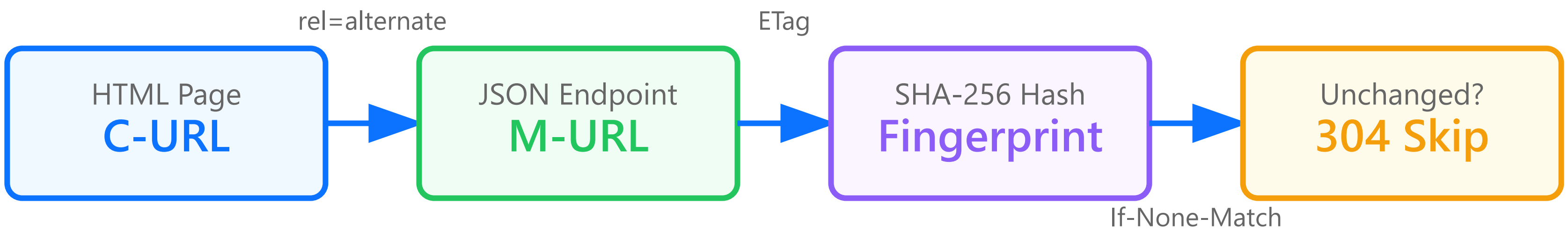
🌱 TCT: Application-Layer Efficiency for Web Crawling

Measured use of existing HTTP semantics (RFC 9110/9111)

✅ **IETF Draft Published:** draft-jurkovikj-collab-tunnel-00

📅 Published: November 4, 2025 · Expires: May 8, 2026



3 production sites · ~970 URLs · Multi-week deployment · Public validator

❌ The Problem: Wasted Bytes = Wasted Joules

Every byte transferred over the network consumes energy at multiple layers:

- **Network transmission:** 0.03-0.14 kWh/GB (datacenter + path)
- **Server generation:** CPU cycles to render HTML, CSS, JavaScript
- **Client parsing:** Energy to parse, extract, normalize content
- **Redundant fetches:** Re-downloading unchanged content wastes all three

83%

of crawled bytes are **HTML markup overhead**

CSS, JavaScript, navigation, ads — not the actual content

🎯 **Goal:** Eliminate wasted bytes at the protocol layer, not just compress them.

17 KB HTML: 103 KB (avg) 83% = Markup Overhead

JSON: 17.7 KB ← Just the content

💡 Two Layers of Savings

Layer 1: Efficient Payloads

JSON instead of HTML

- 17.7 KB vs 103 KB (gzipped)
- No CSS, JavaScript, ads
- Just structured content
- **83% bandwidth reduction**

Server-side fingerprinting

- SHA-256 from normalized content
- Exposed as ETag header
- Template-invariant (survives theme changes)

Layer 2: Zero-Fetch Optimization

Sitemap-first discovery

- Fetch sitemap once
- Compare contentHash to cached ETags
- Skip unchanged URLs **entirely**
- No HTTP request = zero bytes

304 Not Modified fallback

- If-None-Match conditional requests
- Server returns 304 (no body)
- RFC 9110/9111 semantics

Combined Effect: 83% fewer bytes × 90% skip rate = ~98% bandwidth elimination

🔄 Layer 1 reduces size, Layer 2 reduces frequency

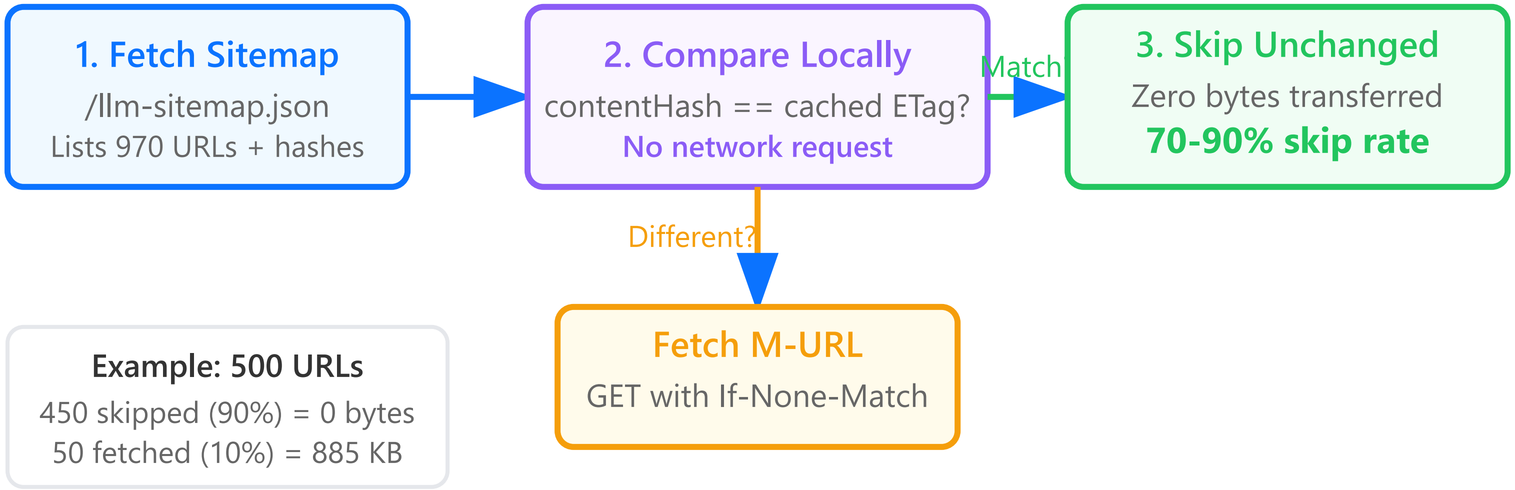
Both use existing HTTP semantics

🚀 Zero-Fetch Innovation

Traditional HTTP caching requires **one request per URL** to check freshness.

TCT's innovation: Check freshness for **all URLs at once** via sitemap.

Sitemap-First Flow (Zero-Fetch)



💡 **Key Insight:** Sitemap provides batch freshness checks. One sitemap fetch (~50 KB) replaces hundreds of conditional requests.

📊 Quantifying the Impact: From Bytes to Kilowatt-Hours

Bandwidth Savings

HTML → JSON: **83% reduction**

103 KB → 17.7 KB (gzipped avg)

Skip Rate

Zero-fetch: **70-90% skip**

Steady-state revisits (multi-week)

Production Scale

970 URLs across 3 sites

WordPress + LiteSpeed + TCT plugin

Network Energy

0.03-0.14 kWh/GB (cited)

Conservative midpoint: 0.06 kWh/GB

Energy Impact Chain

Bytes Saved
83% reduction
× 90% skip
= 98% eliminated

→

Network Tx
0.06 kWh/GB
(datacenter + path)
Direct & measurable

→

Server Gen
CPU to render
HTML vs JSON
Deployment-dependent

→

Client Parse
Extraction
Model-dependent

⚠️ Network transmission savings are direct and measurable. Server + client impacts depend on infrastructure and models.

🏗️ Full-Stack Approach: Collaboration with GREEN WG

TCT operates across the full web stack — each layer offers efficiency opportunities:

Application Layer
TCT: JSON payloads, sitemap discovery, content fingerprinting

HTTP Semantics Layer
RFC 9110/9111: ETags, 304 Not Modified, Cache-Control, If-None-Match

Transport Layer
TLS 1.3, HTTP/2, compression (gzip/brotli), connection reuse

Network/Infrastructure

🤝 GREEN WG Collaboration Opportunities:

- **Energy Reporting Standards:** How should protocol drafts quantify efficiency?
- **Measurement Methodology:** Bytes/URL? 304 rates? kWh conversions?
- **Cross-Layer Interactions:** How do application-layer optimizations interact with transport/infrastructure?
- **Best Practices:** Should GREEN document common HTTP caching pitfalls for efficiency?

🎯 Next Steps & Call for Collaboration

✅ Completed

- IETF draft published (draft-jurkovikj-collab-tunnel-00)
- Production deployment (3 sites, 970 URLs)
- Open-source implementations (WordPress, Python)
- Public validator + measurement tools
- Multi-week stability testing

🔄 In Progress

- Expert review feedback integration
- Energy methodology refinement
- Additional reference implementations
- Larger-scale pilot deployments
- Interoperability testing

🤔 What We're Asking from GREEN WG:

1. Metrics Guidance

- Recommended efficiency metrics?
- Bytes/URL? 304 rate? Revisit cadence?
- Standardized measurement format?

2. Energy Proxies

- Bytes→kWh conversion acceptable?
- How to cite assumptions properly?
- What ranges to report?

3. Best Practices

- Document HTTP caching pitfalls?
- Vary header impacts on efficiency?
- ETag strength considerations?

4. Collaboration

- Interest in GREEN WG adoption?
- Joint work on measurement standards?
- Feedback on draft sections?

📚 Resources Available

📄 IETF Draft

draft-jurkovikj-collab-tunnel-00

https://datatracker.ietf.org/doc/draft-jurkovikj-collab-tunnel/

Published: November 4, 2025 · Expires: May 8, 2026

💻 Spec + Code

https://github.com/antunjurkovic-collab/collab-tunnel-spec

Reference implementation, WordPress plugin, Python validator

✅ Live Validator

https://llmpages.org/validator/

Test your site's TCT compliance

🧑 Contact

Antun Jurkovikj

antunjurkovic@gmail.com

Seeking collaboration with GREEN WG on energy reporting standards