Satellite-Assisted Relay for Browser-Native Light Clients in Proof-First Dual-Ledger Networks:

Transport Feasibility Notes

Abstract

This paper examines the feasibility of using satellite communication links to relay transactions, proofs, and consensus metadata for a proof-first, dual-ledger blockchain architecture that supports browser-native light clients. We analyze how satellite relays can augment peer-to-peer networking without altering consensus rules, enabling participation in network-constrained environments. The analysis focuses on bandwidth requirements, latency tolerance, availability improvements, and security boundaries, with particular attention to architectures that separate execution from ordering and rely on compact state commitments rather than global execution replay.

Status Note

> *This document evaluates transport-layer feasibility under existing architectural constraints. It is not a protocol specification, implementation proposal, or security analysis. The analysis assumes prior work on compact verification models (minimal state frontiers, bounded inclusion mechanisms) and does not re-derive those foundations.*

1. Introduction

Traditional blockchain light client participation assumes reliable terrestrial internet connectivity and long-lived TCP/IP connections. This assumption degrades or excludes participation for users in network-constrained regions.

Lightweight verification models and browser-capable runtimes motivate revisiting satellite networks as transport-layer relays. This paper evaluates bandwidth, latency, and availability characteristics of satellite links for:

- Transaction submission from network-constrained clients
- Distribution of consensus headers and state commitments
- Retrieval of compact verification proofs

Satellite relays operate as untrusted transport endpoints and introduce no additional trust assumptions.

2. Architectural Assumptions

This analysis assumes a blockchain architecture with:

- Dual-ledger structure separating local state transitions from global ordering
- Header-first verification, where consensus commits to state deltas rather than execution traces
- Compact proof artifacts, verifiable without global state or Merkle tree traversal
- Deterministic submission windows, avoiding real-time fee markets

Compact proofs and bounded verification depend on mechanisms for bounded inclusion and minimal state frontier verification. These are prerequisites, not contributions of this work.

Satellite relays operate at the network transport layer only. They are equivalent to untrusted peer-to-peer network nodes.

2.1 Explicit Non-Goals

This analysis does not address:

- **Privacy:** Satellite broadcasts are observable by definition. Traffic analysis and metadata leakage are out of scope.
- **True censorship resistance:** Satellite operators can withhold data. This analysis examines availability improvement, not censorship elimination.
- **Consensus modifications:** No changes to block production, finality, or validator behavior are proposed or required.
- **Trust reduction:** Satellites do not reduce trust assumptions. All data is verified client-side regardless of transport.

3. Satellite Relay Model

3.1 Role Definition

Satellites function as:

- Broadcast relays for consensus metadata (headers, state commitments)
- Uplink relays for transaction and proof submission
- Redundant transport options, not authoritative sources

Satellites do not:

- Validate transactions or proofs

- Participate in consensus
- Produce or sign blocks
- Generate or attest to proofs

3.2 Threat Model

Satellites are untrusted. A malicious or compromised satellite operator may:

- Withhold data selectively or entirely (denial of service)
- Inject invalid or malformed data (rejected deterministically by clients)
- Log and analyze traffic patterns (metadata exposure)
- Delay transmission arbitrarily within physical constraints

Satellites cannot cause false acceptance of invalid state. All data is verified cryptographically by clients upon receipt. Satellite misbehavior affects availability and liveness only.

Critical limitation: Clients relying exclusively on satellite transport remain vulnerable to targeted data withholding. Detection requires comparison with alternative data sources. Prevention is impossible at the transport layer. Recovery requires access to at least one honest relay (satellite or terrestrial).

3.3 Data Types and Sizes

Let:

- H = consensus header size (bytes)
- P = compact proof bundle size (bytes)
- T = transaction payload size (bytes)

For browser-native architectures with bounded verification:

- $H \approx$ 200–500 bytes
- $P \approx$ 1–10 KB
- $T \approx$ 200–1000 bytes

These values fit within satellite broadcast constraints.

4. Bandwidth Analysis

Assume:

- N concurrent receiving clients

- Header broadcast frequency f (Hz)

Total downstream bandwidth:

$$B_{\text{down}} = N \cdot H \cdot f$$

Example calculation:

- $N = 10^6$
- $H = 300$ bytes
- $f = 0.1$ Hz (one header per 10 seconds)

$B_{\text{down}} = 10^6 \times 300 \times 0.1 = 30$ MB/s

Modern satellite broadcast systems support this bandwidth, particularly with IP multicast. Uplink bandwidth (transactions, proof requests) is orders of magnitude smaller and bursty, making contention manageable.

5. Latency Analysis

Satellite round-trip time (RTT):

- Low Earth Orbit (LEO): 30–60 ms
- Geostationary Orbit (GEO): 500–700 ms

The assumed architecture:

- Does not require interactive handshakes for inclusion
- Does not rely on real-time fee auctions
- Accepts delayed submission within deterministic windows

Latency affects time-to-inclusion, not correctness. Let Δ = submission window duration. If $\Delta \gg$ RTT, satellite latency does not block inclusion.

6. Availability in Adversarial Networks

Satellite relays provide alternative transport routes, improving availability in scenarios where terrestrial connectivity is degraded or filtered:

- Network-independent header distribution
- Alternative uplink paths for transaction submission

• Bypass of terrestrial routing and filtering infrastructure

Let p_t = terrestrial connectivity failure probability and p_s = satellite connectivity failure probability. If failures are independent:

$$A = 1 - (p_t \cdot p_s)$$

Independent failure modes improve overall availability. However, this does not eliminate targeted censorship—a determined adversary controlling both terrestrial and satellite infrastructure can still block participation.

7. Security Analysis

7.1 Trust Boundaries

Satellite relays are untrusted transport endpoints:

• All data is verified cryptographically upon receipt

• Invalid data is rejected deterministically

• Satellite transport introduces no additional trust assumptions

7.2 Attack Taxonomy

Eclipse attacks: Clients using satellite transport exclusively remain vulnerable to eclipse. Withholding is *detectable* via peer comparison but not *preventable* at the transport layer. *Recovery* requires access to alternative data sources. Satellite transport improves availability but does not eliminate eclipse risk.

Data withholding: Detectable via cross-validation with other relays. Does not compromise safety if at least one honest relay (satellite or terrestrial) is accessible.

Spam and malformed data: Rejected deterministically based on cryptographic verification. May consume bandwidth and client processing resources but cannot introduce invalid state.

Traffic analysis: Satellite broadcasts are observable. Metadata (timing, volume, source patterns) may be logged and analyzed. This is an inherent property of broadcast transport, not a vulnerability.

No consensus-level attacks are introduced. Satellite transport does not weaken existing security properties.

8. Failure Modes (Non-Adversarial)

Non-adversarial failures affect liveness, not safety:

Temporary proof unavailability: Satellite downtime prevents proof retrieval. Does not invalidate existing state or introduce false data. Clients await restoration or switch to terrestrial relays.

Header propagation lag: Delayed broadcasts increase synchronization latency. Clients verify headers upon receipt; delays do not compromise correctness.

Submission deadline misses: Transactions relayed via satellite may miss inclusion windows due to latency. Clients can retry or use terrestrial uplinks.

Partial desynchronization: Clients relying exclusively on satellite feeds may lag behind the global chain tip during interruptions. Normal synchronization resumes upon connectivity restoration.

These failures degrade user experience but do not compromise protocol safety.

9. Browser Client Implications

Satellite transport enables browser clients to:

- Receive consensus headers without RPC dependencies
- Verify state transitions independently
- Submit transactions without persistent connections

Browser clients function as verifying peers rather than passive RPC consumers. This is particularly relevant for:

- Mobile-only users without fixed terrestrial connectivity
- Emergency or disaster recovery scenarios
- Jurisdictions with degraded internet infrastructure

10. Discussion

Satellite relays are not a scaling mechanism. Their value is limited to architectures that:

- Minimize global state requirements
- Commit to facts rather than execution traces
- Enable bounded verification workloads

In such systems, satellite transport augments—but does not replace—terrestrial peer-to-peer networks. The primary benefit is availability improvement in adversarial network conditions, not performance or cost reduction.

11. Conclusion

Satellite communication is feasible as a transport-layer extension for proof-first, dual-ledger blockchain architectures. When combined with compact proofs and header-based verification, satellite relays enable participation in network-constrained environments without altering consensus rules or introducing trusted intermediaries.

Feasibility is not limited by cryptography or available bandwidth, but by architectural discipline. Only systems designed for bounded verification and compact state commitments can benefit from satellite-assisted transport.