Bitcoin as an Interplanetary Monetary Standard with Proof-of-Transit Timestamping

Jose E. Puente
Mobiletalk-Q, SL
jepuente@talk-q.com

Carlos Puente
Mobiletalk-Q, SL
carlospuenteg@talk-q.com

Abstract

We explore the feasibility of deploying Bitcoin as the shared monetary standard between Earth and Mars, accounting for physical constraints of interplanetary communication. We introduce a novel primitive, Proof-of-Transit Timestamping (PoTT), to provide cryptographic, tamper-evident audit trails for Bitcoin data across high-latency, intermittently-connected links. Leveraging Delay/Disruption-Tolerant Networking (DTN) and optical low-Earth-orbit (LEO) mesh constellations, we propose an architecture for header-first replication, long-horizon Lightning channels with planetary watchtowers, and secure settlement through federated sidechains or blind-merge-mined (BMM) commit chains [1, 2]. We formalize PoTT, analyze its security model, and show how it measurably improves reliability and accountability without altering Bitcoin consensus or its monetary base. Near-term deployments favor strong federations for local settlement; longer-term, blind-merge-mined commit chains (if adopted) provide an alternative. The Earth L1 monetary base remains unchanged, while Mars can operate a pegged commit chain or strong federation with 1:1 pegged assets for local block production. For transparency, if both time-beacon regimes are simultaneously compromised, PoTT-M2 (and PoTT generally) reduces to administrative assertions rather than cryptographic time-anchoring.

Keywords: Bitcoin; Lightning Network; Delay/Disruption-Tolerant Networking (DTN); Interplanetary networking; Time transfer; Satellite communications.

1 Introduction

Bitcoin's scarcity, decentralized verification, and predictable issuance make it a natural candidate for a universal monetary standard. As humanity moves toward establishing permanent settlements on Mars, a critical question arises: how can Bitcoin function reliably across interplanetary distances? The one-way light time (OWLT) between Earth–Mars ranges from about 3 to 22 minutes over a synodic cycle (from JPL Horizons/NAIF SPICE ephemerides [3]); combined with intermittent contacts and blackouts, this renders synchronized mining impractical, yet leaves room for verification, local payments, and asynchronous settlement [4].

To make notation unambiguous for readers and reviewers, Table 1 summarizes the symbols and parameters used throughout the paper (e.g., OWLT, RTT, jitter allowance J, and the incremental Lightning margin $\Delta_{\mathrm{CLTV}}^{\mathrm{extra}}$). We operationalize these quantities in Fig. 3, which converts Earth–Mars light-time into concrete `CLTV` blocks, and show the precise layer attachment points for PoTT metadata in Fig. 4.

Scope and regime. While our design targets AU-scale interplanetary distances typical of a star's circumstellar habitable zone (CHZ), not just Earth–Mars, we use the Earth–Mars pair as the canonical, near-term case study because it offers clear operational constraints, traffic models, and measurements. The mechanisms we propose (e.g., PoTT receipts, latency-aware Lightning policies, and header-first replication) generalize to other CHZ pairs.

Our main contributions are summarized in Section 2; we review the foundations and prior systems in Section 3. Early relativistic analyses anticipated that global block intervals would need to scale with interplanetary distances to preserve fairness (e.g., [5]); we instead preserve Bitcoin's base-layer parameters and shift adaptation to higher layers plus PoTT-based transport accountability.

Paper organization. Section 2 summarizes our main contributions. Section 3 reviews the state of the art across DTN/BPv7/BPSec, lunar networking frameworks (LunaNet/Moonlight), and Lightning timing/security results relevant to our setting. Section 4 formalizes the system model and assumptions for AU-scale, intermittently connected links in the circumstellar habitable zone (CHZ). Section 5 defines PoTT and its verification rules. Section 6 presents the end-to-end design, header-first replication, latency-aware Lightning policy, and settlement rails, together with the CLTV/CSV parameterization. (CSV can be time-based via BIP-68 relative-locktime units with 512-second granularity; height-based CSV should use the general formula with b_{target}, not a hard-coded "/10".) Section 7 analyzes security and outlines verification profiles. Section 8 details operational guidance and a phased deployment roadmap. Section 9 concludes. Appendix A specifies a minimal PoTT wire format and includes a test vector.

Requirement levels. We use the keywords MUST, SHOULD, and MAY as defined in RFC 2119 and RFC 8174 [6, 7] when, and only when, they appear in all capitals.

2 Contributions

We distill the paper's contributions as follows:

- **Interplanetary Bitcoin architecture.** A physics-aware architecture that preserves Bitcoin's base-layer parameters while enabling reliable operation across AU-scale, intermittently connected links via header-first replication, latency-aware Lightning, and asynchronous settlement rails.

- **Proof-of-Transit Timestamping (PoTT).** A new transport-level receipt primitive that cryptographically chains hop-timed custody attestations to Bitcoin payload hashes, yielding tamper-evident propagation histories suitable for off-chain disputes and operational accountability.

- **Latency-aware Lightning policy.** A closed-form parameterization of interplanetary timelocks (CLTV/CSV) that incorporates one-way light time and jitter allowances, and a packaging of PoTT evidence for watchtowers and adjudicators.

- **Security, verification profiles, and operations.** A threat model and verification profiles (e.g., PoTT-M2) that combine time-beacon audits, OWLT envelopes, and administrative diversity, plus practical guidance on key management, retention, and phased deployment.
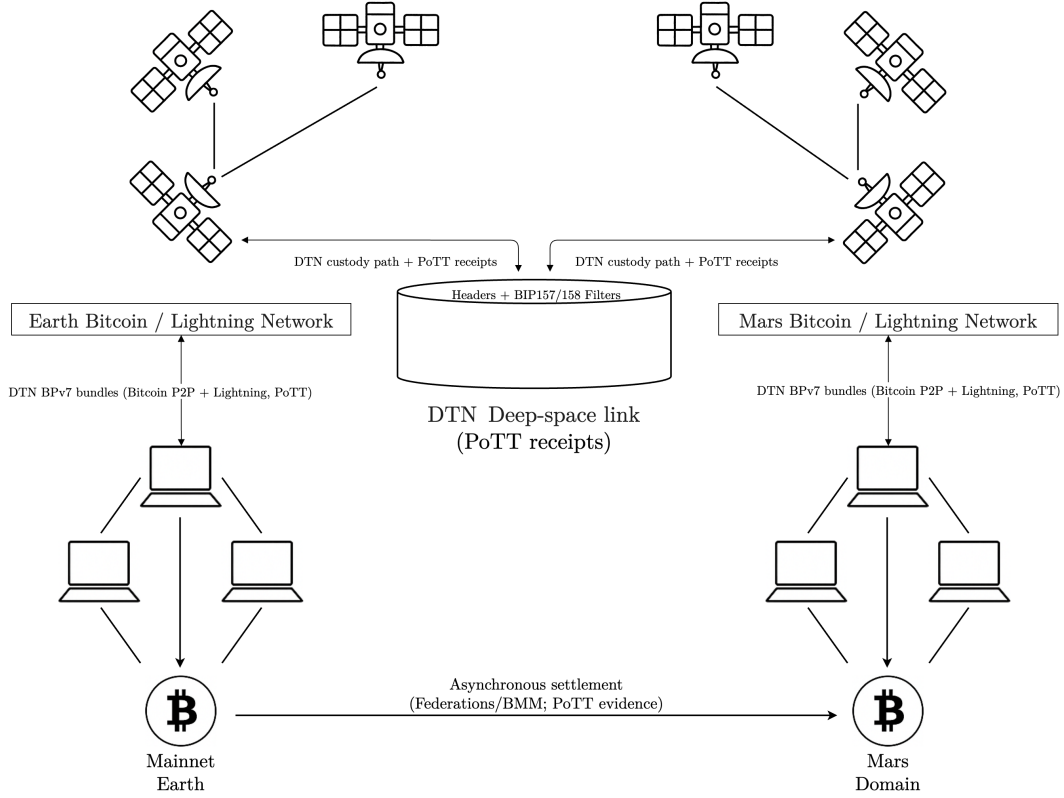
Figure 1: Conceptual architecture for interplanetary Bitcoin using Proof-of-Transit Timestamping (PoTT). DTN deep-space link carries BPv7 bundles with PoTT receipts; each planetary domain runs its own Bitcoin/Lightning network; asynchronous settlement via federations or BMM sidechains.
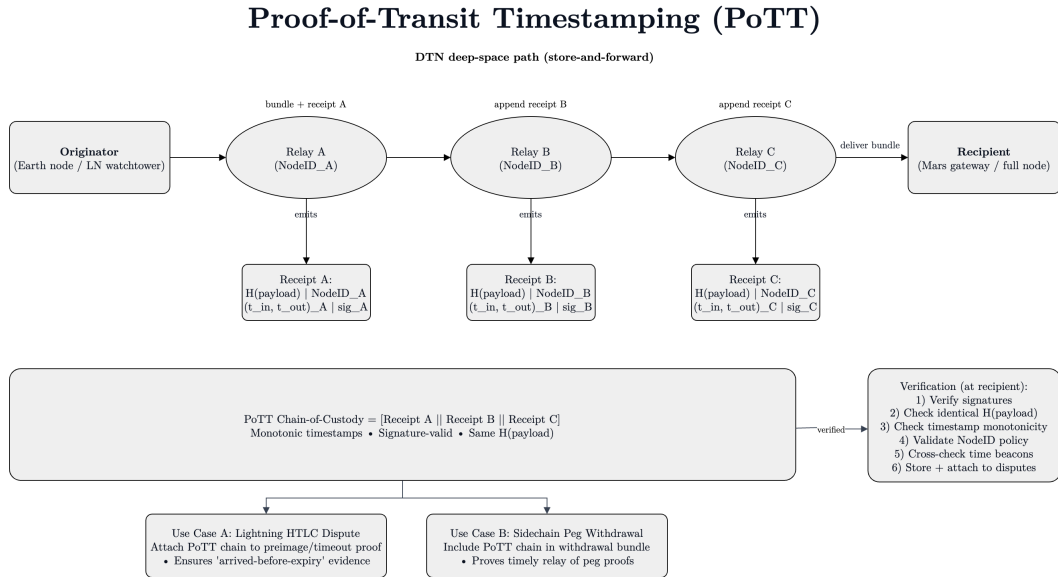

Figure 2: Proof-of-Transit Timestamping (PoTT) detailed flow across a DTN deep-space path. Each relay appends a signed receipt and the concatenated receipts form a verifiable chain-of-custody used for Lightning dispute evidence and sidechain pegs.

Table 1: Symbols and parameters used for latency-aware Lightning timelocks and PoTT verification.

| Symbol	Definition	Units/Example
OWLT	One-way light time Earth–Mars	minutes (3–22)
RTT	Round-trip light time; $\mathrm{RTT} = 2 \cdot \mathrm{OWLT}$	minutes
$\mathrm{RTT}_{\max}$	Maximum expected RTT over the contact plan	minutes
CLTV	Absolute timelock used in HTLCs (BIP65 `OP_CHECKLOCKTIMEVERIFY`)	blocks
CSV	Relative timelock for commitment/penalty flows (BIP112 `OP_CHECKSEQUENCEVERIFY`)	blocks
J	Contact/jitter allowance (DTN custody, scheduling, processing)	minutes (0, 30, 60)
b_{target}	Target L1 block interval	minutes (Bitcoin: 10)
$\Delta_{\mathrm{CLTV}}^{\mathrm{extra}}$	Additional `CLTV` blocks due to interplanetary latency; $\left\lceil \frac{\mathrm{RTT}+J}{b_{\mathrm{target}}} \right\rceil$	blocks (1–11)
B_{base}	Operator base CLTV margin (Earth policy)	blocks (e.g., 144 at Bitcoin's 10 min target)
M_{op}	Operational margin for queueing/retries	blocks (e.g., 1–2)
B_{total}	Recommended interplanetary CLTV; $= B_{\mathrm{base}} + \Delta_{\mathrm{CLTV}}^{\mathrm{extra}} + M_{\mathrm{op}}$	blocks
$t_{\mathrm{in}}, t_{\mathrm{out}}$	Ingress/egress timestamps encoded in TAI; UTC is for display-only conversions; implementations MUST use TAI consistently.	TAI seconds (CCSDS CUC, epoch 1958-01-01)
NodeID	Authorized relay identifier (public key; BIP-340 x-only) used in PoTT	32-byte public key
ν	Per-message nonce chosen at the origin; unique per payload hash h; relays **MUST** echo ν unchanged (see §5 and App. A)	16 bytes
$H(P)$	Payload digest included in receipts; **Bitcoin-native** where applicable (double-SHA256 for transactions/headers; BIP157/158 filter identifiers), otherwise SHA-256 over the canonical bytes P	bytes (e.g., 32)
δ	Safety allowance used in the arrived-before-expiry condition (convert to seconds when used in equations; $\delta \geq J + 2\sigma_t$).	minutes (convert to seconds in equations; e.g., $\delta = J + 2\sigma_t$)
σ_t	Time-source uncertainty bound (beacons/clock skew)	minutes (convert to seconds when used in equations; ≤ 1 min)
Δ_{MTP}	Policy bound on MTP–UTC skew used in disputes	minutes (convert to seconds when used in equations; e.g., ≤ 60)
κ	Optional reorg margin used in the MTP check	blocks (e.g., 0–6)
h_{expiry}	HTLC on-chain expiry height used in the arrived-before-expiry test (BOLT #2/#3); absolute block height	blocks
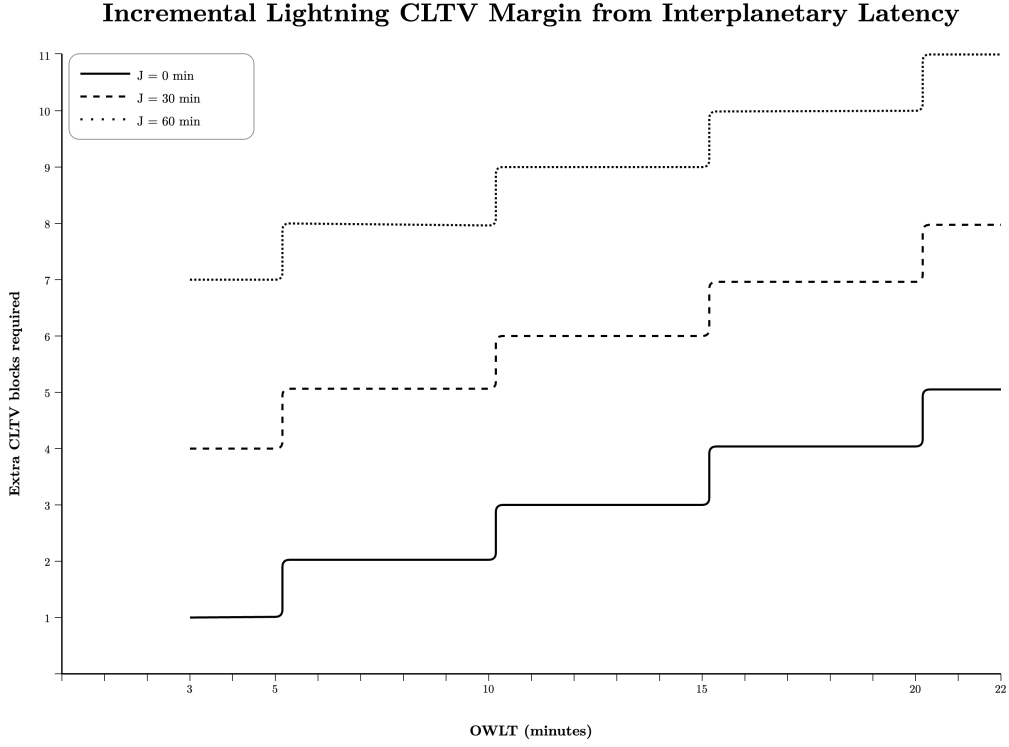
Figure 3: Incremental Lightning CLTV margin due to interplanetary latency. For a given jitter allowance J, the additional blocks are $\Delta_{\text{CLTV}}^{\text{extra}} = \left\lceil \frac{\text{RTT} + J}{b_{\text{target}}} \right\rceil$, where $\text{RTT} = 2 \cdot \text{OWLT}$ and b_{target} is the L1 target block interval (Bitcoin: 10 minutes). Steps occur at $\text{RTT} + J = k\,b_{\text{target}}$ and increase only when $\text{RTT} + J > k\,b_{\text{target}}$ for $k \in \mathbb{Z}_{\geq 0}$; at equality the value remains k. Operators then add their base policy (e.g., 144 blocks).

Interplanetary Bitcoin - Stack Integration Map

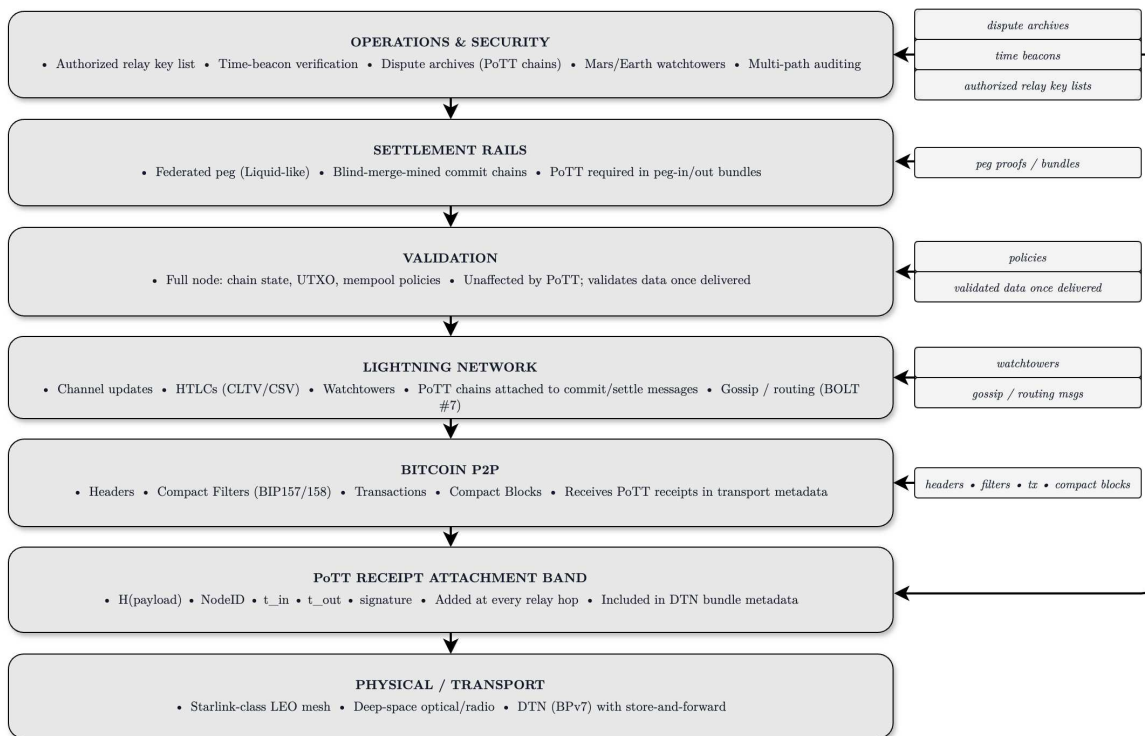|	
OPERATIONS & SECURITY • Authorized relay key list • Time-beacon verification • Dispute archives (PoTT chains) • Mars/Earth watchtowers • Multi-path auditing	*dispute archives* *time beacons* *authorized relay key lists*
SETTLEMENT RAILS • Federated peg (Liquid-like) • Blind-merge-mined commit chains • PoTT required in peg-in/out bundles	*peg proofs / bundles*
VALIDATION • Full node: chain state, UTXO, mempool policies • Unaffected by PoTT; validates data once delivered	*policies* *validated data once delivered*
LIGHTNING NETWORK • Channel updates • HTLCs (CLTV/CSV) • Watchtowers • PoTT chains attached to commit/settle messages • Gossip / routing (BOLT #7)	*watchtowers* *gossip / routing msgs*
BITCOIN P2P • Headers • Compact Filters (BIP157/158) • Transactions • Compact Blocks • Receives PoTT receipts in transport metadata	*headers • filters • tx • compact blocks*
PoTT RECEIPT ATTACHMENT BAND • H(payload) • NodeID • t_in • t_out • signature • Added at every relay hop • Included in DTN bundle metadata	
PHYSICAL / TRANSPORT • Starlink-class LEO mesh • Deep-space optical/radio • DTN (BPv7) with store-and-forward	

Figure 4: Stack integration map. PoTT receipts are attached in transport metadata (out-of-band) at every relay; Bitcoin P2P carries headers, filters, compact blocks, and transactions.
Lightning/watchtowers and settlement rails consume receipts; full-node validation remains unchanged.

3 State of the Art

We first review the systems and standards our design builds upon. DTN and the Interplanetary Internet form the transport substrate that makes planetary-scale networking feasible. The *Bundle Protocol v7* (BPv7) and its security framework, *BPSec*, specify store-and-forward forwarding, custody transfer, and hop-by-hop as well as end-to-end protection primitives suitable for long one-way light times (OWLT) and disconnections [8, 9, 10]. Institutional lunar architectures, NASA's *LunaNet* and ESA's *Moonlight*, are codifying navigation/communications services on top of DTN and are the nearest-term deployment targets where Bitcoin-related traffic would realistically ride [11, 12].

Terminology note. In the networking literature, "Proof of Transit" usually refers to data-plane path verification/OAM (e.g., SR-based schemes). Our *Proof-of-Transit Timestamping (PoTT)* is different: it produces hop-timed custody *receipts* bound to a Bitcoin payload hash to supply timing evidence and policy signals for Lightning/pegs. PoTT complements BPv7/BPSec; it does not replace them nor assert data-plane path correctness.

Why not just BPSec or Status Reports? *Bundle Status Reports (BSRs)* in BPv7 are optional diagnostic messages and are not globally authenticated; they do not create a per-hop custody time chain. *BPSec* provides end-to-end and/or hop-by-hop authentication of bundle blocks, but it does not yield a signed, per-hop transcript of custody times anchored to external time beacons and an OWLT policy. PoTT complements BPv7/BPSec by producing a verifiable, beacon-audited, hop-timed chain bound to Bitcoin payload hashes that can be attached to Lightning or peg disputes as off-chain evidence; it does not replace DTN security blocks nor assert data-plane path correctness.

Blockchain and payments beyond Earth. On-Earth, the Bitcoin network has already been decoupled from the terrestrial Internet via unidirectional broadcast (Blockstream Satellite) and alternative last-mile relays, demonstrating robustness to intermittent connectivity [13]. In orbit, SpaceChain and partners have operated blockchain payloads (multisig signing, custody) on the ISS and commercial satellites, showing that space-based cryptographic operations are operationally viable even if not consensus-critical [14]. These efforts stop short of defining a standards-aligned, physics-aware receipt layer for cross-planet dispute resolution, which is the focus of PoTT.

Lightning and timelocks under long RTT. Payment-channel security and capital efficiency under delay have been studied in the context of terrestrial networks. Sprites introduced constant locktimes across paths and reduced collateral under adversarial timing [15]. Work on routing and reliability (Pickhardt-Richter) quantifies path selection and failure probabilities relevant to setting CLTV/CSV policies under uncertainty [16]. Security analyses of timing manipulation ("time-dilation"/eclipse) further motivate auditable transport timing for watchtower policy [17]. The Bitcoin and Lightning standards (BIPs/BOLTs) provide the script/consensus primitives (CLTV/CSV, compact filters, encrypted transport) PoTT composes with, without altering L1 consensus [22, 24, 27, 28, 18, 19, 21, 26, 29, 30, 25]. PoTT is orthogonal to these mechanisms: it strengthens evidence and operational accountability under DTN constraints rather than changing channel-update or settlement protocols.

Relativity and macroeconomics across planets. Prior thought experiments on *relativistic cryptocurrencies* propose increasing base-layer block intervals proportionally to network diameter to preserve fairness [5]. Separately, the economics of trade with relativistic delays provides intuition for interest, discounting, and settlement risk at interplanetary scales [31]. In contrast, our design keeps Bitcoin L1 parameters unchanged and shifts adaptation to (i) a verifiable transport receipt chain (PoTT) aligned with BPv7/BPSec, and (ii) L2 policy and dispute mechanisms (Lightning/watchtowers, pegs). To our knowledge, there is no prior, peer-reviewed architecture that binds *ordered, hop-timed, cryptographically chained* transit receipts to Bitcoin

payload hashes and integrates those receipts into channel and peg dispute procedures across planetary domains.

4 System Model

We consider two primary operational domains, Earth and Mars, with the following elements.

- **Transport:** A combination of optical LEO mesh [32, 33] constellations for intra-planet connectivity and deep-space links (e.g., DSOC [34]) for interplanetary relays, operated under the Delay/Disruption-Tolerant Networking (DTN) model (Bundle Protocol v7). DTN provides custody transfer, late binding, and contact plans. PoTT receipts are carried in a dedicated DTN extension block parallel to the payload, or as a sidecar bundle, leaving core bundle semantics unchanged.

- **Nodes:** Bitcoin full nodes and Lightning nodes in each domain; optional sidechain federation members; watchtowers for Lightning monitoring.

- **Time sources:** Each relay and gateway maintains a local clock and access to one or more reference time-beacons (e.g., GNSS on Earth, optical two-way time transfer on deep-space links). We denote clock uncertainty bound by σ_t.

- **Adversary:** Can drop, delay, reorder, or inject messages; can compromise a subset of relays; cannot break standard cryptography (hashes, signatures) nor cause network-wide time-beacons to lie outside σ_t for extended periods.

- **Goals:** Preserve Bitcoin's monetary base; enable reliable verification and payments; provide auditable evidence for disputes without modifying Bitcoin consensus.

5 Proposed Approach: Proof-of-Transit Timestamping (PoTT)

5.1 Motivation

In interplanetary operations, disputes frequently hinge on *when* a message arrived relative to a timelock or operational deadline. Missing or delayed headers, transactions, or HTLC updates can force costly rollbacks or channel closures. Existing layers offer no cryptographically verifiable record of message propagation.

5.2 Formal Definition

Identifiers and signature scope. We define the identifiers carried in PoTT receipts and the exact byte range covered by each relay signature.

Wire format. Receipts are serialized as canonical CBOR [35] (big-endian integers); the signature covers keys 0–5 as specified. Appendix A defines the field keys and includes a test vector. We standardize that each per-hop signature s_i is a Schnorr signature over the encoded message $M_i = \text{encode}(h \parallel \nu \parallel \text{NodeID}_i \parallel t_{\text{in}}^{(i)} \parallel t_{\text{out}}^{(i)} \parallel \text{prev}_i)$. We instantiate NodeID as a BIP340 x-only secp256k1 public key (32 bytes) and set $s_i = \text{Sign}_{\text{NodeID}_i}(M_i)$. Other schemes are possible, but we standardize on TAI timestamps; see the "Timestamp scale" paragraph.

Timestamp scale. PoTT timestamps **MUST** encode 64-bit International Atomic Time (TAI) seconds using the CCSDS 301 Unsegmented Count (CUC) convention with epoch **1958-01-01 00:00:00 TAI**. Sub-second fields are *optional*; if present, implementations MAY include a 32-bit fractional field consistent with CUC coarse/fine encoding. This is a wire-format choice only and does not imply the use of UTC on-path. Verifiers **MUST** use a leap-second table valid at the claimed time $t_{\text{TAI}}^\star$ when converting to UTC for display or for comparison with

Bitcoin Median-Time-Past (MTP); disputes SHOULD apply a configured bound Δ_{MTP} when interpreting "arrived-before-expiry." See CCSDS 301 for details [38].

Canonical payload hashing. H is the content-identifier used throughout the PoTT chain. When the payload is a Bitcoin object that already has a canonical identifier, $H(\cdot)$ **MUST** equal that identifier: double-SHA-256 for block headers and transactions; the BIP157/158 identifiers for compact filters. For all other payload types, $H(\cdot)$ **MUST** be SHA-256 of the canonical byte encoding of the payload. The same choice of H **MUST** be applied consistently at every hop and is mirrored in Appendix A's test vector.

Let a payload P (e.g., a header, compact filter, transaction batch, or Lightning update) traverse a path of N relay nodes with identities $\{\mathrm{NodeID}_i\}_{i=1}^{N}$. Each relay i appends a *receipt*

$$R_i = (h, \nu, \mathrm{NodeID}_i, t_{\mathrm{in}}^{(i)}, t_{\mathrm{out}}^{(i)}, \mathrm{prev}_i, s_i)$$

Relation to BPSec. BPSec [9] authenticates and/or encrypts bundles at the block level but does not standardize custody timestamps, ordered ingress/egress times, or an anti-splice hash chain bound to a payload digest and a per-message nonce. PoTT is complementary: it yields dispute-grade, monotonic transit evidence without replacing DTN security blocks. In deployments that carry Lightning onion routing, Sphinx packets (BOLT #4) are placed in the BPv7 *payload*, while PoTT receipts reside in a BPv7 *extension block*; relays need not understand any Bitcoin semantics to forward bundles.

Chaining and anti-splice. Let $h = H(P)$ be the payload hash and let ν be a 128-bit random nonce minted by the originator for this bundle. Retransmissions of the same payload **MUST** mint a new ν; verifiers treat each (h, ν) as a distinct evidence set, and multiple PoTT chains may exist for the same h. ν MUST be unique per payload instance h (not reused across retransmissions) to preclude reuse of identical (h, ν) pairs across parallel paths. The originator sets ν once per payload instance and includes it in all bundle copies; relays **MUST** echo ν unchanged. PoTT binds each hop to the previous one with a hash chain and signs all binding material (specifically, $h, \nu, \mathrm{NodeID}_i, t_{\mathrm{in}}^{(i)}, t_{\mathrm{out}}^{(i)}, \mathrm{prev}_i$).

For hop $i \geq 0$ we define

$$\mathrm{prev}_0 := 0^{256}, \qquad \mathrm{prev}_i := H(R_{i-1} \setminus s_{i-1}) \text{ for } i \geq 1.$$

Here 0^{256} denotes a 256-bit all-zero string. Notation: For any receipt R_i, we write $R_i \setminus s_i$ to denote the byte-serialization of R_i with its signature field removed. Consequently, removing or reordering any interior hop changes prev_j for all subsequent receipts and invalidates their signatures; a chain cannot be cut-and-rejoined without detection.

Each relay i produces a receipt

$$R_i = (h, \nu, \mathrm{NodeID}_i, t_{\mathrm{in}}^{(i)}, t_{\mathrm{out}}^{(i)}, \mathrm{prev}_i, s_i),$$

where the signature covers

$$M_i = \mathrm{encode}\Big(h \parallel \nu \parallel \mathrm{NodeID}_i \parallel t_{\mathrm{in}}^{(i)} \parallel t_{\mathrm{out}}^{(i)} \parallel \mathrm{prev}_i\Big), \quad s_i = \mathrm{Sign}_{\mathrm{NodeID}_i}(M_i).$$

Verification. A verifier **MUST** (i) verify all signatures under an authorized relay key list; Operators **SHOULD** rotate relay signing keys on a fixed cadence and upon suspected compromise, and publish co-signed allowlists and revocation lists via DNSSEC-anchored distribution channels; To limit collusion, verifiers SHOULD require administrative diversity along the custody path (distinct operator keys and disjoint time-beacon sources); chains that fail diversity tests are downgraded in assurance or rejected. (ii) check h and ν are identical across all receipts; (iii)

check timestamp monotonicity $t_{\text{in}}^{(i)} \leq t_{\text{out}}^{(i)} < t_{\text{in}}^{(i+1)}$; (iv) check $\text{prev}_i = H(R_{i-1} \setminus s_{i-1})$ for every $i \geq 1$; (v) apply relay-policy checks and time-beacon audits. The inclusion of (ν, prev_i) prevents splice/replay attacks in which valid receipts for the same payload hash are recombined into a synthetic path. The detailed end-to-end receipt flow is shown in Fig. 2. We denote the chain-of-custody as $\mathcal{C}(P) = [R_0 || R_1 || \cdots || R_N]$ carried alongside P.

Overhead. For hash length 32 bytes, key identifier 32 bytes, nonce 16 bytes, ingress/egress timestamps 8 bytes each (TAI seconds; optional 32-bit fractional field per timestamp), previous-link hash 32 bytes, and a 64-byte Schnorr signature, each receipt is ≈ 203 bytes (roughly 200–205 B depending on CBOR map overhead). Thus chain overhead is $\approx 203N$ bytes for N hops (e.g., 10 hops ≈ 2.0 kB). If sub-second fractions are carried, we append a 32-bit fractional field per timestamp (8 bytes total for t_{in} and t_{out}).

5.3 Security Properties

PoTT aims to provide: (i) **Integrity** of payload and receipts; (ii) **Authenticity** of relay claims; (iii) **Ordering/Timeliness** via monotonic timestamps; (iv) **Publicly attributable claims** at the relay level. PoTT does not prevent censorship, but makes it auditable and attributable to a segment of the path.

5.4 Integration with Bitcoin Layers

- **Bitcoin P2P:** PoTT chains are attached in transport metadata (out-of-band) carried alongside headers, filters (BIP157/158), transactions, or compact blocks. Full-node validation remains unchanged; PoTT is not part of consensus. (e.g., BIP152 compact blocks; optional v2 transport BIP324) [26, 29].

- **Lightning:** For cross-planet hashed timelock contracts (HTLCs), *Evidence packaging:* PoTT chains are attached to commit/settle evidence packages for watchtowers and counterparties; we do not alter BOLT #2/#3 wire formats. PoTT chains are included with preimage/timeout evidence and monitored by watchtowers, strengthening disputes about "arrived-before-expiry." The additional `CLTV` margin is parameterized by Eq. (1). PoTT enhances operational accountability and out-of-band arbitration; it does not change on-chain LN enforceability, which remains purely timelock/script-based. Additionally, operators SHOULD provision CSV deltas for unilateral closes to cover $\text{RTT}_{\text{max}} + J$ plus an operational margin.

- **Sidechains:** Federated pegs or blind-merge-mined commit chains. In our design, we require PoTT evidence for peg-in/out bundles, providing auditability without altering Bitcoin's monetary base.

Anchoring "arrived-before-expiry" to Bitcoin time. When PoTT evidence is used for Lightning HTLC disputes or watchtower policy, timeliness is assessed against Bitcoin's MedianTimePast (MTP) as defined in BIP-113 [25], not wall-clock time. Let $C = [R_0, \ldots, R_k]$ be a verified PoTT chain with terminal timestamp $t_{\text{TAI}}^\star = t_{\text{out}}^{(k)}$ (TAI). We compare on UTC/Unix seconds by setting $t_{\text{UTC}}^\star = t_{\text{TAI}}^\star - \Delta_{\text{TAI}\to\text{UTC}}(t_{\text{TAI}}^\star)$, where $\Delta_{\text{TAI}\to\text{UTC}}(\cdot)$ denotes the current TAI–UTC offset. Let $H^\star$ be a best-chain header at height $h^\star$ with MTP $T_{\text{MTP}}(H^\star)$. A verifier *accepts* that a message arrived before expiry if

$$t_{\text{UTC}}^\star + \delta \leq T_{\text{MTP}}(H^\star) + \Delta_{\text{MTP}} \quad \text{and} \quad h^\star \leq h_{\text{expiry}} - \kappa.$$

Here δ is a safety allowance with $\delta \geq J + 2\sigma_t$, Δ_{MTP} is a policy bound on potential MTP–UTC skew (operators SHOULD choose a conservative value; we use $\Delta_{\text{MTP}} = 1\,\text{h}$ in examples). Unless otherwise specified, we set the MTP drift estimation window to $n = 4032$ blocks (approximately

28 days); dispute bundles **MUST** include the window and the observed bound, and operators **MAY** tighten n with local evidence, and κ is an optional reorg margin. Operationally, κ **SHOULD** exceed an empirical bound on historical MedianTimePast drift estimated over a recent window of n blocks, in addition to beacon uncertainty, and disputes **MUST** include the exact block range used for the MTP check. In this paper we set $\delta = J + 2\sigma_t$ by default (operators MAY increase it under degraded beacons). Historic observations of MTP–UTC divergence on mainnet are typically within tens of minutes; therefore we adopt, as a conservative *policy* bound (not an empirical guarantee), $\Delta_{\mathrm{MTP}} \leq 1$ hour to accommodate miner timestamp variance while minimizing false accepts. Operators *MAY* tighten this bound with local evidence. *All quantities in the inequality are in seconds; when policy parameters are maintained in minutes (e.g., $J, \sigma_t, \Delta_{\mathrm{MTP}}$), implementations convert by $\times 60$.* This use of PoTT is strictly out-of-band: it does not alter Bitcoin consensus or BOLT wire formats; it informs watchtower decisions, routing policy, and off-chain arbitration. In our prototype, PoTT is carried as a BPv7 extension block within DTN bundles and is consumed by relays and watchtowers; it remains outside Bitcoin consensus and BOLT wire protocols.

6 Architecture

Monetary domains. Earth retains the unchanged Bitcoin L1 as the monetary base. Mars operates a pegged commit chain (we use "commit chain" synonymously with a BMM sidechain) or strong federation for local block production with 1:1 pegged assets; no new L1 issuance is introduced. Operationally, we assume strong federations as a practical near-term path and treat BMM commit chains as optional if adopted.

We implement three coordinated layers (see Figs. 1 and 4).

1. **Header-first replication.** Gateways and relays prioritize *block headers* for timely fork choice and MTP anchoring. At $\sim$52,560 blocks/year this is about $80\,\mathrm{B} \times 52{,}560 \approx 4.2\,\mathrm{MB/yr}$ (i.e., ≈ 1.07 bps). *Compact filters* (BIP157/158) are treated separately: with a conservative median of $20\,\mathrm{kB}$ per block they amount to $\sim 1.05\,\mathrm{GB/yr}$ (≈ 267 bps). Deep-space relays *MAY* therefore ship headers immediately and schedule filters opportunistically (or by checkpoint), preserving safety while matching link budgets; transactions and compact blocks are fetched on demand via the custody path.

2. **Lightning for retail with latency-aware timelocks.** Cross-planet channels and routes set conservative CLTV/CSV deltas. The additional margin due to light-time and jitter is

$$\Delta_{\mathrm{CLTV}}^{\mathrm{extra}} = \left\lceil \frac{\mathrm{RTT} + J}{b_{\mathrm{target}}} \right\rceil \quad \text{blocks (Bitcoin: } b_{\mathrm{target}} = 10\,\mathrm{min}), \tag{1}$$

Here b_{target} is the target L1 block interval; for Bitcoin $b_{target} = 10$ minutes.

Worked example (sanity check). For an Earth–Mars one-way light time OWLT $= 22$ minutes, the round-trip is RTT $= 44$ minutes. With jitter allowance $J = 60$ minutes, the additional timelock margin is $\Delta_{\mathrm{CLTV}}^{\mathrm{extra}} = \left\lceil \frac{44 + 60}{10} \right\rceil = 11$ blocks. With operator base margin $B_{\mathrm{base}} = 144$ and operational margin $M_{\mathrm{op}} = 2$, the recommended total is $B_{\mathrm{total}} = 144 + 11 + 2 = 157$ blocks. CSV can be expressed in time using BIP-68 relative-locktime units (512-second granularity), i.e., a time-based CSV of t seconds corresponds to $\lceil t/512\,\mathrm{s} \rceil$ sequence units [23].
Note: On-chain enforcement remains purely script/timelock-based; PoTT evidence informs policy, watchtowers, and off-chain arbitration but does not alter on-chain validity.

3. **Asynchronous settlement rails.** A strong federation [37] or blind-merge-mined commit chain bridges domains; peg-in/out bundles *must* include PoTT evidence to prove timely relay of peg proofs and to provide tamper-evident logistics histories.

Observation 1 (Throughput–fairness heuristic bound; proof sketch). *Consider two planetary domains with one-way light time (OWLT) $\tau(t)$ between them that varies over a synodic cycle (e.g., Earth–Mars, with $\tau(t) \in [3, 22]$ minutes). Any fixed base-layer block interval b that preserves cross-domain mining fairness (bounded stale rate and approximately symmetric tip visibility) across the full cycle must satisfy*

$$b \gtrsim 2 \max_t \tau(t) + M.$$

Here M accounts for validation, queuing, and operational safety margins. In practice this implies b on the order of an hour or longer. Consequently, keeping fairness at base layer via a static b imposes a throughput penalty of at least $b/10$ relative to Bitcoin's current 10-minute cadence.

Sketch. We model block arrivals as a Poisson process of rate $1/b$ and approximate the worst-case one-way latency by $\max_t \tau(t)$ over a synodic cycle. Let the end-to-end propagation/validation delay be $D \approx 2 \max_t \tau(t) + M$, where M covers validation, queuing, and operational safety margins. In this back-of-the-envelope model, the probability that a found block becomes stale is $1 - e^{-D/b} \approx D/b$ for $D \ll b$. To keep the stale fraction $\leq \varepsilon$, it suffices to choose

$$b \gtrsim \frac{2 \max_t \tau(t) + M}{\varepsilon}.$$

For Earth–Mars, taking $\max_t \tau(t) \approx 22$ minutes, $\varepsilon = 0.05$, and M small relative to b gives $b \gtrsim 44/0.05 = 880$ minutes ≈ 14.7 hours. This is a heuristic argument (not a formal proof); a careful treatment would require an explicit miner/network model and policy analysis, which we leave for future work. Practically, this motivates keeping L1 parameters fixed and shifting adaptation to higher layers (e.g., PoTT receipts, Lightning/sidechains). □

7 Security Analysis

Threat model and guarantees.

> **Assumptions and Guarantees.** *Assumes* unbroken hash/signature primitives; availability of signed time beacons; at least one honest relay on each verified path; and integrity of ephemerides/contact plans used for OWLT envelopes. *Guarantees* custody attestation via a per-hop receipt chain, per-hop ordering with monotonic timestamps, and splice-resistance across the path; PoTT *does not* guarantee liveness or delivery.

Assurance summary. *PoTT provides accountability—tamper-evident custody plus per-hop timing—but it does not by itself provide liveness or censorship-resistance; those remain policy- and topology-dependent. Our verification profiles (e.g., **PoTT-M2**) mitigate collusion by requiring administrative diversity and independent time-beacon audits, and can be strengthened with path diversity for high-stakes disputes.*

In the degenerate case where *all* relays on the path collude and public time-beacons are compromised, PoTT evidence reduces to administrative assertions about custody (NodeIDs and path admission); verifiers *SHOULD* treat such chains as non-probative absent corroboration from independent anchors. In keeping with this, PoTT is intended to degrade gracefully to administrative assertions; policy profiles (Table 2) **SHOULD** require at least two independent, TAI-traceable timing anchors for any path whose evidence is used in adjudication.

PoTT provides *custody attestation with monotonic timestamps*: it binds an ordered set of relay claims to an unchanged payload and discloses ingress/egress times at each relay. It does

Table 2: Verification profile **PoTT-M2**. Operational knobs for verifiers.

In deployment, a receiver's watchtower may reject HTLCs that lack an accompanying PoTT chain per a stated policy profile (e.g., PoTT-M2), providing an explicit enforcement hook. Operators **MAY** enforce PoTT-M2 compliance in watchtower policy and federation membership, rejecting non-compliant evidence packages.

Parameter	Value/Requirement
Minimum relays (hops)	≥ 3 receipts
Administrative diversity	≥ 2 distinct operator domains; at least one anchor from each planetary domain
Time anchors	Signed public time-beacons on each domain (Earth, Mars)
Per-hop timing bound	$\lvert \Delta t \rvert \leq J$ and consistent with current OWLT envelope
Path diversity (high-stakes)	Require ≥ 2 path-diverse chains (disjoint operators/time-beacon regimes)
Authorized relay keys	Maintain allowlist; revoke failing audits and exclude unauthorized NodeIDs
Chain size cap	≤ 32 hops $or \leq 8$ kB of receipts per bundle
Retention window	Keep receipts ≥ 90 days; longer if attached to disputes

not guarantee liveness or delivery; relays may drop traffic after signing, and censorship remains possible. Absolute time need not be perfectly synchronized, verification requires monotonic per-hop times within policy bounds, and consistency with external anchors and contact-plan geometry.

Relation to BPSec. See §5.2.

Time anchors and policy profile. Verifiers SHOULD check PoTT chains against (i) signed public time-beacons on each domain (e.g., Earth and Mars) and (ii) the ephemeris-derived one-way light time (OWLT) envelope for the claimed window. We recommend a practical profile, *PoTT-M2*: at least three relays from at least two administrative domains, with at least one anchor from each domain; per-hop times must satisfy $\lvert \Delta t \rvert \leq J$ and be consistent with the current OWLT envelope. Operators maintain authorized relay key lists and revoke keys that fail audits. By default, for high-stakes disputes, verifiers **MUST** require at least two path-diverse PoTT chains anchored to disjoint time-beacon regimes; a single chain **MAY** suffice for routine settlements.

As a concrete instantiation of these checks, see Table 2. *OWLT envelope source:* Verifiers SHOULD derive the one-way/round-trip light-time envelope from published contact plans or ephemerides (e.g., SPICE kernels) and record the version used during verification.

Assumption (PoTT-M2). At least one audited relay on the verified path is honest, and the public time-beacon regimes on each planetary domain remain within σ_t of true time; if all relays collude and external beacons are compromised, PoTT evidence degrades to administrative assertions.

Multipath receipts and diversity policy. For high-stakes disputes, verifiers **SHOULD** require evidence from at least two path-diverse PoTT chains (disjoint relay operators and, when feasible, disjoint time-beacon regimes). When multipath was available but only a single path is presented, adjudicators may treat the evidence as weak. Verification policy rejects any PoTT chain whose timing is only explainable by back-dating beyond the declared clock-uncertainty bound σ_t and the published contact plans; chains must be consistent with beacon cross-checks.

Relay collusion or forged timestamps. Multiple independent time-beacons and cross-checks on contact plans limit feasible skew. Multi-path routing yields divergent chains if a colluding subset lies about timing.

Receipt omission or truncation. Verifiers enforce minimum-hop and diversity policies; missing receipts invalidate policy compliance. DTN custody supports custody-based retransmission (RFC 9171) on failure.

Sybil relays. Operators maintain authorized relay key lists with revocation. Watchtowers and federation members reject chains containing unauthorized NodeIDs.

Denial-of-service via oversized metadata. Cap PoTT chains to ≤ 32 hops or $\leq 8\,\mathrm{kB}$; compress receipts; use per-hop admission control.

Privacy. PoTT receipts expose hop count and coarse path structure to any party that inspects them. Deployments SHOULD minimize leakage by using the privacy mode (commit-and-reveal) described below.

Privacy mode (commit-and-reveal). Relays **MUST** onion-encrypt receipt metadata hop-to-hop (Sphinx per BOLT #4 [20]); routine attestations **SHOULD** redact per-hop times, and only disputes reveal full transcripts. For routine attestations, relays commit to the full receipt transcript by publishing $H_{\mathrm{txpt}} = H(R_0 \parallel \cdots \parallel R_N)$. The wire conveys only the tuple $(H_{\mathrm{txpt}}, \min(t_{\mathrm{in}}), \max(t_{\mathrm{out}}), \mathrm{hopcount})$. Thus observers learn only $[\min(t_{\mathrm{in}}), \max(t_{\mathrm{out}})]$ and hopcount; relay identities and per-hop timestamps remain encrypted and are revealed only if a dispute requires opening the transcript. In case of dispute, parties reveal the transcript and verify (i) that the recomputed H_{txpt} matches, (ii) per-hop times lie within $[\min(t_{\mathrm{in}}), \max(t_{\mathrm{out}})]$, and (iii) policy bounds are met. This achieves minimal disclosure while preserving verifiability of end-to-end timing. Deployments **SHOULD** minimize leakage by aggregating to $[\min(t_{\mathrm{in}}), \max(t_{\mathrm{out}})]$ and hop count for routine attestations, carrying receipts under layered onion encryption so each relay learns at most its predecessor and successor, and disclosing full transcripts only during disputes once policy requirements are satisfied (cf. Table 2).

8   Operational Considerations and Roadmap

Key management. Relays use hardware-backed keys with periodic rotation; public keys distributed via signed manifests. Relay allowlists are distributed as signed manifests (e.g., DNSSEC-anchored or RFC-style registries) with routine audits and revocation procedures.

Cross-domain trust (Earth↔Mars). Relay allowlists are published as DNSSEC-anchored manifests with mirrored roots in each planetary domain; a federated governance group co-signs periodic checkpoints and CRLs. Revocations propagate with maximum staleness bounded by the manifest TTL (hours–days); verifiers MUST account for this latency when auditing evidence that spans domains. **Time synchronization.** Combine GNSS (Earth), onboard oscillators, and deep-space two-way time transfer (e.g., CCSDS time-code formats and interoperability [38]); keep σ_t within operational bounds.
If beacons are unavailable or exceed the configured uncertainty bound σ_t, operators SHOULD increase the verification margin δ and/or quarantine PoTT evidence until clocks recover; for high-stakes operations (e.g., peg withdrawals) systems MUST fail closed. Operators **SHOULD** source at least two independent, TAI-traceable public time beacons (e.g., DSN/USNO and ESA/ESTEC) and record the beacon identities alongside any PoTT evidence attached to disputes.

Default margins. Unless otherwise stated, set $\delta = J + 2\sigma_t$; operators *MAY* increase this during degraded beacons.

Retention. Store PoTT chains for a fixed horizon (e.g., 90 days) unless attached to disputes.

Phased deployment. Phase 0: Earth testbeds; Phase 1: cis-Mars demos; Phase 2: Mars orbit/surface networks; Phase 3: scale security via blind merge-mining and mature governance.

9 Conclusion

PoTT supplies the missing accountability layer for delay-tolerant Bitcoin without any L1 consensus changes; together with header-first replication and latency-aware Lightning policy, it enables a practical interplanetary Bitcoin economy. Physics forbids synchronous, cross-planet competitive mining at Bitcoin's current parameters, but it does *not* forbid a Bitcoin-only economy across planets. This paper showed that by (i) prioritizing header-first replication for chain awareness, (ii) operating Lightning with latency-aware CLTV/CSV policies derived from one-way light time (OWLT) and jitter budgets, and (iii) settling asynchronously via strong federations or blind-merge-mined (BMM) commit chains (we use "commit chain" synonymously with BMM sidechain), a practical and incrementally deployable architecture emerges for interplanetary Bitcoin.

The core technical contribution is **Proof-of-Transit Timestamping (PoTT)**, a transport-level receipt layer that binds an ordered, hop-timed chain of custody to the hash of the transported Bitcoin payload (headers, filters, transactions, compact blocks, or Lightning updates). PoTT does *not* alter consensus or script semantics; instead, it raises the quality of *evidence* available to watchtowers, counterparties, and peg operators. By combining per-hop signatures, a splice-resistant hash chain, and TAI-encoded ingress/egress times, PoTT turns otherwise opaque DTN store-and-forward behavior into auditable artifacts. We specified a verification profile (PoTT-M2) that cross-checks receipts against authorized relay keys, public time-beacons, and OWLT envelopes, and we showed how evidence packages anchor "arrived-before-expiry" claims to Bitcoin's MedianTimePast (BIP-113) without introducing new consensus rules. PoTT is strictly *out-of-band*: receipts are consumed by watchtowers and peg federations; no changes to Bitcoin consensus or BOLT wire formats are required.

From an operations perspective, our design keeps failure domains small and controllable. Header-first replication requires only an $\mathcal{O}(\text{MB/yr})$ *annual budget*—about 4.2 MB/yr for headers (i.e., ≈ 1.07 bps sustained)[1]—and compact filters at 20 kB/block require ≈ 1.05 GB/yr (i.e., ≈ 267 bps). Lightning channels pay a predictable timelock premium $\Delta_{\text{CLTV}}^{\text{extra}} = \lceil \frac{\text{RTT}+J}{b_{\text{target}}} \rceil$ blocks (Fig. 3), and PoTT receipts add kilobytes, not megabytes, per interplanetary bundle. The architecture composes with existing BIPs/BOLTs (e.g., BIP152, BIP157/158, BIP324) and DTN/BPv7/BPSec without special treatment from Bitcoin full nodes. This separation of concerns preserves Bitcoin's monetary base and validation model while letting higher layers internalize the unavoidable costs of distance.

Limitations and scope. PoTT provides accountable *custody attestation*, not liveness; relays may still censor or drop traffic. Our threat model assumes sound cryptography, bounded clock uncertainty, and independently operated time-beacons; extreme faults in these primitives can weaken evidence quality. Finally, we have not attempted to equalize miner fairness across planets at L1, a path that would require hour-scale blocks and collapse global throughput. Instead, we accept temporarily segmented fee markets. During periods of opposition or limited interdomain connectivity, participants should expect temporary rebalancing costs and FX-like spreads across domains; these frictions diminish as liquidity and path diversity improve, and rely on asynchronous pegs and periodic liquidity rebalancing across domains.

---

[1]Computation: headers at 80 B per block $\times$ 52,560 blocks/yr ≈ 4.2 MB/yr ≈ 1.07 bps; compact filters at 20 kB per block $\times$ 52,560 blocks/yr ≈ 1.05 GB/yr ≈ 267 bps.

Implications. Under these constraints, Bitcoin can function as a shared monetary standard between Earth and Mars: verification remains local and cheap; retail payments proceed over LN with safety margins that scale with physics; and high-value flows settle asynchronously with tamper-evident logistics histories. In short, PoTT supplies the missing accountability layer for delay-tolerant Bitcoin, turning a hard physical limit (light time) into explicit policy knobs instead of consensus changes.

10 Future Work

While our proposal is already concrete enough for early prototyping, three areas stand out as the most impactful next steps:

1. **Formal security and timing proofs.** Develop a rigorous, game-based security proof for PoTT's anti-splice construction and a complete end-to-end timing model that incorporates beacon uncertainty, contact-plan geometry, and DTN queueing. This would strengthen confidence in PoTT as a standard.

2. **Interoperable specifications and reference implementation.** Produce an open, I-D/RFC-style specification for PoTT's DTN extension block and receipt wire format, and implement a public reference library with Bitcoin/Lightning integration for watchtowers and sidechain pegs.

3. **End-to-end testbeds.** Run terrestrial emulation of AU-scale RTT and blackout conditions, followed by public pilots linking separated ground stations with header-first replication, Lightning channels, and PoTT evidence enforcement.

These steps would take PoTT from a research proposal to an operational, cross-vendor standard ready for deployment in interplanetary Bitcoin networks.

Appendix A: Minimal PoTT Wire Format and Test Vector

Receipt structure. Receipts use *canonical CBOR* maps with the following integer keys and types (canonical ordering by key):

- **0: h**   *bstr* (32) — payload digest $H(P)$ (Bitcoin-native: double-SHA256 for transactions/headers; BIP157/158 filter identifiers; otherwise SHA-256 of canonical P)
- **1: ν**   *bstr* (16) — per-message nonce; MUST be unique per payload instance h (not reused across retransmissions)
- **2: node** *bstr* (32) — relay NodeID (public key; BIP-340 x-only)
- **3: tin**   *int* — ingress time (TAI seconds; CCSDS CUC, epoch 1958-01-01)
- **4: tout** *int* — egress time (TAI seconds; CCSDS CUC, epoch 1958-01-01)
- **5: prev** *bstr* (32) — $\text{prev}_i = H(R_{i-1} \setminus s_{i-1})$ — anti-splice binding to the prior hop's receipt sans signature
- **6: sig**   *bstr* (64) — BIP340 Schnorr over the canonical CBOR of keys 0–5

Minimal test vector (normative). The following CBOR hex encodes one receipt (keys 0–6). Values are provided in full for reproducibility; the signature is a dummy (parser-checkable).

```
00 58 20 83 a0 12 ac 61 2c 83 f6 89 17 73 87 35 34 65 fb 96 13 56 e8 1b cd 8a da    #0:h = H(P)
4b a0 d6 57 da 1c 26 85
01 50 22 19 c6 46 c0 c3 53 d1 87 ef b2 ca b9 ef 61 5b                               #1:nu
02 58 20 d4 06 3a ea 17 03 81 ce ca f4 d4 3b 1e 8d d3 2e c1 34 9f ac 78 ed c0 75     #2:node
ce 08 fb 36 4d 60 40 43
03 1B 0000000065B9B8A0                                                              #3:tin
04 1B 0000000065B9BD40                                                              #4:tout
05 58 20 2c 77 0e 00 80 83 e6 2a fd 13 76 98 ce 19 6d b6 5c b4 06 eb 2b 4c 50 6c     #5:prev
b6 fa 0c 54 6f 95 d8 55
06 58 40 db d5 95 30 45 c5 b1 31 a2 5e ca bd 6f 2d 78 6b 28 7e e1 da 3a e2 84 5b     #6:sig (dummy)
27 89 b5 1c cd c3 82 ef 83 68 e0 36 50 87 9c 71 75 5b 7f da 46 6b 44 a7 32 18 f6
82 06 25 e9 59 2f cc b3 a6 13 3b 92 b2
```
Implementations **MUST** reject non-canonical encodings and unknown keys.

Acknowledgments

References

[1] Paul Sztorc, CryptAxe. *BIP 300. Hashrate Escrows (Consensus layer)*. (2017). https://github.com/bitcoin/bips/blob/master/bip-0300.mediawiki.

[2] Paul Sztorc, CryptAxe. *BIP 301. Blind Merged Mining (Consensus layer)*. (2019). https://github.com/bitcoin/bips/blob/master/bip-0301.mediawiki.

[3] JPL Horizons / NAIF. *JPL Horizons On-Line Ephemeris System; NAIF SPICE Toolkit*. (Accessed: August 1, 2025). https://ssd.jpl.nasa.gov/horizons/; https://naif.jpl.nasa.gov/naif/.

[4] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. (2008). https://bitcoin.org/bitcoin.pdf.

[5] A. Ladha, *Hypothetical Problems concerning the Theory of Relativity on Cryptographic Currency Implementations*, arXiv:1604.04265 [cs.CR], 2016. https://arxiv.org/abs/1604.04265.

[6] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119 (1997). https://www.rfc-editor.org/rfc/rfc2119.

[7] B. Leiba. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. IETF RFC 8174 (2017). https://www.rfc-editor.org/rfc/rfc8174.

[8] S. Burleigh et al. *Bundle Protocol Version 7*. IETF RFC 9171 (2022). https://www.rfc-editor.org/rfc/rfc9171.

[9] E. Birrane III, K. McKeever, *Bundle Protocol Security (BPSec)*, IETF RFC 9172. (2022). https://www.rfc-editor.org/rfc/rfc9172.

[10] NASA. *Delay/Disruption Tolerant Networking*. (Accessed: August 1, 2025). https://www.nasa.gov/communicating-with-missions/delay-disruption-tolerant-networking/.

[11] NASA. *LunaNet Interoperability Specification (LNIS) - Version 5, Baseline*. (2025). https://www.nasa.gov/wp-content/uploads/2025/02/lunanet-interoperability-specification-v5-baseline.pdf.

[12] European Space Agency. *Moonlight*. (Accessed: August 1, 2025). https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Moonlight.

[13] Blockstream. *Blockstream Satellite.* (Accessed: August 1, 2025). https://blockstream.com/satellite/.

[14] SpaceChain. *Blockchain payloads on the ISS and commercial satellites.* (Accessed: August 1, 2025). https://spacechain.com/.

[15] Andrew Miller, Iddo Bentov, Ranjit Kumaresan, Patrick McCorry. *Sprites and State Channels: Payment Networks that Go Faster than Lightning.* IACR Cryptology ePrint Archive 2017/1076 (2017). https://eprint.iacr.org/2017/1076.

[16] Rene Pickhardt and Stefan Richter. *Optimally Reliable & Cheap Payment Flows on the Lightning Network.* arXiv:2107.05322 (2021). https://arxiv.org/abs/2107.05322.

[17] Antoine Riard and Gleb Naumenko. *Time-Dilation Attacks on the Lightning Network.* arXiv:2006.01418 (2020). https://arxiv.org/abs/2006.01418.

[18] *BOLT #2. Peer Protocol for Channel Management.* (ongoing). https://github.com/lightning/bolts/blob/master/02-peer-protocol.md.

[19] *BOLT #3. Bitcoin Transaction and Script Formats.* (ongoing). https://github.com/lightning/bolts/blob/master/03-transactions.md.

[20] *BOLT #4. Onion Routing Protocol.* (ongoing). https://github.com/lightning/bolts/blob/master/04-onion-routing.md.

[21] *BOLT #7. P2P Node and Channel Discovery.* (ongoing). https://github.com/lightning/bolts/blob/master/07-routing-gossip.md.

[22] Peter Todd. *BIP 65. OP_CHECKLOCKTIMEVERIFY.* (2014). https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki.

[23] R. O'Connor, P. Todd, and M. Friedenbach. *BIP 68: Relative lock-time using consensus-enforced sequence numbers.* (2015). https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki.

[24] BtcDrak, Mark Friedenbach, Eric Lombrozo. *BIP 112. OP_CHECKSEQUENCEVERIFY.* (2016). https://github.com/bitcoin/bips/blob/master/bip-0112.mediawiki.

[25] Thomas Kerin, Mark Friedenbach. *BIP 113. Median time-past as endpoint for lock-time calculations.* (2015). https://github.com/bitcoin/bips/blob/master/bip-0113.mediawiki.

[26] Matt Corallo. *BIP 152. Compact Block Relay.* (2016). https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki.

[27] Olaoluwa Osuntokun, Alex Akselrod, Jim Posen. *BIP 157. Client Side Block Filtering.* (2017). https://github.com/bitcoin/bips/blob/master/bip-0157.mediawiki.

[28] Olaoluwa Osuntokun, Alex Akselrod. *BIP 158. Compact Block Filters for Light Clients.* (2017). https://github.com/bitcoin/bips/blob/master/bip-0158.mediawiki.

[29] Dhruv Mehta, Tim Ruffing, Jonas Schnelli, Pieter Wuille. *BIP 324. Version 2 P2P Encrypted Transport Protocol.* (2019). https://github.com/bitcoin/bips/blob/master/bip-0324.mediawiki.

[30] Pieter Wuille, Jonas Nick, Tim Ruffing. *BIP 340. Schnorr Signatures for secp256k1.* (2020). https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki.

[31] Paul R. Krugman. *The Theory of Interstellar Trade.* Economic Inquiry 48(4), 1119–1123 (2010). https://doi.org/10.1111/j.1465-7295.2010.00269.x.

[32] Starlink. *Business | Aviation.* (Accessed: August 1, 2025). https://www.starlink.com/business/aviation.

[33] SpaceX. *Starshield.* (Accessed: August 1, 2025). https://www.spacex.com/starshield.

[34] NASA/JPL. *Deep Space Optical Communications (DSOC) Mission Updates.* (2023-2024). (Accessed: August 1, 2025). https://www.jpl.nasa.gov/missions/deep-space-optical-communications-dsoc/.

[35] C. Bormann and P. Hoffman. *Concise Binary Object Representation (CBOR).* IETF RFC 8949 (2020). https://www.rfc-editor.org/rfc/rfc8949.

[36] International Earth Rotation and Reference Systems Service (IERS). *Bulletin C (Leap Second Announcements).* (Accessed: August 1, 2025). https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html.

[37] J. Dilley et al. *Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks.* arXiv:1612.05491 (2016). https://arxiv.org/abs/1612.05491.

[38] Consultative Committee for Space Data Systems (CCSDS). *Time Code Formats and Interoperability.* CCSDS 301.0-B-4, Blue Book (2010). https://public.ccsds.org/Pubs/301x0b4e1.pdf.