision layer evaluates the record in the following fixed order. The order is fixed because it keeps safety-invariant constraints ahead of policy logic; the representation of each stage remains application-level.

- **Structural gate** (framework-specified): the coordinate was validated before the record existed (Section Hardware Acceleration); a record never carries an invalid coordinate.

- **Constraint gate** (application-level, fixed order): the local system evaluates its physical and operational constraints (battery level, proximity, operational mode, safety interlocks) against the record. A failed constraint discards or defers the observation. This gate takes precedence over every policy.
- **Policy matching** (application-level): the local context maps (coordinate, local state) to a candidate set of actions, for example through Tagma-KV.
- **Dispatch** (application-level): the highest-priority candidate executes, or a new coordinate path is re-emitted as feedback.

The boundary between the structural gate and the remaining stages is the boundary of this document: the gate is framework-specified, the remaining stages are application-level, and their order is fixed to preserve safety.

7.9 Mesh Networks and Broadcast Concurrency

Radio propagation is inherently concurrent: one transmission illuminates a region, and every receiver in range hears it simultaneously. Connection-oriented protocols treat this concurrency as a problem to be managed; the structural model treats it as the addressing mechanism.

- **Broadcast is the default:** a coordinate path is transmitted once and received by every device in range. No destination address is prepended, because the coordinate's axes carry the addressing: a receiver decides relevance by structural filter (does axis[3] match my group?) at hardware speed. Addressing is receiver-side filtering, not sender-side routing.
- **Mesh is emergent:** any device that hears a coordinate can relay it. Because the coordinate is content-addressed (same message maps to the same coordinate), a relay drops duplicates by a membership test — the dedup mechanism of flooding is the coordinate itself, not a message cache. A mesh of N devices needs no routing tables and no connection state; it is relay plus structural filtering.
- **Hierarchy is re-emission with narrowing fields:** any receiver that acts may re-emit, and a re-emission that narrows a group field (group, subgroup, individual) shrinks the self-selecting audience at each hop. A single physical emission can branch into a command tree with no routing infrastructure: the tree lives in field values, not in forwarding state. When and how far to narrow is relay policy, which remains a network-layer choice as stated under Limits are physical.
- **The medium is an instance:** BLE, LoRa, UHF, optical, and acoustic are interchangeable physical layers for the same coordinate stream. Bluetooth is one protocol instance among many, not the architecture. The broadcast concurrency property holds for every radio medium and is weakened only for inherently directional media (free-space optical links).
- **Limits are physical:** broadcast media collide — concurrent transmissions corrupt each other, and the coordinate validity check detects only 24.1% of single-bit errors, so collision recovery still requires a medium-access policy (listen-before-talk, TDMA) at the physical layer. Relay policy (hop count, when to relay) remains a network-layer choice, and authentication with replay protection is required (Section Cryptographic Boundaries).

7.10 Streaming and Telemetry

Streaming exposes the model's actual boundary. Each stream unit is a 16-bit symbol carrying 13.45 bits of information, so the comparison baseline matters: against packetized streams the structural model removes framing, sequencing, and reassembly; against raw bitstreams (UART, I2S, ADC, SPI) it adds a 16% bandwidth tax for self-validation while contributing no framing advantage, because raw streams already have none.

Where the structural model is genuinely advantageous in a stream:

- **Self-identifying symbols:** command, data, and event symbols are distinguished by axis structure, so a mixed stream needs no demultiplexing layer. A raw bitstream carries no type information at all.
- **In-stream axis filtering:** the receiver can filter or demux by axis values at hardware speed (300-gate decoder), a capability raw streams do not provide.
- **Idempotent command resync:** in a fixed-N command convention, any valid symbol can restart a command, with no sequence state to reconcile.

- **Deterministic processing delay:** per-symbol digital cost is fixed, though raw streams share this property.

The honest limits follow from the same structure:

- **13.45 bits per symbol** means the information rate is 84% of the symbol rate; raw data streams skip this tax.
- **Per-symbol detection does not scale to long streams:** at 24.1% single-bit detection, undetected corruption accumulates linearly with stream length, so integrity-critical telemetry needs block-level redundancy or an application checksum regardless of the coordinate check.
- **Dimensional precision:** 13.45 bits per axis is roughly three decimal digits; a float32 coordinate needs three axes, so raw dimensional streams ($4 \times \text{float32} = 16$ bytes) transmit at a loss versus 12-syllable coordinate paths (24 bytes).
- **Timing cascades remain:** a long burst can still lose physical symbol timing, desynchronizing subsequent symbols until resync; only the packet-level cascade is eliminated.

8 Comparison with Existing Communication Paradigms

Feature	TCP/IP	UDP	BLE	LoRa	Tagma-WAV
Protocol overhead	High	Medium	Medium	Medium	Zero
Checksum	Yes	Yes	Yes	Yes	Partial (24.1% single-bit)
Parser required	Yes	Yes	Yes	Yes	No
Latency per message	μs –ms	μs	ms	ms	<10 ns digital (physical excluded)
Deterministic validation	No	No	No	No	Yes
Hardware acceleration	Limited	Limited	Limited	Limited	300-gate decoder
Per-message energy	High	Medium	Medium	Medium	10–100× lower
Medium independence	No	No	No	No	Yes (RF, light, sound, any)

This model does not compete with TCP/IP or UDP on their own terms. It replaces the container-based model entirely. Where TCP/IP is a protocol stack, the coordinate path is a structural transmission primitive.

8.1 Content-Centric Networking Precedent

The content-addressed property of the coordinate path has a named precedent: Named Data Networking (NDN) routes by content names instead of host addresses⁶. The shared principle is that content identity, not endpoint address, is the network primitive. The difference is the dimension of the addressing structure.

NDN names are hierarchical strings resolved by longest-prefix matching at line speed. A string is a container: it must be parsed, compared, and matched against routing state, and its validity is unbounded by structure. The coordinate path replaces the string layer with a structurally constrained symbol space: every symbol is a member of the 11,172-syllable block, validates by arithmetic decomposition in 0.39 ns, and carries 13.45 bits of information. Where NDN maintains routing state for strings (forwarding table, content store, pending interest table), the coordinate model adds no routing state: receivers self-select by axis membership at hardware speed.

⁶Named Data Networking: Jacobson, Smetters, Thornton, Plass, Briggs, and Braynard, “Networking Named Content”, ACM CoNEXT 2009: <https://dl.acm.org/doi/10.1145/1658939.1658941>

The interaction pattern differs on a separate axis. NDN is content-centric and still request-response: an Interest packet asks, a Data packet answers. The wave model is emission-only: there is no request, no per-pair state, and response diversity is produced entirely on the receiver side. Content addressing and request-response are orthogonal properties, and the coordinate model occupies the emission-only corner of that plane.

9 Hardware Acceleration

The Tagma-WAV decoder is a combinational circuit of approximately 300 gates that validates a 16-bit coordinate and extracts its structural components (initial, medial, final) in a single cycle. For transmission, the same decoder can be placed at the receiving side of any physical medium. It validates the incoming 16-bit values and reconstructs the coordinate path without any software involvement. The result is a hardware-accelerated signal layer where the cost of receiving a message is the cost of a single combinational gate delay.

On the transmission side, the same 300-gate decoder can be used to encode messages into coordinate paths directly from the application. The encoding cost is the same as the validation cost: 0.39 ns per coordinate. This makes Tagma-WAV a fully symmetric, hardware-accelerated signal primitive.

10 Cryptographic Boundaries

The coordinate channel is not an authentication mechanism. This section states the design space for layering integrity and replay protection onto coordinate streams without reintroducing a packet layer. The mechanisms below are research directions, not a specified scheme: the exact arithmetic, key lifecycle, and reserved positions are implementation choices not yet made.

Coordinate Vernam. Modular addition in the syllable space preserves structural validity: a cipher coordinate $c' = (c + k) \bmod 11,172$ maps back to a valid syllable, so every received symbol passes the structural gate (Section Error Detection Comparison) before decryption. If the key stream is used once, this is a Vernam cipher in the syllable space; reuse degrades it to a substitution cipher, so key distribution and re-keying dominate the cost.

Nonce coordinates. A reserved position in the coordinate path carries a counter or random nonce, making each emission unique and replay detectable. The nonce consumes capacity: N syllables with one nonce position leaves 11172^{N-1} effective combinations.

Key identifiers. A keyid position selects among a small set of shared keys, so re-keying does not require reconfiguring the fleet. Keyid and nonce together give per-emission authentication with replay protection, at the cost of two reserved positions per emission.

The structural gate remains the first line of defense regardless of the authentication layer, and the self-validation rate (24.1% of single-bit flips, Section Error Detection Comparison) is unchanged by these mechanisms.

11 Phased Rollout

The core transmission primitives are implemented in Rust and can be simulated on any platform that supports the Tagma coordinate space. As a preprint, this document precedes the physical-layer phases; the digital path is the validated portion today.

Phase 1: Software simulation. The transmission pipeline is simulated in Rust, with the same coordinate validation and conversion logic as Tagma-KV. This phase is complete; it covers the digital path only.

Phase 2: Physical medium integration. The transmission path is ported to specific physical media: RF modules (LoRa, BLE, 802.15.4), optical transceivers (SFP, Li-Fi), and acoustic transducers. This phase will measure and replace the T_{physical} model parameter with empirical data.

Phase 3: Hardware acceleration. The 300-gate decoder is integrated into the receiving path of custom hardware, enabling sub-nanosecond validation at the physical layer.

12 The Closing Question

We spent decades building increasingly complex protocol stacks to compensate for the fact that packets are containers, not structures. The container itself is unnecessary. What else becomes possible when the message is the coordinate, and the coordinate is the message?