Resilient Communications for an Unreliable World

A Technical White Paper on Hybrid Cloud-Mesh Encrypted Messaging

MeshLink Protocol Specification v1.0

Authors: MeshLink Foundation
Date: January 2026
Status: Public Draft

Abstract

This paper presents MeshLink, an open protocol and reference implementation for resilient encrypted messaging that operates across both cloud infrastructure and local Bluetooth Low Energy (BLE) mesh networks. As governments increasingly weaponize internet shutdowns and centralized infrastructure proves vulnerable to both deliberate disruption and natural disasters, the need for communication systems that can function independently of traditional networks has become urgent. MeshLink addresses this through a hybrid transport architecture that maintains end-to-end encryption while enabling automatic failover between cloud and mesh transports based on network conditions. We describe the protocol design, cryptographic foundations, and a novel "bridge relay" mechanism inspired by Apple's Find My network that allows users with connectivity to relay encrypted messages for those without. Our analysis of deployment costs suggests a sustainable donation-funded model at scale.

Table of Contents

---

1. Introduction

The internet was designed to route around damage. Forty years later, that principle has been largely abandoned in favor of centralized architectures that create single points of failure, both technical and political. When a government orders telecommunications providers to disable service, or when a hurricane destroys cell towers, or when a stadium full of concert-goers overwhelms local network capacity, millions of people lose the ability to communicate at precisely the moments when communication matters most.

This paper introduces MeshLink, a messaging protocol designed around a simple premise: messages should find a way. MeshLink combines the convenience and global reach of cloud-based messaging with the resilience of local mesh networking, switching transparently between them based on conditions. When internet connectivity is strong, messages travel through encrypted cloud infrastructure. When connectivity degrades or fails, messages automatically route through nearby devices using Bluetooth Low Energy, hopping from phone to phone until they reach their destination or find a path back to the internet.

The contributions of this paper are:

1. A hybrid transport architecture that unifies cloud and mesh messaging under a single cryptographic identity
2. A "Rally Mode" protocol for location-bounded public channels during emergencies or large gatherings
3. A "Bridge Relay" mechanism that allows users with connectivity to relay encrypted messages for isolated users, inspired by Apple's Find My network but applied to general messaging
4. An analysis of deployment costs demonstrating viability of a donation-funded sustainability model

MeshLink builds upon and acknowledges the foundational work of BitChat [1], an open-source BLE mesh chat protocol, extending its design with cloud integration, bridge relay capabilities, and a focus on seamless user experience.

---

2. Background and Motivation

2.1 The Growing Crisis of Communication Blackouts

Internet shutdowns are no longer exceptional events. According to Access Now and the #KeepItOn coalition,

2024 marked a record-breaking year with 296 documented shutdowns across 54 countries [2]. This represented a 35% increase in affected countries compared to 2022, with seven nations implementing shutdowns for the first time. The economic cost of these disruptions reached $7.69 billion globally in 2024, rising to $19.7 billion in 2025 [3].

The human cost is more difficult to quantify but far more significant. During Iran's January 2026 protests, authorities imposed a nationwide internet blackout that Amnesty International described as "a serious human rights violation in itself" [4]. With reports of hundreds or thousands killed, the blackout served its intended purpose: preventing documentation of abuses and coordination among protesters. As one researcher noted, the internet is viewed as "an enemy" by authoritarian governments seeking to "control and suppress it" [5].

These shutdowns follow predictable patterns:

Trigger	2024 Instances	Notable Examples
Conflict	103	Myanmar (74), Gaza (6), Sudan
Protests	74	Iran, Kenya, Mozambique
Elections	12	Azerbaijan, Uganda, Venezuela
Exams	16	Iraq, Algeria, Jordan

The concentration is striking: Myanmar, India, Pakistan, and Russia accounted for nearly 70% of all 2024 shutdowns [6]. But the practice is spreading. Seven countries joined the "first-time offenders" list in 2024, including France and Malaysia.

2.2 Infrastructure Vulnerability During Disasters

Natural disasters reveal a parallel vulnerability in centralized communications infrastructure. Hurricane Maria in 2017 knocked out 88% of Puerto Rico's cell sites [7]. Hurricane Harvey disabled over 360 cell towers and left 200,000 homes without internet or telephone service [8]. The 2023 Maui wildfires destroyed telecommunications infrastructure at the moment residents most needed to coordinate evacuation and contact loved ones [9].

The pattern repeats with each major storm. When Hurricane Ian struck Florida in 2022, fiber lines providing service to Sanibel Island were severed, cutting the entire island off from communication with the mainland [10]. First responders deployed drones simply to assess which towers remained functional, a workaround that highlights the fragility of systems we depend upon.

Mesh networking has already proven valuable in these scenarios. During Hurricane Helene's aftermath in Eastern Tennessee, volunteers deployed approximately 120 Meshtastic radios for search and rescue coordination [11]. The technology worked precisely because it required no infrastructure, with messages hopping between devices up to 100 miles apart using LoRa radio frequencies.

2.3 The Congestion Problem at Scale

Infrastructure failure is not limited to authoritarian crackdowns and natural disasters. Large gatherings routinely overwhelm cellular networks through sheer volume. At music festivals, sporting events, and protests, tens of thousands of people competing for the same cell towers create conditions where devices show "full bars" but cannot send or receive data [12].

This congestion has real consequences. Payment systems fail at festival vendor booths. Safety teams cannot coordinate responses. Attendees cannot contact family members or access event information. One analysis noted that "festival internet issues" have become an expected part of large events, with organizers routinely advising attendees to download maps and information before arrival rather than assuming connectivity will be available [13].

The standard industry response involves temporary "cells on wheels" (COWs), distributed antenna systems (DAS), and dedicated event Wi-Fi. These solutions are expensive, require advance planning, and remain unavailable to spontaneous gatherings like protests. A mesh network that forms automatically among participants' existing smartphones would provide resilience without requiring specialized infrastructure.

2.4 Design Goals

Based on these observations, MeshLink targets the following design goals:

1. **Resilience**: Messages should be deliverable when internet is available, unavailable, or degraded
2. **Privacy**: End-to-end encryption must be maintained across all transport mechanisms
3. **Seamlessness**: Transport switching should be automatic and invisible to users
4. **Accessibility**: No specialized hardware should be required beyond a standard smartphone
5. **Sustainability**: The system should be operable through community funding without advertising or surveillance-based business models

---

3. Related Work

3.1 Existing Encrypted Messaging Protocols

Signal Protocol [14] established the standard for end-to-end encrypted messaging, combining the Double Ratchet Algorithm, prekeys, and a triple Diffie-Hellman handshake. Signal's protocol provides forward secrecy and break-in recovery, ensuring that compromise of long-term keys does not compromise past messages. However, Signal requires internet connectivity and centralized servers for message delivery.

Matrix [15] offers a federated approach to encrypted messaging. Users can run their own homeservers while maintaining interoperability with the broader network. Matrix's Megolm protocol enables efficient group encryption. MeshLink adopts Matrix as its cloud transport layer, benefiting from its existing encryption infrastructure while extending it with mesh capabilities.

3.2 Mesh Networking Protocols

BitChat [1] provides the most direct precedent for MeshLink's mesh layer. Developed by permissionless.tech and released into the public domain, BitChat implements BLE mesh messaging with Noise Protocol encryption, multi-hop routing (up to 7 hops), and IRC-style channels. MeshLink's mesh transport is directly derived from BitChat's protocol specification, extending it with cloud integration and bridge relay functionality.

Briar [16] takes a different approach, using Tor for internet transport and direct Wi-Fi/Bluetooth connections for local communication. Briar emphasizes journalist and activist security but requires manual connection establishment for local messaging rather than automatic mesh formation.

Meshtastic [17] operates on LoRa radio frequencies, enabling long-range communication (up to 10 km in open terrain) with dedicated hardware. Several proposals have explored integrating Meshtastic with smartphone-based mesh networks [18], and MeshLink's architecture could accommodate such integration as a future transport layer.

3.3 Crowd-Sourced Location Networks

Apple's Find My Network [19] demonstrates the viability of crowd-sourced relay networks at scale. Lost AirTags broadcast encrypted BLE beacons containing their public key. Any nearby iPhone (with user consent) can detect these beacons, encrypt its own location using the AirTag's public key, and upload the encrypted report to Apple's servers. The owner later retrieves and decrypts these reports. Crucially, the relaying device learns nothing about what it relayed or for whom.

MeshLink's Bridge Relay mechanism adapts this model for general messaging. Users with internet connectivity can relay encrypted message envelopes for users without, with the same privacy guarantees: the bridge cannot read message contents or identify senders.

---

4. System Architecture

4.1 Design Principles

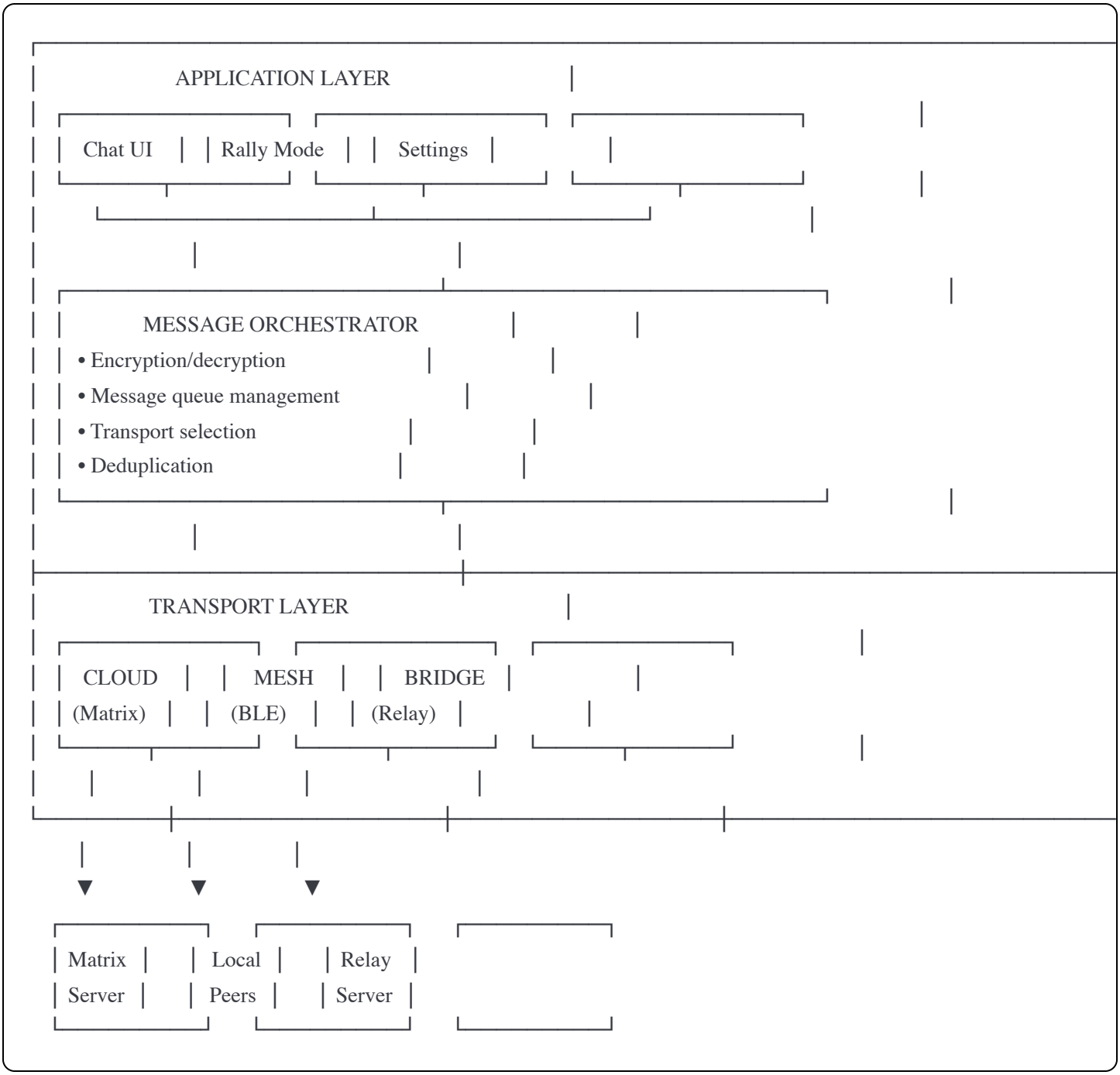
MeshLink's architecture follows three guiding principles:

Unified Identity: Users maintain a single cryptographic identity across all transports. Keys generated during account creation derive identifiers for cloud, mesh, and bridge relay communication. This eliminates the complexity of managing separate identities while ensuring that a contact verified on one transport is automatically verified on all others.

Transport Abstraction: Applications interact with a unified message queue. The transport manager handles routing decisions transparently, selecting cloud, mesh, or bridge relay based on current conditions. Applications receive delivery confirmations and status updates regardless of which transport delivered the message.

Encryption-First: Messages are encrypted before transport selection. The same ciphertext can travel via cloud, mesh, or bridge relay without re-encryption, ensuring consistent security properties across all paths.

4.2 System Components



Message Orchestrator: The central coordinator responsible for:

- Maintaining the outbound message queue
- Selecting appropriate transport for each message
- Handling delivery confirmations and retries
- Deduplicating messages received via multiple transports
- Managing encryption keys and sessions

Cloud Transport (Matrix): For messages to recipients not currently in mesh range, or when internet connectivity is strong. Uses Matrix's Megolm protocol for group encryption and Olm for 1:1 sessions.

Mesh Transport (BLE): For messages to recipients within multi-hop Bluetooth range, or when internet is unavailable. Based on BitChat's protocol with Noise Protocol encryption.

Bridge Transport: For messages when the sender lacks internet but nearby users can relay. Encrypted envelopes are passed to bridge nodes for upload to relay servers.

4.3 Transport Selection

The transport manager evaluates several signals to select the optimal delivery path:

```
TRANSPORT SELECTION ALGORITHM:

1. If recipient is a direct BLE peer (1 hop):
   → Use MESH (lowest latency for local peers)

2. If internet quality is GOOD and recipient is not nearby:
   → Use CLOUD

3. If internet quality is DEGRADED and mesh peers exist:
   → Use MESH with CLOUD fallback
   → Queue for cloud delivery if mesh delivery unconfirmed

4. If internet quality is NONE and mesh peers exist:
   → Use MESH
   → If recipient not in mesh, queue for BRIDGE relay

5. If internet quality is NONE and no mesh peers:
   → Queue for later delivery
   → Attempt BRIDGE relay if bridge nodes detected
```

Network quality assessment considers:

- Reachability of Matrix homeserver (ping success/failure)
- Latency to homeserver (>500ms considered degraded)
- Packet loss indicators
- User preference overrides

4.4 Message Deduplication

Because messages may travel via multiple paths (cloud and mesh simultaneously during degraded conditions), deduplication is essential. Each message carries a deterministic identifier:

```
message_id = SHA256(
    sender_public_key ||
    recipient_public_key ||
    timestamp_ms ||
    content_hash
)[0:16]
```

Recipients maintain a Bloom filter of recently received message IDs. Messages with IDs already in the filter are dropped without processing. The Bloom filter is sized for approximately 10,000 messages with a false positive rate below 0.01%.

---

5. Protocol Specification

5.1 Packet Format

MeshLink packets follow a binary format optimized for BLE's constrained MTU while remaining usable across all transports:

MESHLINK PACKET FORMAT

Offset	Size	Field	Description
0	1	Version	Protocol version (0x01)
1	1	Type	Message type
2	1	TTL	Hop limit (1-7, mesh only)
3	1	Flags	Bitmask (see below)
4	8	Timestamp	Unix milliseconds (uint64 BE)
12	16	MessageID	Deterministic ID for dedup
28	8	RecipientID	Truncated public key hash
36	2	PayloadLen	Payload length (uint16 BE)
38	N	Payload	Encrypted content
38+N	64	Signature	Ed25519 (optional)
...	P	Padding	PKCS#7 to block boundary

FLAGS:

0x01	hasRecipient	Unicast (set) vs broadcast (clear)
0x02	hasSignature	Signature present
0x04	isCompressed	LZ4 compression applied
0x08	isFragmented	Part of multi-packet message
0x10	requiresAck	Delivery confirmation requested

MESSAGE TYPES:

0x01	TEXT	Standard message
0x02	MEDIA_HEADER	Media attachment metadata
0x03	MEDIA_CHUNK	Media attachment fragment
0x04	ACK	Delivery acknowledgment
0x05	NOISE_INIT	Noise handshake initiation
0x06	NOISE_RESP	Noise handshake response
0x07	PEER_ANNOUNCE	Mesh peer advertisement
0x08	RELAY_REQUEST	Bridge relay envelope
0x09	RALLY_BROADCAST	Rally Mode public message

PADDING:

Packets are padded to fixed sizes to resist traffic analysis:

< 192 bytes → 256 bytes

< 448 bytes → 512 bytes

< 960 bytes → 1024 bytes

< 1984 bytes → 2048 bytes

5.2 Mesh Routing

MeshLink employs a flooding (gossip) protocol for mesh routing, consistent with BitChat's approach. When a node receives a packet not destined for itself:

1. Check Bloom filter for message ID
2. If ID likely seen before, drop packet
3. Add ID to Bloom filter
4. Decrement TTL
5. If TTL > 0, rebroadcast to all connected peers

This approach prioritizes reliability over efficiency, appropriate for the low-bandwidth, high-latency characteristics of BLE mesh networks. The TTL limit (default 7) prevents infinite propagation while allowing messages to traverse reasonable mesh diameters.

Private Message Routing: For unicast messages, relay nodes forward the complete encrypted packet without decryption. Only the recipient possessing the correct Noise session keys can access the payload. This maintains end-to-end encryption even as messages traverse multiple hops.

Broadcast Message Routing: Packets with the broadcast recipient ID (0xFFFFFFFFFFFFFFFF) are decrypted and processed by all receiving nodes, then rebroadcast according to the flooding algorithm. Rally Mode uses this mechanism for public channel communication.

5.3 BLE Service Definition

MeshLink operates as a BLE peripheral advertising the following service:

```
SERVICE UUID: 0x1234-MESH-LINK-0001 (example, actual UUID TBD)

CHARACTERISTICS:
  TX (Write):   UUID 0x0002, Write Without Response
                Used by central to send packets to peripheral

  RX (Notify):  UUID 0x0003, Notify
                Used by peripheral to send packets to central

  MTU (Read):   UUID 0x0004, Read
                Reports negotiated MTU for chunking
```

Devices operate in both central and peripheral roles simultaneously, enabling bidirectional mesh connectivity. Connection management prioritizes maintaining diverse peer connections to maximize network resilience.

---

6. Cryptographic Design

6.1 Identity and Key Hierarchy

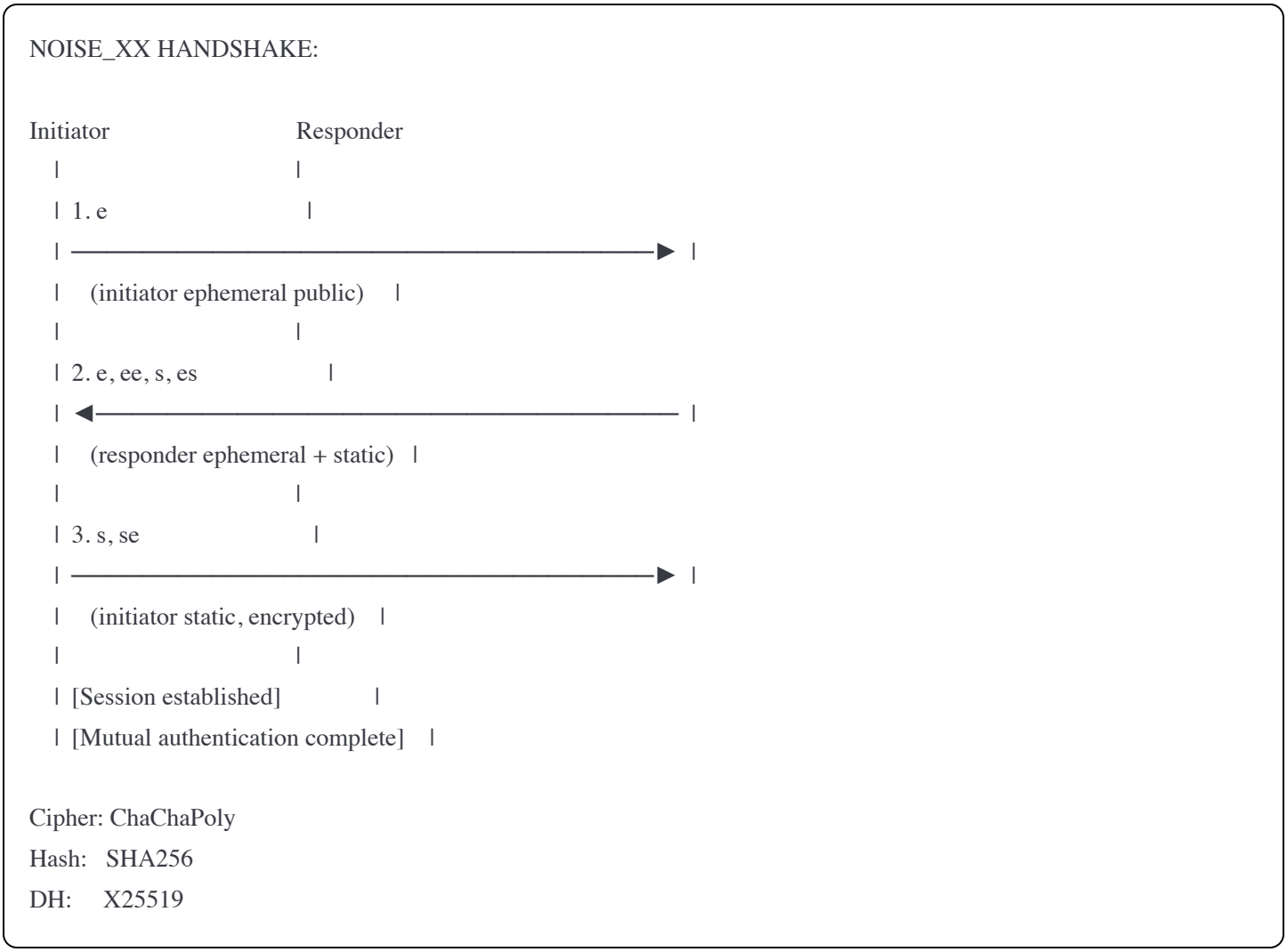
Each MeshLink identity derives from a single 32-byte seed generated using the platform's secure random number generator:

```
SEED: 32 bytes from platform CSPRNG

SIGNING KEY (Ed25519):
  private = Ed25519_PrivateKey(SEED)
  public  = Ed25519_PublicKey(private)

EXCHANGE KEY (X25519):
  private = X25519_PrivateKey(SEED)
  public  = X25519_PublicKey(private)

DERIVED IDENTIFIERS:
  mesh_peer_id    = SHA256(Ed25519_public)[0:8]
  matrix_user_id  = @base58(Ed25519_public[0:10]):server
  relay_key_hash  = SHA256(X25519_public)
```

The Ed25519 key provides signing capability for message authentication and identity verification. The X25519 key enables Diffie-Hellman key exchange for session establishment. Deterministic derivation ensures that all identifiers correspond to the same underlying identity.

6.2 Noise Protocol Sessions

MeshLink uses the Noise Protocol Framework [20] with the XX pattern for session establishment:



The XX pattern provides mutual authentication: both parties learn each other's static public keys and prove possession of the corresponding private keys. Forward secrecy is achieved through ephemeral key exchange.

6.3 Message Encryption

Once a Noise session is established, messages are encrypted using the derived transport keys:

```
ENCRYPTION:
  nonce      = counter (incremented per message)
  ciphertext = ChaCha20-Poly1305(transport_key, nonce, plaintext)

MESSAGE STRUCTURE:
  [payload_type: 1 byte]
  [content: variable]
  [padding: PKCS#7 to block boundary]
```

The payload type byte indicates content format (text, media reference, reaction, etc.). PKCS#7 padding ensures ciphertext lengths do not reveal plaintext characteristics.

6.4 Group Encryption

For group chats via cloud transport, MeshLink uses Matrix's Megolm protocol:

1. Group initiator generates Megolm session with random initial ratchet state
2. Session key is distributed to members via Olm (pairwise encrypted)
3. Messages are encrypted using current ratchet state
4. Ratchet advances after each message (forward secrecy)
5. New members receive session key with current ratchet (cannot decrypt history)

For mesh-only groups, MeshLink uses symmetric group keys:

```
GROUP KEY DERIVATION:
  group_seed    = random(32)
  group_key     = HKDF(
                  ikm  = group_seed,
                  salt = "meshlink-group-v1",
                  info = group_id,
                  len  = 32
                )
```

Group keys are rotated when membership changes. Members leaving trigger immediate rotation; members joining receive only the new key.

---

7. Emergency Broadcast System

7.1 Rally Mode Design

Rally Mode creates temporary public channels bounded by geographic location and time. It addresses the scenario where many people in physical proximity need to communicate but lack infrastructure access, such as during protests, festivals, or disaster response.

Channel Discovery: Channels are identified deterministically from location and time:

```
RALLY CHANNEL DERIVATION:

geohash      = encode(latitude, longitude, precision=6)  // ~1.2km cell
time_bucket  = floor(unix_time / (4 * 3600))             // 4-hour windows
channel_id   = SHA256(geohash || ":" || time_bucket)[0:16]
channel_key  = HKDF(
                 ikm  = channel_id,
                 salt = "meshlink-rally-v1",
                 info = geohash || time_bucket,
                 len  = 32
               )
```

Any user in the same approximate location during the same time window will derive identical channel identifiers and keys. No coordination or server interaction is required to join.

Ephemeral Identity: Rally Mode participants use session-specific identities:

```
SESSION KEY GENERATION:
  session_keypair = X25519.generate()  // Fresh each join

ANONYMOUS NAME DERIVATION:
  hash = SHA256(session_public_key)
  adjective = WORDLIST_ADJ[hash[0] % len(WORDLIST_ADJ)]
  noun      = WORDLIST_NOUN[hash[1] % len(WORDLIST_NOUN)]
  number    = hash[2] % 100
  name      = "{adjective}-{noun}-{number}"
  // Example: "brave-river-42"
```

Session keys are discarded when leaving Rally Mode. No persistent identity links users across sessions.

7.2 Safety and Moderation

Rally Mode's public nature requires safeguards:

Age Verification: Self-declared age verification (16+) gates access. While not cryptographically enforceable, it establishes consent and provides legal protection.

Local Reputation: Each device maintains reputation scores for observed peers:

```
REPUTATION CALCULATION:
  messages_seen    = count of messages from peer
  reports_against  = count of reports submitted against peer
  blocks_by_others = count of distinct peers who blocked this peer

  report_rate = reports_against / max(1, messages_seen)
  trust_score = max(0, 1.0 - report_rate * 10 - blocks_by_others * 0.1)

  HIDDEN if trust_score < 0.3
```

Reputation is local and subjective. Different users may see different views based on their own observations and those of their immediate peers.

Content Filtering: Optional on-device ML classification can flag potentially harmful content. Users choose whether to enable filtering and at what sensitivity level.

Reporting: Users can report messages in categories (spam, harassment, threats, child safety). For child safety reports, content hashes are queued for upload to appropriate authorities when connectivity permits.

---

8. Bridge Relay Network

8.1 Concept

The Bridge Relay network extends MeshLink's reach by allowing users with internet connectivity to relay encrypted messages for users without. This mechanism draws inspiration from Apple's Find My network but applies the concept to general messaging.

Consider a protest where cellular service is jammed or overwhelmed. Participants in the interior cannot reach the internet. But participants at the edges, or in nearby buildings, may have connectivity through different providers or Wi-Fi. With Bridge Relay, interior participants' messages can hop through the mesh to edge participants, who relay them to cloud infrastructure.

8.2 Protocol

Bridge Relay uses an envelope format that reveals no information about message content or sender:

```
RELAY ENVELOPE:
{
  "recipient_key_hash": base64(SHA256(recipient_X25519_public)),
  "encrypted_payload":  base64(noise_encrypted_packet),
  "ttl_hours":          4,
  "priority":           "normal" | "urgent" | "emergency",
  "nonce":              base64(random(16)),
  "created_at":         unix_milliseconds
}
```

Privacy Guarantees:

- Bridge node cannot read payload (Noise encrypted to recipient)
- Bridge node cannot identify sender (not present in envelope)
- Relay server sees only recipient key hash (not full public key)
- TTL prevents indefinite storage or traffic analysis over time

8.3 Bridge Node Behavior

Users opt into serving as bridge nodes. When enabled, their device:

1. Monitors for mesh packets with RELAY_REQUEST type
2. Evaluates relay eligibility (bandwidth budget, battery level)
3. If eligible, uploads envelope to relay server
4. Tracks bandwidth usage against daily limit

```
BRIDGE NODE CONFIGURATION:
  enabled          = false (requires explicit opt-in)
  bandwidth_budget  = 10 MB/day (user configurable)
  battery_threshold = 30% (pause below this level)
  relay_for        = "all" | "contacts_only"
```

Consent Flow: First-time bridge potential triggers an explanatory dialog:

> You have internet access, but 47 nearby people don't. You can relay their encrypted messages to the outside world.
>
> What you should know:
>
> - Messages are end-to-end encrypted
> - You cannot read the content
> - Uses approximately 2-5 MB/hour

> - Pauses below 30% battery
>
> [Enable] [Not Now]

8.4 Relay Server

The relay server is intentionally minimal:

```
ENDPOINTS:

POST /relay/upload
  Body: RelayEnvelope
  Auth: None (rate limited by IP)
  Action: Store envelope keyed by recipient_key_hash with TTL

GET /relay/poll?key_hash={hash}
  Auth: None
  Action: Return and delete all envelopes for key_hash

STORAGE:
  Redis with TTL-based expiration
  No logging of envelope contents
  Rate limiting: 60 uploads/minute per IP
```

Recipients poll using their key hash (which they can derive from their own public key). Retrieved envelopes are decrypted using the recipient's Noise session keys.

The relay server can be self-hosted or federated. MeshLink provides a reference deployment, but organizations can operate their own relay infrastructure for their communities.

---

9. Implementation Considerations

9.1 Platform Constraints

iOS Background Execution: iOS restricts BLE operations in the background. Advertising intervals slow to approximately once per second, and connections may be terminated after 30 seconds of inactivity. MeshLink mitigates this through:

- State restoration for BLE central/peripheral roles
- Opportunistic transmission when app briefly activates
- Push notification triggers from cloud to prompt foreground activity

Android Battery Optimization: Aggressive battery optimization on Android (particularly Samsung, Xiaomi, and Huawei devices) can terminate background services. MeshLink requires:

- Foreground service with persistent notification
- User guidance to disable battery optimization for the app
- WorkManager for deferred cloud sync tasks

BLE MTU Negotiation: Default BLE MTU is often 20-23 bytes. MeshLink negotiates larger MTU (up to 512 bytes) where supported:

- iOS negotiates automatically
- Android requires explicit requestMtu() call
- Packet chunking handles cases where negotiation fails

9.2 Performance Characteristics

Based on BitChat's measurements and our testing:

Metric	Value	Notes
Mesh message latency (1 hop)	50-200ms	Depends on connection state
Mesh message latency (7 hops)	1-5s	Worst case with congestion
BLE range (typical)	10-50m	Varies by device and environment
BLE range (optimal)	Up to 100m	Line-of-sight, modern devices
Mesh peer capacity	7-10 concurrent	Platform dependent
Message throughput	~10 msg/s	Per peer connection

9.3 Battery Impact

Continuous mesh operation consumes battery. Our measurements on iPhone 14:

Mode	Hourly Battery Impact
Cloud only (idle)	1-2%
Mesh active (8 peers)	3-5%
Bridge relay (active)	4-6%
Rally Mode (active)	5-8%

MeshLink implements adaptive power management:

- Reduce scan frequency when few peers detected
- Pause mesh when battery below configurable threshold
- Shift to cloud-only when on low power mode

---

10. Security Analysis

10.1 Threat Model

MeshLink protects against:

Threat	Mitigation
Mass surveillance	E2E encryption (Noise Protocol, Megolm)
Server compromise	Servers never see plaintext
Network eavesdropping	TLS for cloud, Noise for mesh
Traffic analysis	Fixed packet sizes, padding
Metadata leakage	Mesh reduces server visibility
Single point of failure	Hybrid transport with automatic failover
Targeted blocking	Mesh bypasses network-level blocks

MeshLink partially protects against:

Threat	Limitation	Mitigation
Device compromise	Keys stored on device	Secure Enclave/Keystore where available
Social engineering	User education required	Verification badges, safety prompts
Mesh peer identification	BLE advertising visible	Rotating identifiers, optional stealth mode

MeshLink does not protect against:

- Physical device access with passcode
- Compromised recipient sharing messages
- Legal compulsion of end users
- Zero-day vulnerabilities in cryptographic libraries

10.2 Cryptographic Strength

MeshLink's cryptographic choices follow current best practices:

Function	Algorithm	Security Level
Key exchange	X25519	~128-bit
Signing	Ed25519	~128-bit
Symmetric encryption	ChaCha20-Poly1305	256-bit key
Hashing	SHA-256, BLAKE2b	256-bit
KDF	HKDF-SHA256	Depends on input entropy

The Noise Protocol framework has undergone extensive academic analysis [21] and is used in production by WireGuard, WhatsApp, and Lightning Network.

10.3 Bridge Relay Privacy

Bridge Relay maintains strong privacy properties:

1. **Content Privacy**: Payloads are Noise-encrypted to the recipient. Bridge nodes and relay servers cannot decrypt.
2. **Sender Anonymity**: The relay envelope contains no sender identification. The bridge node sees only the mesh peer ID of the immediate sender (which may be a relay, not the originator).

3. **Recipient Pseudonymity**: Relay servers see only SHA256 hashes of recipient public keys, not the keys themselves or any human-readable identifiers.
4. **Temporal Privacy**: TTL-based expiration prevents long-term traffic analysis. Default retention is 4 hours.

Remaining risks:

- Traffic correlation by relay server operators observing upload/poll timing
- Compromised bridge nodes logging mesh peer IDs (mitigated by rotating IDs)
- Global passive adversary correlating cloud and mesh traffic patterns

---

11. Cost and Sustainability

11.1 Infrastructure Costs

MeshLink's server infrastructure consists of:

1. Matrix homeserver (message sync, push notifications)
2. Relay server (bridge message storage)
3. Supporting services (database, monitoring)

Cost projections per 1,000 Monthly Active Users (MAU):

Scale	Monthly Cost	Per User	Configuration
1,000 MAU	$39	$0.039	Single VPS, all services
10,000 MAU	$165	$0.017	Separated services, basic redundancy
100,000 MAU	$745	$0.007	Full HA, Kubernetes
1,000,000 MAU	$4,500	$0.0045	Multi-region, auto-scaling

These projections assume:

- 50 messages/user/day average
- 10% messages include media (~200KB average)
- 5% messages route through bridge relay
- 30-day message retention

11.2 Comparison to Existing Services

For context, Signal reportedly operates on approximately $50 million annually for ~40 million MAU, roughly $1.25/user/year [22]. MeshLink's lower cost projection ($0.05-0.10/user/year at scale) results from:

- Mesh traffic offloaded to user devices (no server cost)
- No voice/video infrastructure initially
- Simpler server architecture (Matrix handles complexity)
- Bridge relay is user-powered

11.3 Sustainability Model

MeshLink adopts Signal's donation-funded model. Analysis of donation scenarios:

Assumption	**10K MAU**	**100K MAU**	**1M MAU**
Conservative (1% donate, $5/year avg)			
Revenue	$500	$5,000	$50,000
Costs	$2,000	$8,940	$54,000
Surplus	-$1,500	-$3,940	-$4,000
Moderate (2% donate, $10/year avg)			
Revenue	$2,000	$20,000	$200,000
Costs	$2,000	$8,940	$54,000
Surplus	$0	+$11,060	+$146,000
Optimistic (3% donate, $15/year avg)			
Revenue	$4,500	$45,000	$450,000
Costs	$2,000	$8,940	$54,000
Surplus	+$2,500	+$36,060	+$396,000

Break-even requires approximately 10,000 MAU with moderate donation rates. Initial funding (grants, foundation support) is needed for the first 6-12 months of operation.

Donation tiers with cosmetic badges (not affecting functionality) provide recognition:

Tier	Amount	Badge
Supporter	Any amount	Green heart
Contributor	$5+/month	Star
Champion	$20+/month	Trophy
Lifetime	$500 once	Permanent sparkle

---

12. Conclusion

The internet's promise of global, resilient communication has been undermined by centralization, making it vulnerable to both deliberate shutdowns and infrastructure failures. In 2024 alone, governments imposed 296 internet shutdowns affecting hundreds of millions of people. Natural disasters routinely destroy the cell towers and fiber lines that modern communication depends upon. Large gatherings overwhelm cellular networks, leaving participants unable to contact each other or access information.

MeshLink addresses these challenges through a hybrid architecture that combines cloud-based messaging with local mesh networking. When internet connectivity is available, messages travel through encrypted cloud infrastructure with the convenience users expect. When connectivity fails, messages automatically route through nearby devices using Bluetooth Low Energy. The Bridge Relay mechanism extends this resilience further, allowing users with connectivity to relay encrypted messages for those without, inspired by Apple's Find My network but applied to general messaging.

The protocol maintains end-to-end encryption across all transport mechanisms, ensuring that the security guarantees users rely upon are preserved regardless of how messages are delivered. Rally Mode provides emergency broadcast capability for location-bounded public channels during large gatherings or crises. All of this operates on standard smartphones without specialized hardware.

Our cost analysis demonstrates that a donation-funded sustainability model is viable at scale, following the precedent established by Signal. Infrastructure costs of approximately $0.01 per user per month enable community-funded operation without advertising or surveillance-based business models.

MeshLink builds upon the foundation established by BitChat and the broader mesh networking community. We release this specification and reference implementation under open licenses, inviting collaboration to build communication infrastructure that serves human needs rather than authoritarian control.

Messages should find a way.

---

13. References

[1] permissionless.tech. "BitChat: Bluetooth Mesh Chat Protocol." GitHub, 2025. https://github.com/permissionlesstech/bitchat

[2] Access Now. "Emboldened offenders, endangered communities: Internet shutdowns in 2024." February 2025. https://www.accessnow.org/internet-shutdowns-2024/

[3] Top10VPN. "The Cost of Internet Shutdowns." January 2026. https://www.top10vpn.com/research/cost-of-internet-shutdowns/

[4] Amnesty International. "Iran internet shutdown hides violations in escalating protests." January 2026. https://www.amnesty.org/en/latest/news/2026/01/internet-shutdown-in-iran/

[5] Wikipedia. "2026 Internet blackout in Iran." January 2026. https://en.wikipedia.org/wiki/2026_Internet_blackout_in_Iran

[6] Context News. "2024 was the worst year for internet shutdowns around the world." March 2025. https://www.context.news/digital-rights/2024-was-the-worst-year-for-internet-shutdowns

[7] Scientific American. "Cell Phone Service Must Be Restored Quicker after Hurricanes." February 2024. https://www.scientificamerican.com/article/cell-phone-service-must-be-restored-quicker-after-hurricanes/

[8] GovTech. "Will Your Cell Service Work if a Hurricane Rolls Through?" April 2021. https://www.govtech.com/em/disaster/will-your-cell-service-work-if-a-hurricane-rolls-through

[9] Scientific American. "How Fires, Floods and Hurricanes Create Deadly Pockets of Information Isolation." February 2025. https://www.scientificamerican.com/article/how-fires-floods-and-hurricanes-create-deadly-pockets

[10] Inside Unmanned Systems. "Restoring Communication After a Disaster." December 2023. https://insideunmannedsystems.com/restoring-communication-after-a-disaster/

[11] Gulf Coast News Now. "Meshtastic: How to stay connected after a hurricane hits." August 2025. https://www.gulfcoastnewsnow.com/article/meshtastic-stay-connected-hurricane-florida/

[12] Ticket Fairy. "Beyond Wi-Fi: Satellite & Mesh Networking for Festival Connectivity." October 2025. https://www.ticketfairy.com/blog/beyond-wi-fi-satellite-mesh-networking

[13] YohoMobile. "Why Your Internet Crawls at Crowded Events." 2025. https://yohomobile.com/guide-fix-slow-internet-at-crowded-events

[14] Marlinspike, M. and Perrin, T. "The X3DH Key Agreement Protocol." Signal, 2016. https://signal.org/docs/specifications/x3dh/

[15] Matrix.org Foundation. "Matrix Specification." https://spec.matrix.org/

[16] Briar Project. "Briar: Secure Messaging, Anywhere." https://briarproject.org/

[17] Meshtastic. "An open source, off-grid, decentralized, mesh network." https://meshtastic.org/

[18] permissionless.tech. "BitChat over LoRa: a decentralized, offline messaging framework." GitHub Issue #180, July 2025. https://github.com/permissionlesstech/bitchat/issues/180

[19] Apple Inc. "Find My network accessory specification." 2021.

[20] Perrin, T. "The Noise Protocol Framework." https://noiseprotocol.org/

[21] Kobeissi, N., Nicolas, G., and Bhargavan, K. "Noise Explorer: Fully Automated Modeling and Verification for Arbitrary Noise Protocols." IEEE Euro S&P, 2019.

[22] Signal Foundation. "Signal's 2023 Annual Report." 2024.

[23] Council on Foreign Relations. "Iran's Protests and the Internet Blackout That Followed." January 2026. https://www.cfr.org/article/irans-protests-and-internet-blackout

[24] goTenna. "Mobile Mesh Networks Can Ensure Communication in Disaster." January 2021. https://gotenna.com/blogs/newsroom/mobile-mesh-networks-can-ensure-communication-in-disaster

[25] Schneier, B. "Deliberate Internet Shutdowns." Schneier on Security, December 2025. https://www.schneier.com/blog/archives/2025/12/deliberate-internet-shutdowns.html

---

14. Acknowledgments

MeshLink builds directly upon the work of the BitChat project and the permissionless.tech community. The BLE mesh protocol, packet format, and routing algorithms described in this paper are derived from BitChat's public domain specification. We are grateful for their decision to release this foundational work without restriction.

We acknowledge the researchers and advocates at Access Now, the #KeepItOn coalition, and digital rights organizations worldwide who document internet shutdowns and fight for communication freedom. Their work informs and motivates this project.

The Noise Protocol Framework, developed by Trevor Perrin, provides the cryptographic foundation for MeshLink's security. The Matrix.org Foundation's work on federated encrypted messaging enables our cloud transport layer.

Finally, we acknowledge the countless individuals who have risked their safety to document abuses during internet blackouts, often using whatever communication channels remained available. This project exists because their stories reached the outside world despite the obstacles placed in their path.

Appendix A: Glossary

Term	Definition
BLE	Bluetooth Low Energy, the wireless protocol used for mesh networking
Bridge	A user with internet connectivity who relays messages for users without
E2E	End-to-end encryption, where only sender and recipient can read messages
Geohash	A string encoding of geographic coordinates used for location-based channels
Megolm	Matrix's group encryption protocol
Mesh	Network topology where devices relay messages through each other
MTU	Maximum Transmission Unit, the largest packet size a connection supports
Noise Protocol	Cryptographic framework for secure channel establishment
Rally Mode	Public channel for all users in a geographic area
TTL	Time-to-live, controls how long messages or packets persist
X25519	Elliptic curve Diffie-Hellman function for key exchange

Appendix B: Document History

Version	Date	Changes
1.0-draft	January 2026	Initial public draft