On the Distribution of DTN Reachability Information: A Quantitative Push-vs-Pull Analysis

Juan A. Fraire*†,
*Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France
†D3TN GmbH, Dresden, Germany

***Abstract*—As space networks scale from isolated point-to-point links to multi-operator, multi-hop infrastructures, nodes must discover how to reach each other's Bundle Protocol endpoints. Today, this Endpoint Identifier (EID) to Convergence-Layer Adapter (CLA) binding is pre-provisioned; it will not scale. Two Internet paradigms offer competing solutions: proactive flooding (*à la* BGP) that replicates every binding everywhere, and on-demand resolution (*à la* DNS) that queries an authority when needed. We build OMNeT++ models of both and evaluate them across 13 experiments on synthetic grids (up to 400 nodes, 500 EIDs) and three realistic scenarios (terrestrial disaster response, lunar Artemis, and Mars exploration) covering one-way delays from 5 ms to 12 min. The results reveal a sharp, delay-driven regime map: below ∼100 ms round-trip time, DNS delivers 9–141× lower overhead with sub-200 ms query latency; above ∼3 s, BGP's one-time convergence cost amortizes after a single query, making it the only viable option at Mars distances; and a contested crossover emerges at cislunar delays (∼1–3 s), where a hybrid strategy may be optimal. Under EID churn, BGP maintains 100% accuracy at all rates, while DNS degrades to 33% when the churn interval falls below the cache TTL. These findings give the first quantitative guidelines for choosing, or combining, push and pull EID resolution across the solar system.**

***Index Terms*—Delay-tolerant networking, endpoint identifier resolution, BGP, DNS, space networking, cislunar communications, interplanetary Internet**

I. Introduction

Every packet on the Internet relies on two infrastructures so mature they are almost invisible: BGP floods reachability to every backbone router, while DNS resolves names on demand. They embody a decades-old design tension—*push everything now* versus *pull only when needed*—that the terrestrial Internet resolved pragmatically: routing tables are pushed, name bindings are pulled. On the Internet the two protocols are complementary and run side by side, each applying its paradigm to a different problem. What we compare in this paper are the paradigms themselves, applied to a single DTN problem for which either one is a candidate: distributing the binding between an endpoint identifier and the transport address that serves it.

Space networking is reopening this question. The Delay-Tolerant Networking (DTN) architecture [1], [2] introduces Bundle Protocol Endpoint Identifiers (EIDs) that must be mapped to Convergence-Layer Adapter (CLA) addresses before any bundle can be forwarded—the exact analogue of IP routing and DNS resolution. Yet the DTN standards deliberately leave this binding unspecified [3], and current deployments rely on static, pre-provisioned tables. Static configuration becomes untenable once programs such as Artemis [4] and future Mars architectures [5] reach dozens of mobile, multi-agency nodes.

Recent proposals have begun to fill this gap from both directions. Feldmann et al. [6] extend BGP UPDATE messages to carry DTN EID reachability, leveraging BGP's path-vector machinery; the original scope of that work targets *edge BPA configuration*—how a Bundle Protocol Agent is configured by its immediate gateway—rather than flooding EID bindings across a general multi-hop mesh. Kline [7] proposes the `ipn.arpa` DNS zone to resolve `ipn`-scheme EIDs through the existing DNS hierarchy. Both approaches are technically sound, but they inherit fundamentally different cost structures: BGP pays $O(N)$ messages per EID change for $O(1)$ lookups; DNS pays $O(1)$ per registration but $O(D \cdot d)$ latency per query, where D is the network diameter and d the link delay. In terrestrial networks, where d is milliseconds, this difference is negligible. In space, where d ranges from 1.3 s (Earth–Moon) to 12 min (Earth–Mars), the choice is existential.

No prior work has systematically compared push and pull EID resolution across the full DTN delay spectrum as a *general* multi-hop problem. This paper closes that gap—deliberately broadening the scope beyond the edge-gateway configuration originally targeted by [6]—with three contributions: (i) **simulation models**: OMNeT++ implementations of a BGP-like path-vector protocol and a DNS-like query-response protocol with caching, operating over identical topologies and publicly available [8]; (ii) **comprehensive evaluation**: 13 experiments spanning synthetic grids (25–400 nodes, 10–500 EIDs, 10 ms–20 s link delays) and three realistic hierarchical deployments covering churn, caching, and scalability; and (iii) **a delay-driven regime map**: DNS saves 9–141× in overhead below 100 ms RTT, BGP is the only viable option above 3 s, and a contested cislunar band in between calls for hybrid strategies.

II. Background

A. Delay-Tolerant Networking and the EID Resolution Gap

The DTN architecture [9], [1] introduces a store-carry-forward overlay tolerating intermittent links, high delays, and asymmetric rates. Its core is the Bundle Protocol (BPv7, RFC 9171 [2]), which interfaces with heterogeneous transports through *Convergence-Layer Adapters* (CLAs) [10]; each endpoint is identified by a URI-style *Endpoint Identifier* (EID,

e.g., `ipn:13.1`) that must be resolved to a concrete CLA address before a bundle can be forwarded. DTN is operational on the ISS and standardized by CCSDS [11], [12], and LunaNet [4] and Mars architectures [5] will demand interoperation among dozens of multi-agency assets. Yet the architecture leaves the EID-to-CLA *binding* unspecified [3]; in practice it is pre-provisioned, which does not scale to dynamic, multi-operator networks [13].

Under *late binding*, a bundle carries only a destination EID and the full mapping (EID → reachable node → next contact → convergence-layer peer) is resolved at each hop at forwarding time [1]. Viewing DTN as an overlay interconnecting islands of conventional networks, a service such as BGP or DNS can resolve the next-hop CLA parameters within each island; the question studied here is how such per-domain services should operate under the extreme, heterogeneous delays of space deployments.

B. BGP: Push-Based Reachability Distribution

BGP-4 [14], the Internet's inter-domain routing protocol, is a *path-vector* protocol operating on a *push* model: new prefixes are announced to all peers and flooded until every speaker converges, within seconds to minutes [15]. Its strengths are instant lookup (post-convergence resolution is a local table read) and strong consistency; its weaknesses are state explosion and update storms on every change. In DTN these traits are amplified, since convergence time scales as $D \cdot d$ for diameter D and link delay d while post-convergence lookups stay instantaneous.

Feldmann et al. [6] adapt BGP for DTN EID distribution by extending UPDATE messages with DTN-specific attributes, targeting the configuration of end-user-facing BPAs by their gateways: local state at a single administrative boundary. We generalize that push paradigm to multi-hop deployments so it can be compared with the DNS pull approach; our results therefore characterize *generalized* flooding, and leave the original, narrower edge-gateway use case untouched.

C. DNS: Pull-Based Name Resolution

DNS [16] resolves names on demand: a recursive resolver traverses the authority hierarchy only when a client issues a query, and caches responses for a record-specified Time-to-Live (TTL), which removes 60–90% of authority queries under typical web traffic [17], [18]. The pull model offers minimal background overhead and centralized state, at the cost of per-query latency on every cache miss: under 100 ms terrestrially, seconds (cislunar) to minutes (interplanetary) in space. The `ipn.arpa` proposal [7] maps `ipn`-scheme EIDs to DNS resource records, allowing space networks to reuse the global DNS ecosystem.

DNS *zone replication*—synchronizing authority databases to remote sites—could eliminate per-query latency, but only by turning DNS into a push model in which every binding change propagates across interplanetary links; the tradeoff studied here is a paradigm-level one. Deeply remote networks such as a Mars surface mesh will in any case form separate administrative domains with locally managed namespaces, serving most queries without interplanetary round-trips.

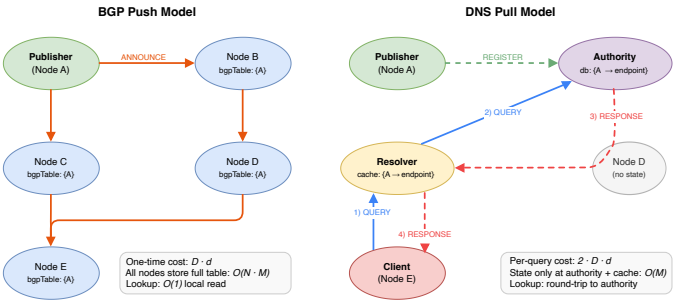

Fig. 1. Push vs. pull EID resolution. *Left:* BGP-like flooding replicates every binding to every node (one-time cost $D \cdot d$, $O(N \cdot M)$ network state, $O(1)$ lookup). *Right:* DNS-like resolution stores bindings at an authority (grey nodes hold no EID state) and resolves on demand (per-query cost $2 \cdot D \cdot d$, $O(M)$ centralized state).

III. Model

Push protocols pay an upfront cost to replicate state everywhere, enabling instant local lookups; *pull* protocols avoid replication overhead but pay a per-query round-trip cost (Fig. 1). We implement both in OMNeT++ [19] over identical physical topologies, with the code and all experiment configurations publicly available [8].

A. System and Network Model

We consider a DTN of N nodes interconnected by links with propagation delay d, hosting a set $\mathcal{E} = \{e_1, \ldots, e_M\}$ of EIDs. For each EID e_k there is exactly one authoritative binding $\text{endpoint}(e_k) = \langle \text{nodeId}, \text{claProtocol}, \text{port} \rangle$. Any node must resolve $\text{endpoint}(e_k)$ under three conflicting objectives that no single protocol satisfies at once: **correctness** (matching the current authoritative binding, even under churn), **timeliness** (acceptable resolution latency), and **efficiency** (low overhead and state on bandwidth-constrained links). Nodes assume the roles of *publishers* (originate EID bindings), *clients* (resolve EIDs), and, for DNS only, *resolvers* (caching intermediaries) and *authorities* (authoritative record stores). Bindings are not permanent: assets are deployed or retired and endpoints move between platforms, so a publisher may withdraw an EID or rebind it to a new endpoint at any time. We call this ongoing rate of binding changes *churn*, and parameterize it by the mean interval Δ_{churn} between successive changes at a publisher.

1) Synthetic Grid Topology: The primary controlled topology is a $w \times h$ grid with nodes connected to their horizontal and vertical neighbors, giving diameter $D = w + h - 2$ (e.g., $D = 18$ for 10×10). We vary grid size from 5×5 to 20×20 and sweep link delay from $d = 10$ ms (terrestrial) through $d \in \{1, 5, 10, 20\}$ s (deep-space). Links are delay channels; transmission time is abstracted away, which is appropriate for small control-plane messages.

2) Realistic Hierarchical Topologies: To validate the grid findings, we model three hierarchical deployments spanning the DTN latency spectrum (Fig. 2): terrestrial disaster response (22 nodes, 5–30 ms), lunar Artemis (12 nodes, 5 ms–1.3 s),

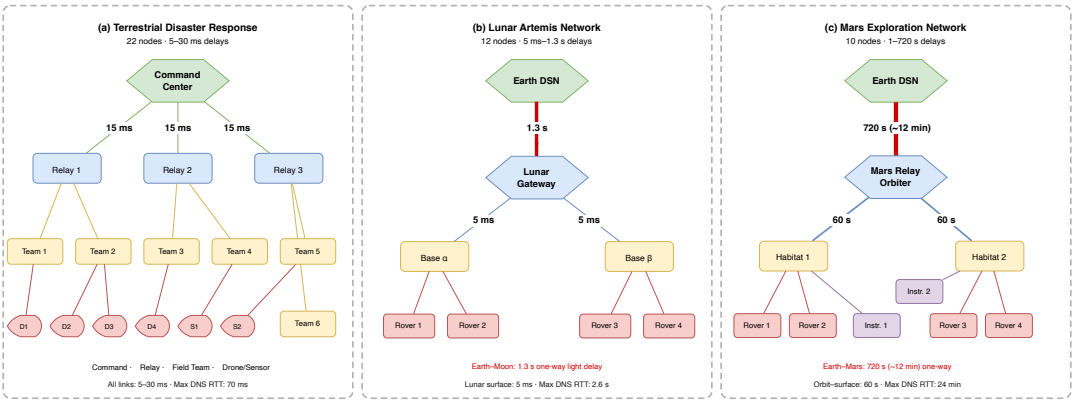

Fig. 2. Realistic deployment topologies: (a) terrestrial disaster response, 4-tier hierarchy with 5–30 ms links; (b) lunar Artemis, where the 1.3 s Earth–Moon link dominates latency; (c) Mars exploration, with a 720 s interplanetary link and 60 s orbit-to-surface links.

and Mars exploration (10 nodes, 1–720 s), detailed in Section IV-F. In all three, links are modeled as *stable, always-on delay channels*—no disruptions or contact schedules. This deliberate simplification isolates the *control-plane overhead question* from the *disruption-tolerance question*: results characterize protocol behavior given established connectivity. Real space links are of course interrupted, and the two paradigms absorb an interruption differently: a flooded update waits once at a closed trunk and then propagates, whereas every pull query crossing that trunk waits again. Complementary runs in the public repository [8], in which the lunar and Mars trunks follow a periodic contact schedule of duty cycle δ, confirm the asymmetry: at $\delta = 0.25$ the mean latency of Earth-involved lunar queries inflates by up to $81\times$, while BGP overhead is byte-identical across duty cycles and its accuracy stays at 100%. The always-on results below are therefore a best case for pull, and adding interruptions moves the crossover of Section IV-G toward push. Integrating contact-plan-aware routing with EID resolution is left as future work.

B. BGP-like Push Model

The push model implements a path-vector protocol inspired by BGP-4 [14]: EID bindings are flooded to all nodes upon publication, and each node maintains a routing table (`bgpTable`) mapping each known EID to its origin, next hop, path length, version number, endpoint binding, and full path vector. Two message types are used: **BgpAnnounce** ($64 + 4|\text{path}|$ bytes), carrying EID reachability with the path vector, and **BgpWithdraw** (32 bytes).

1) Operation: When a node publishes EID e, it creates a local entry (path length 0), increments the per-EID version, and floods an announcement to all neighbors. A receiving node discards announcements containing itself in the path vector (loop detection); otherwise it accepts the update if (i) the EID is unknown, (ii) the version is newer from the same origin, (iii) the path is shorter, or (iv) it ties an equal-length path with a lower origin ID, then re-floods it on all gates except the arrival gate. Withdrawals are accepted only from the entry's origin with version $\geq$ the stored one, then removed and forwarded.

2) Complexity: For diameter D and link delay d, convergence takes $T_{\text{conv}}^{\text{BGP}} = D \cdot d$, e.g., 0.18 s on a 10×10 grid at $d = 10$ ms, and this cost is paid *once* per EID publication; all subsequent lookups are local table reads. Per-node state is $O(M)$ and network-wide state $O(N \cdot M)$ (full replication), message cost is $O(N)$ per publication, and sustained overhead under churn is $O(N \cdot M/\Delta_{\text{churn}})$, which we expect to dominate at high churn rates.

C. DNS-like Pull Model

The pull model implements on-demand resolution inspired by DNS [16]. Publishers register bindings at authoritative nodes (assigned by modular hashing over the authority set); resolvers cache responses with TTL-based expiry. Four message types are used: **DnsQuery** (48 bytes), **DnsResponse** (80 bytes), **DnsRegister** (72 bytes), and **DnsDeregister** (24 bytes).

1) Operation: A client query for EID e first checks the local cache; on a miss it is forwarded to the resolver, which checks its own cache and, if necessary, forwards to the responsible authority. The response traverses the reverse path, with resolvers caching the result for its TTL τ. When a cache exceeds capacity, expired entries are evicted first, then the entry with the earliest expiry.

2) Complexity: A cache-miss query costs $L_{\text{query}} \approx 2 \cdot D \cdot d$—exactly $2\times$ the BGP convergence time, but paid on *every* uncached resolution. For Q queries to distinct EIDs, cumulative DNS latency is $Q \cdot 2Dd$ versus BGP's fixed Dd, so BGP amortizes its convergence cost after a single query in the worst case. Authority state is $O(M)$, resolver caches $O(\min(M, C))$ for capacity C, and network-wide state $O(M)$, independent of N; registration and queries cost $O(1)$ messages. Churn updates the authority locally with no network-wide propagation, but cached entries go stale, giving a freshness–efficiency tradeoff governed by τ.

D. Fair Comparison Design

DNS in practice operates over an already-routed IP network, so routing its queries hop-by-hop over a not-yet-converged network would penalize it for a problem orthogonal to our question. We therefore run DNS in a "direct mode" using OMNeT++'s `sendDirect()` with RTT $= 2 \cdot D \cdot d$: BGP announcements propagate hop-by-hop through the physical topology, while DNS queries logically traverse Client $\rightarrow$

TABLE I
PROTOCOL COMPLEXITY COMPARISON

Aspect	BGP (Push)	DNS (Pull)
State per node	$O(M)$	$O(1)$ client, $O(C)$ resolver
Network state	$O(N \cdot M)$	$O(M)$
Publication cost	$O(N)$ messages	$O(1)$ messages
Query cost	$O(1)$ (local)	$O(D)$ hops
Churn cost	$O(N)$ per change	$O(1)$ per change
Convergence time	$O(D \cdot d)$	N/A (on-demand)
Discovery latency	$D \cdot d$	$2 \cdot D \cdot d$
Resolution latency	≈ 0 (local)	$2 \cdot D \cdot d$

Resolver $\rightarrow$ Authority and back, isolating the *information distribution strategy* from lower-layer routing.

E. Staleness, Accuracy, and Convergence

A central `GroundTruth` module maintains authoritative records for all EIDs, enabling unbiased correctness verification. A node's BGP entry is *correct* if it matches the ground-truth endpoint (or is absent for withdrawn EIDs), and network-wide accuracy is the fraction of nodes with correct entries; because updates are flooded incrementally, we expect BGP accuracy to stay at 1 after initial convergence at any churn rate. A DNS response is *stale* if the cached record predates the last authoritative change; for TTL $\tau > \Delta_{\text{churn}}$, $P(\text{stale}) \approx \Delta_{\text{churn}}/\tau$, so shorter TTLs improve freshness at the cost of authority query load. An EID is *converged* when $\geq$99% of nodes hold correct information, and a critical regime arises once the *convergence-to-churn ratio* $R = T_{\text{conv}}/\Delta_{\text{churn}}$ exceeds 1, since the network then cannot fully converge between successive changes. Client query patterns follow a Zipf distribution with skewness α ($\alpha = 0$: uniform; $\alpha = 1$: typical web traffic [18]).

F. Complexity Comparison and Hypotheses

Table I compares both approaches: BGP's $O(N \cdot M)$ state and $O(N)$ per-change churn cost grow with network size and dynamism, favoring DNS, while DNS's $O(D \cdot d)$ per-query cost grows with propagation delay, favoring BGP. This motivates four testable hypotheses:

- **H1**: *DNS dominates in low-latency regimes* ($2Dd < 1$ s), where its $O(1)$ publication and churn costs yield lower overhead than BGP.
- **H2**: *BGP dominates in high-latency regimes*: once $2Dd$ exceeds tens of seconds, the cumulative DNS cost ($Q \cdot 2Dd$) exceeds BGP's one-time convergence after as few as one query.
- **H3**: *A crossover exists near cislunar delays* ($\sim$1–2 s one-way), where DNS retains an overhead advantage while BGP provides lower amortized latency.
- **H4**: *DNS accuracy degrades with the convergence-to-churn ratio*: for $R > 1$, cache invalidation cannot keep pace with EID changes, while BGP stays correct at any R.

IV. EVALUATION

We evaluate both protocols through 13 experiments in three groups: synthetic grid experiments (Sections IV-A–IV-D), deep-space experiments (Section IV-E), and realistic scenarios (Section IV-F). All experiments use the direct-mode DNS underlay of Section III-D. Fig. 3 previews the delay-driven regime map that synthesizes the findings.

Parameters and reproducibility. Message sizes are those of Sections III-B and III-C, DNS TTLs range from 15 to 300 s, and caches are unbounded except in Exp. 9, which contrasts a 100-entry cache with none at all. Churn is modeled at the publishers: every Δ_{churn}, each published EID changes with probability 0.1–0.5, and 10–30% of those changes are withdrawals while the rest rebind the EID to a new endpoint. Client queries are drawn from a Zipf distribution with $\alpha \in [0, 2]$. Every configuration is repeated 3–10 times with the seed set taken from the repetition index; across repetitions, overhead and latency vary by at most 11% and accuracy by at most 5 percentage points, and the flooding and query-volume metrics of the churn-free experiments are seed-invariant. The complete `.ini` files, the models, and the analysis scripts are available for research and educational use in the companion repository [8].

A. Baseline Overhead and Latency

We first characterize both protocols on a 10×10 grid (100 nodes, $d = 10$ ms) with 50 EIDs and no churn. **Overhead:** BGP generates 1,355 KB of control traffic to flood announcements to all nodes, versus 9.6 KB for DNS (100 client queries × 80-byte responses)—a 141× ratio. **State:** BGP replicates the full table everywhere (5,000 entries network-wide); DNS stores 50 records at the authority plus the resolver cache. **Discovery latency (Exp. 4):** BGP converges in 0.20 s, after which lookups are instantaneous; DNS resolves in 0.25 s per round-trip—BGP's cost is *one-time*, while DNS pays on *every* resolution.

B. Scalability

Fig. 4 shows overhead scaling with network size (Exp. 2) and EID count (Exp. 3). **Network size:** growing the grid from 25 to 400 nodes increases BGP overhead from 236 KB to 8.10 MB—34× for a 16× node growth, confirming super-linear $O(N^2)$ scaling—while DNS remains constant at 8 KB, depending only on query count. **EID count:** on a fixed 10×10 grid, increasing EIDs from 10 to 500 grows BGP overhead from 280 KB to 13.56 MB ($\sim$27 KB per EID), with network-wide state reaching 50,000 entries versus 500 at the DNS authority; DNS overhead stays at 8 KB.

C. Churn Resilience

Experiments 5 and 6 evaluate both protocols under EID churn on the 10×10 grid with 20 EIDs (Fig. 5). **Overhead:** at a 5 s churn interval, BGP generates 5.6 MB of update traffic—233× DNS's constant 24 KB—since every change triggers a network-wide flood, while churn only modifies the DNS authority's local database. **Accuracy:** BGP maintains 100% at all churn rates; DNS degrades to 33% correct responses at 5 s churn with 60 s TTL, and recovers to 89% once the churn interval matches the TTL. **Staleness vs. TTL (Exp. 6):** under

10 s churn, shorter TTLs improve accuracy (83% at TTL=15 s vs. 35% at TTL=120 s) at the cost of query load; the optimal TTL approximates the expected churn interval.

D. DNS Cache Effectiveness

Experiment 7 evaluates DNS caching under Zipf-distributed queries (1,000 queries over 100 EIDs, 60 s TTL). The cache absorbs 44.7% of lookups under uniform queries ($\alpha = 0$), 46.1% at $\alpha = 1.0$ (typical web traffic [18]), and 48.7% at $\alpha = 2.0$. With a query rate far above the TTL expiry rate, temporal reuse dominates and access skew contributes little: the hit rate is set by the ratio of TTL to query interval, which is the parameter an operator should tune.

E. Deep-Space Scenarios

Experiments 8–10 use a 5×5 grid ($D = 8$) with link delays from 10 ms to 20 s (Fig. 6). **Latency scaling:** both BGP convergence and DNS query RTT scale linearly with delay; at $d = 20$ s BGP converges once in 160 s while each DNS query takes 400 s, so BGP's "pay once, use forever" model wins for any workload beyond a single query. **Caching at high delay (Exp. 9):** with Zipf $\alpha = 1.0$ and 300 s TTL, DNS sustains a 45.1% hit rate independent of delay, cutting mean query latency by 41%—at $d = 20$ s, DNS still averages 235 s per query versus BGP's 0 s after convergence. **Churn at high delay (Exp. 10):** when $R < 1$, DNS maintains 95–98% accuracy; when $R > 2$, it degrades to 80–88%, while BGP holds 100% at all ratios because incremental updates propagate before global convergence completes.

F. Realistic Deployment Scenarios

We now evaluate the three deployments of Fig. 2, with DNS TTLs of 30 s (terrestrial), 60 s (lunar), and 120 s (Mars) and churn intervals of 30–60 s, 120 s, and 300 s, respectively.

1) Terrestrial Disaster Response (Exp. 11): This scenario models 22 nodes in a 4-tier hierarchy with 5–30 ms links, representing a field network in which IP connectivity has been established, e.g., via mesh radios or satellite uplinks. Under 30 s churn, BGP generates 212.8 KB—13× DNS's 16.4 KB; under 60 s churn the ratio is 9× (120.1 vs. 12.9 KB) (Fig. 7a). DNS query latency averages an operationally acceptable 188 ms and both protocols achieve ≥96.7% accuracy, making DNS the preferred choice here.

2) Lunar Artemis Network (Exp. 12): The lunar scenario spans Earth and Moon with 12 nodes, where the 1.3 s Earth–Moon one-way delay creates the interesting tradeoff. BGP converges in ∼1.6 s (one-time), while DNS pays ∼0.6 s per lunar-local query and ∼2.8 s for Earth-involved queries; the *break-even point* occurs at ∼2.7 queries (Fig. 7b). DNS uses 3.8× less bandwidth (6.3 vs. 23.8 KB) and both achieve 100% accuracy at 120 s churn. This contested regime suggests a hybrid: DNS for infrequent lunar-local queries, BGP for Earth-involved services.

3) Mars Exploration Network (Exp. 13): The Mars scenario models 10 nodes with a 720 s Earth–Mars one-way delay and 60 s orbit-to-surface links. BGP converges in ∼2 min (one-time), after which lookups are instantaneous; DNS pays ∼4 min per Mars-local query and ∼25 min for Earth-involved queries—after 10 queries, DNS accumulates 40 min of delay versus BGP's fixed 2 min (Fig. 7c). DNS overhead is 3.4× lower (5.3 vs. 18.0 KB), but this is irrelevant when each query takes minutes. BGP is the only viable option at Mars distances.

G. Cross-Scenario Analysis

Fig. 3 synthesizes the results into a delay-driven decision map, linking each delay band to the confirmed hypothesis and realistic scenario. Three deployment regimes emerge:

Terrestrial (<100 ms RTT), H1 confirmed: DNS is preferred, since overhead savings of 9–141× dominate and query latency (<200 ms) is operationally acceptable. **Cislunar (100 ms–3 s RTT), H3 confirmed:** contested, as DNS retains a 3.8× overhead advantage while BGP amortizes convergence after ∼3 queries; a hybrid strategy may be optimal. **Deep-space (>3 s RTT), H2 confirmed:** BGP is required, because DNS per-query latency stays prohibitive even with caching (a 41% reduction still leaves 235 s per query at $d = 20$ s) and the break-even point occurs after a single query at Mars distances. **H4** is confirmed across all churn experiments (Fig. 6b): DNS accuracy degrades as R grows from 0.33 to 8.0, while BGP maintains 100% throughout.

V. Conclusions

This paper presented the first systematic comparison of push (BGP-like) and pull (DNS-like) paradigms for distributing EID reachability in DTNs, spanning one-way delays from 5 ms to 12 min across 13 experiments; all four hypotheses were confirmed. The results yield a practical design rule: *the one-way link delay is the single most predictive variable*

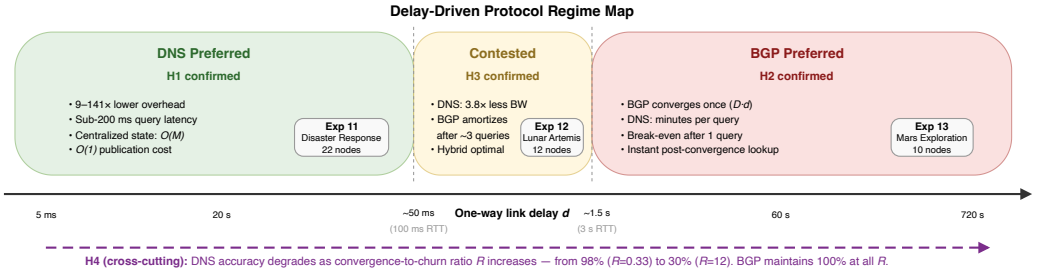

Fig. 3. Delay-driven protocol regime map: DNS preferred (<50 ms, H1), contested/hybrid (50 ms–1.5 s, H3), and BGP preferred (>1.5 s, H2). The cross-cutting arrow (H4) indicates that DNS accuracy degrades with the convergence-to-churn ratio R in all regimes, while BGP is unaffected.

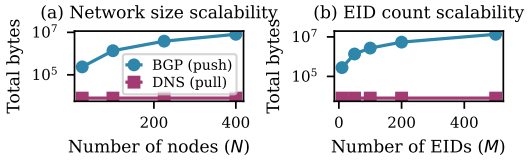

Fig. 4. Scalability of network overhead. (a) BGP grows super-linearly with node count ($O(N^2)$ in a grid) while DNS stays constant. (b) BGP scales linearly with EID count ($O(N \cdot M)$); DNS depends only on the query count.

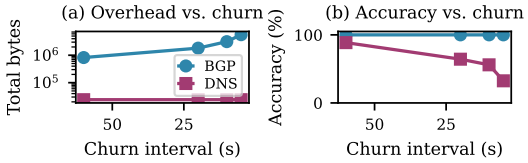

Fig. 5. Churn resilience. (a) BGP overhead scales inversely with churn interval (34–233× more than DNS). (b) BGP maintains 100% accuracy at all churn rates; DNS accuracy degrades to 33% (5 s interval) with 60 s TTL.

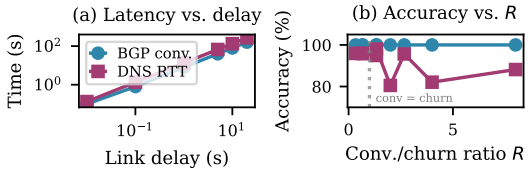

Fig. 6. Deep-space experiments. (a) BGP convergence and DNS query latency both scale linearly with link delay, DNS paying $2 \times D \times d$ per query against BGP's one-time $D \times d$. (b) DNS accuracy degrades once R exceeds 1 (vertical line); BGP stays at 100%.

for protocol selection. Operators can use the regime map of Fig. 3, DNS below ∼100 ms RTT, BGP beyond ∼3 s and hybrids in between, keeping in mind that the BGP conclusions apply to the generalized flooding model evaluated here and not to the edge gateway-to-BPA configuration targeted by [6], and that DNS accuracy requires the TTL to track the churn interval. Future work includes a concrete hybrid protocol for the cislunar band, hierarchical DNS authorities, integration with contact-plan-based routing [20], and the overhead of security mechanisms such as BPSec [21].

Acknowledgment

This work was funded by the DARKSOL project (https://darksol.cloud/en/), co-financed by the project partners, the European Union, and the Free State of Saxony.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 27–34.

[2] S. Burleigh, K. Fall, and E. Birrane, "Bundle Protocol Version 7," RFC 9171, January 2022.

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC 4838, April 2007.

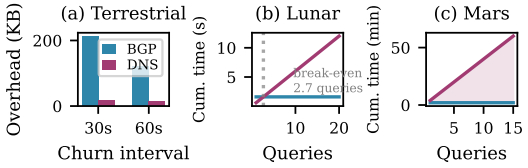

Fig. 7. Realistic deployment scenarios; the legend in (a) holds throughout. (a) Terrestrial disaster response: DNS achieves 9–13× lower overhead. (b) Lunar network: BGP amortizes convergence after ∼3 queries. (c) Mars network: DNS cumulative query time grows linearly while BGP stays flat after convergence.

[4] D. J. Israel, K. D. Mauldin, C. J. Roberts, J. W. Mitchell, A. A. Pulkkinen, L. V. A. Cooper, M. A. Johnson, S. D. Christe, and C. J. Gramling, "LunaNet: A Flexible and Extensible Lunar Exploration Communications and Navigation Infrastructure," in *2020 IEEE Aerospace Conference*. IEEE, 2020, pp. 1–14.

[5] A. Alhilal, T. Braud, and P. Hui, "The sky is not the limit anymore: Future architecture of the interplanetary internet," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 8, pp. 22–32, 2019.

[6] M. Feldmann, T. Tchilinguirian, and F. Walter, "A Border Gateway Protocol Extension for Distributing Endpoint Identifier Reachability Information in Delay-tolerant Networks," in *IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, 2025, in Press.

[7] E. Kline, "The ipn.arpa Zone and IPN DNS Operations," Internet-Draft draft-ek-dtn-ipn-arpa-00, November 2025, work in Progress.

[8] D. Project, "bgp-dns-omnetpp: Omnet++ models for dns vs. bgp eid reachability comparison," 2026, accessed: 2026-02-15. [Online]. Available: https://github.com/darksol-cloud/bgp-dns-omnetpp

[9] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[10] B. Sipos, M. Demmer, J. Ott, and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4," RFC 9174, January 2022.

[11] CCSDS, "CCSDS Bundle Protocol Specification," Consultative Committee for Space Data Systems, Washington, D.C., Tech. Rep. CCSDS 734.2-B-1, 2015.

[12] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay-and disruption-tolerant networking (dtn): An alternative solution for future satellite networking applications," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1980–1997, 2011.

[13] J. A. Fraire, O. De Jonckère, and S. C. Burleigh, "Routing in the Space Internet: A contact graph routing tutorial," *Journal of Network and Computer Applications*, vol. 173, p. 102884, 2021.

[14] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, January 2006.

[15] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 293–306, 2001.

[16] P. Mockapetris, "Domain Names - Implementation and Specification," RFC 1035, November 1987.

[17] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS Performance and the Effectiveness of Caching," *IEEE/ACM Transactions on Networking*, vol. 10, no. 5, pp. 589–603, 2002.

[18] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1. IEEE, 1999, pp. 126–134.

[19] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*. Springer, 2010, pp. 35–59.

[20] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, "Contact graph routing in dtn space networks: Overview, enhancements and performance," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, 2015.

[21] E. B. III and K. McKeever, "Bundle Protocol Security (BPSec)," RFC 9172, January 2022.