Network Working Group
Internet-Draft
Intended status: Informational
Expires: 17 September 2026

B.A. Fisher
DPA R&D Ltd (https://www.dpa-cloud.co.uk)

16 March 2026

UZP: Universal Zero-Port Transport Protocol
draft-dpa-uzp-transport-01

Abstract

The Universal Zero-Port Transport Protocol (UZP) defines an identity-addressed, encrypted-by-default transport for the Universal Zero-Port Interconnect Framework (UZPIF; [UZPIF]). Instead of exposing IP:port listeners, both endpoints establish outbound, identity-bound sessions to one or more Rendezvous Nodes (RNs). The RN performs flow stitching but never terminates end-to-end cryptography or holds long-term secrets. Application data is carried over an authenticated encryption (AEAD) channel keyed by a handshake based on modern and post-quantum-capable primitives, and reliability is expressed at the block level rather than at the TCP segment or stream level.

This document is part of an experimental, research-oriented Independent Stream suite. It defines the current normative baseline for trust objects, validation rules, and security semantics within its scope. Hard interoperability is expected for shared object semantics and validation rules. Full wire-level, clustering, and proof-family interoperability is not claimed everywhere yet; the remaining details are intentionally profile-defined or deferred where noted.

Note to Reviewers

This document is part of an experimental, research-oriented suite prepared for the Independent Stream. It is published to enable structured technical review, interoperability discussion, and disciplined specification development, and it remains a work-in-progress research artefact rather than a finished specification.

Within that suite, this revision defines the current normative baseline for trust objects, validation rules, and security semantics within UZP. Hard interoperability is expected for shared object semantics and validation rules. Full wire-level, clustering, and proof-family interoperability is not claimed everywhere yet; the remaining details are intentionally profile-defined or deferred where noted.

Where this document provides numeric guidance (for example timers, windows, or congestion-related tuning), the intent is to offer recommended bounds suitable for experimentation; profile-based behaviour and implementation discretion are explicitly expected within stated limits.

UZP is designed for identity-first, zero-port environments where conventional port-based transport assumptions do not hold. It is intended for those deployment conditions, and outbound-only attachment does not by itself solve privacy, decentralisation, or RN availability.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 17 September 2026.

Copyright Notice

Table of Contents

1. Scope and Status

This Internet-Draft is part of an experimental, research-oriented suite prepared for the Independent Stream. It describes UZP for the Universal Zero-Port Interconnect Framework (UZPIF; [UZPIF]), while remaining open to substantial revision through review and implementation experience.

Within that suite, this document defines the current normative baseline for trust objects, validation rules, and security semantics within UZP, especially for handshakes, Grant processing, proof processing, and RN behaviour. Hard interoperability is expected for shared object semantics and validation rules.

Full wire-level, clustering, and proof-family interoperability is not claimed everywhere yet. In particular, some byte-level wire encodings, clustering behaviour, proof families, and deployment profiles remain intentionally profile-defined or deferred.

Unless otherwise stated, design parameters and numeric values are provided as numeric guidance and recommended bounds, rather than as fixed constants. Implementations may adopt alternative congestion tuning, profile-based behaviour, and implementation-defined choices within the constraints explicitly described in the relevant sections.

It is designed for experimentation and profile-driven deployments within its target environment. Privacy, decentralisation, and RN availability remain deployment- and profile-dependent properties.

2. Introduction

Many deployed Internet services continue to rely on listening sockets bound to IP:port tuples. This exposes reachable services to scanning, unsolicited ingress, and a wide class of lateral-movement and amplification attacks. Contemporary defences such as WAFs, DDoS scrubbing, layered ACLs, and micro-segmentation can mitigate portions of that exposure, but they do not change the underlying listener model.

The Universal Zero-Port Interconnect Framework (UZPIF; [UZPIF]) defines an architecture in which services are reached via outbound, identity-bound connections to Rendezvous Nodes (RNs). UZP is the transport protocol that operates beneath UZPIF ([UZPIF]) and is designed for identity-first, zero-port environments where conventional port-listening assumptions do not hold. In that context, it provides transport and security semantics comparable to conventional TCP+TLS or QUIC+TLS data planes, while allowing legacy applications or attached equipment to be mediated through a Hardware Integration Layer (HIL).

The design builds on ideas from QUIC [RFC9000], HIP [RFC7401], and Zero Trust Architecture [NIST-SP800-207]. Rendezvous-based systems such as Tor [TOR2004] show that endpoint identity can be decoupled from network location. UZP is intended to be compatible with post-quantum cryptography profiles [NIST-PQC].

This draft should therefore be read as part of an experimental, research-oriented Independent Stream suite and as the current normative baseline for trust objects, validation rules, and security semantics within UZP. Hard interoperability is expected for shared object semantics and validation rules. Full wire-level, clustering, and proof-family interoperability is not claimed everywhere yet; the

remaining details are intentionally profile-defined or deferred. Outbound-only attachment reduces direct inbound exposure, but it does not by itself solve privacy, decentralisation, or RN availability.

2.1. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses terminology from UZPIF ([UZPIF]) and related work:

* Endpoint (EP): A host participating in UZP communication. EPs never listen on public IP:port tuples.

* Rendezvous Node (RN): A relay and coordination node that accepts outbound connections from EPs, validates Pantheon credentials and Grant artefacts under policy, and stitches identity-bound flows. An RN is not by itself an issuer of identity, Grant, or revocation truth.

* Pantheon: A federated or deployment-scoped identity, attestation, and policy plane used by UZPIF ([UZPIF]) and UZP that binds identity, policy, and trust metadata to keys or selectors accepted under local policy, may validate or certify those bindings, and may issue credentials, Grants, and delegations over them.

* Canonical Identity (CID): A long-lived, cryptographic identifier derived from a principal’s public signing key.

* Ephemeral Identity (EID): A per-session identity bound to short-lived key material.

* Block: The semantic unit of reliability (acknowledgement / retransmission) in UZP.

* Frame: The semantic unit of application payload mapping inside a block; this draft does not yet assign a final wire encoding to frame types.

* Proof-of-Reachability (PoR): An RN-relayed authenticated liveness proof in which a peer returns profile-defined proof material bound to identity, nonce, RN context, expiry, and session context; see Section 12.

3. Design Goals

UZP is intended to satisfy the following primary goals:

* Zero exposed ports: EPs MUST operate with no listening sockets reachable from the public Internet. All communication originates from outbound connections to RNs.

* Identity-first addressing: The fundamental addressing object is a cryptographic identity (CID/EID), not an IP:port pair, following the spirit of HIP [RFC7401].

* Encrypted-by-default: All application data MUST be carried over an AEAD-protected channel, with forward secrecy and exporter support comparable to TLS 1.3 [RFC8446].

* Modern and PQ-capable: The handshake MUST support both classical and post-quantum key exchange and authentication, with algorithm agility and central policy control [NIST-PQC].

* Block-level reliability: Reliability is expressed at the block level, enabling selective retransmission and deterministic pacing similar in spirit to, but distinct from, QUIC’s stream framing [RFC9000].

* RN minimal trust: RNs MUST NOT learn application plaintext or hold long-term identity secrets; they may only drop, reorder, or delay encrypted traffic.

Where this document provides numeric guidance (for example, congestion-related tuning, replay windows, or pacing), it is intended as recommended bounds for experimentation. Implementations may apply profile-based behaviour and implementation-defined tuning within any explicit limits stated in the relevant sections.

4. Architectural Overview

At a high level, UZP operates as follows:

1. Each EP opens one or more outbound control channels to one or more RNs.

2. The EP authenticates to Pantheon and obtains Grants that authorise communication with a peer identity under specific policy.

3. To talk to another EP, the initiator submits a Join request to an RN, containing its own CID/EID, the target CID, and the relevant Grant material.

4. The responder independently connects to an RN (which MAY be the same RN or a different RN in the same trust domain) and presents its own Grants.

5. The RN stitches the two UZP sessions into a zero-port interconnect tunnel (ZPIT) once both sides have been validated.

The RN never terminates end-to-end cryptography; each side performs a UZP handshake with the RN, derives end-to-end keys using exporter material, and then switches to E2E AEAD for application data.

4.1. High-Level Session Flow

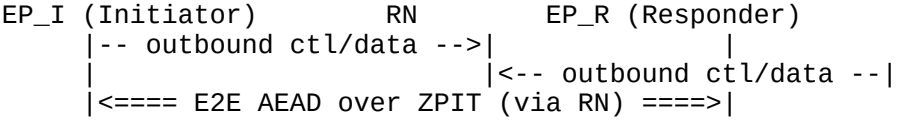

Figure 1: High-level communication pattern: both endpoints initiate outbound- only connections to the RN, which stitches an end-to-end ZPIT.

This figure shows both endpoints initiating outbound sessions to the RN, which stitches them into a ZPIT.

4.2. Session State Model

For interoperable transport behaviour, endpoints MUST model each UZP association using the following baseline states:

new No RN attachment, authenticated peer state, or accepted continuity state has yet been established.

pending RN attachment, Join processing, or handshake evaluation is in progress. Application data other than explicitly permitted early data under Section 9 MUST NOT be released in this state.

authenticated Peer identity, Grant state, and handshake bindings have been accepted, but the association is not yet in normal stitched data transfer.

stitched The RN has completed the authorised stitch and the endpoints may exchange normal AEAD-protected application traffic within the accepted Grant scope.

failover / rebind Continuity is being re-established at the same RN or at an alternate eligible RN. Endpoints MAY use bounded continuity state only as allowed by the failover rules in this document; any authority expansion requires fresh validation, and no RN-local claim alone is sufficient to return the association to "stitched".

closed The association is terminated or abandoned. Further traffic under that session context MUST be rejected unless a separate resumption or re-enrolment mechanism explicitly authorises a new session.

Endpoints MUST begin in "new", MUST pass through "pending" before accepting authenticated state, MUST NOT treat "stitched" continuity as proof of continuing authorisation without the checks required by this document, and MUST treat transition into "failover / rebind" as an availability event rather than as automatic trust continuity. Loss of RN attachment changes rendezvous state; it does not by itself revoke externally verifiable identity, Grant, or revocation truth.

4.3. HIL-Brokered Sessions

Where legacy applications or non-native hardware cannot participate directly in UZP, a Hardware Integration Layer (HIL) as described by UZPIF ([UZPIF]) MAY establish or broker the UZP session on their behalf. In such cases, the HIL is the policy-bound compatibility boundary and the attached legacy system is not automatically a native UZP endpoint.

A HIL-brokered session MUST bind transport accountability to the authenticated HIL identity and MAY also carry an auditable device, slot, or port identifier for the attached legacy system when deployment policy requires that distinction. Grants and any translated session context MUST be scoped so that the HIL cannot silently generalise the authority of the attached system beyond the mediation policy under which it operates.

RN participation in a UZP session, whether native or HIL-brokered, MUST NOT be treated as RN authority over end-to-end identity or authentication truth. The RN may relay flights, validate locally relevant Grant constraints, and enforce forwarding policy, but the authenticated truth of the session remains bound to the cryptographic endpoint or designated HIL endpoint, the handshake transcript, and the applicable Grant state rather than to RN observation of traffic or path position.

4.4. Identity Model and CID Stability

Pantheon authorities bind a principal’s long-term public signing key to identity, policy, and trust metadata; the canonical identity CID is defined as a hash (e.g., BLAKE3) of that public signing key. CIDs are intended to be stable over multi-year time scales and across multiple devices in the same administrative entity, and SHOULD only change when the underlying key is rotated or revoked due to compromise.

Deployments MAY use locally generated or externally attested keys if Pantheon policy allows; Pantheon authorities validate or certify those bindings and are not required to generate or custody the corresponding private keys.

Each transport session also has an Ephemeral Identity (EID), derived from short-lived key material. EIDs are bound to CIDs via Pantheon credentials over that ephemeral key material and are used within the UZP handshake and record layer.

This separation mirrors the long-term vs. ephemeral key split in both HIP [RFC7401] and TLS 1.3 [RFC8446].

5. Handshake Overview

The UZP handshake provides:

* Mutual (or unilateral) authentication bound to CIDs and EIDs.
* Negotiation of cipher suites and AEAD algorithms.
* Derivation of exporter keys for UZPIF ([UZPIF]) and higher layers.
* Integration of Pantheon Grants, including replay protection and policy binding.

The RN forwards handshake messages but does not terminate the cryptographic handshake itself. Instead, both EPs derive end-to-end secrets using exporter material bound to the identities and Grants, and then switch to direct AEAD protection across the ZPIT.

5.1. Specification Boundary

This revision defines UZP message semantics, session-state logic, handshake roles, cryptographic bindings, replay constraints, PoR semantics, RN behaviour, and related transport security semantics needed for semantic interoperability.

Exact byte-level encoding is not yet fixed in this revision. Packet layouts, message serialisations, some transport profiling, registry completeness, retransmission and congestion profiles, PMTU or fragmentation handling, and related deployment-specific choices remain deferred or profile-defined.

Figures and message labels in this document are therefore illustrative. They describe message purpose and sequencing, not a completed interoperable wire format.

5.2. Flight Diagram

Figure Figure 2 sketches a representative two-party handshake mediated by an RN. The message labels are illustrative and show role and sequencing rather than a final byte-level wire format.

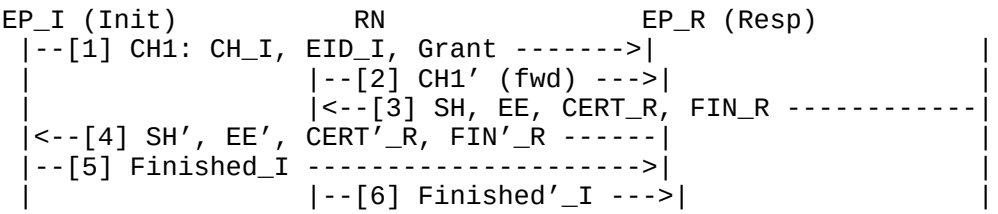

Figure 2: Example UZP handshake flights via an RN. Message indices [1]-[6] are explained in the legend. Labels are illustrative and profile- neutral.

This figure summarizes the RN-relayed handshake flights and where forwarding occurs.

Legend:

[1] CH1: ClientHello_I, EID_I, Grant

[2] CH1': Forwarded ClientHello_I

[3] SH, EE, CERT_R, FIN_R

[4] SH', EE', CERT'_R, FIN'_R

[5] Finished_I (initiator final confirm towards RN)

[6] Finished'_I (forwarded initiator confirm towards responder)

6. Cryptographic Negotiation and AEAD Tag Length

UZP requires negotiation over a cipher-suite identifier space. In this revision, each suite selection binds:

* A key exchange mechanism (e.g., X25519, or a PQ KEM candidate).
* A signature algorithm for authentication.
* An AEAD algorithm for record protection (e.g., AES-GCM-128, ChaCha20-Poly1305).
* A KDF (e.g., HKDF-SHA256 or a BLAKE3-based KDF).

6.1. AEAD Tag Length

The AEAD tag length is algorithm-dependent but subject to the following constraints:

* Implementations MUST use a tag length of at least 96 bits for all UZP application data records.
* When an AEAD algorithm allows variable tag lengths, endpoints SHOULD use the algorithm’s full tag length (typically 128 bits for AES-GCM) and MUST NOT negotiate a tag length below 96 bits.
* The tag length used for a session is fixed for the lifetime of that session and is implied by the negotiated cipher suite.

This requirement follows common practice in modern protocols, which treat 96 bits as a practical lower bound on AEAD authentication tags for wide-area deployments, while allowing future AEADs with different native tag lengths.

7. Blocks, Frames, and Reliability

UZP defines a block-oriented reliability model and associated transport semantics:

* A block is the unit of transmission and acknowledgement semantics.
* Each block contains one or more frames, which carry application data or control information.
* Blocks carry monotonically increasing block numbers, and acknowledgement signalling references blocks rather than individual bytes.

This block-level model requires selective retransmission and deterministic pacing semantics:

* EPs can detect loss patterns and retransmit only affected blocks.

* Congestion control and pacing, when used, operate over block units, similarly in spirit to how QUIC applies logic over packet and frame boundaries [RFC9000].

This draft does not yet standardise exact retransmission timers, acknowledgement encodings, PMTU discovery, fragmentation policy, congestion algorithm choice, or error-signalling registries. Later revisions or deployment profiles may fix those wire- and policy-level details while preserving the block-oriented semantics defined here.

Blocks are encrypted and authenticated with the negotiated AEAD. The associated data includes:

* CID and EID for both parties.

* Direction and block number.

* A context-specific label (e.g., "uzp-application" or "uzp-handshake").

8. Exporters

UZP defines an exporter interface analogous to TLS exporters [RFC8446]. Exporters derive context-specific keys from the main handshake secrets, using labels and context values that MUST be bound to identities and transport parameters.

An exporter input consists of:

* A label (e.g., "uzpif-zpit-key").

* A context value (which may include CIDs, EIDs, RN identifiers, and Grant nonces).

* An output length.

Exported keys are used by:

* UZPIF ([UZPIF]) to derive per-ZPIT keys for monitoring or additional encapsulation.

* Higher-layer protocols wishing to bind application security directly to UZP session properties.

Exporters MUST incorporate both CIDs and the negotiated transport parameters so that re-use of exporters across sessions is cryptographically separated.

9. 0-RTT Data and Replay Handling

UZP allows, but tightly constrains, 0-RTT (early) data:

* Early data MAY be sent by a client that possesses a valid, non-expired session resumption token and a Pantheon Grant that explicitly permits early data.

* Early data MUST be cryptographically distinct from 1-RTT data and MUST be bound to a monotonically increasing Grant nonce.

* RNs MUST treat 0-RTT flows as replayable at the transport level and MUST NOT rely on transport-layer properties for replay prevention.

Replay prevention is handled jointly by:

* Pantheon Grants, which include a Grant nonce and time window; replayed Grants outside their window MUST be rejected.

* Endpoints, which track nonces and session tickets, and MUST refuse to process 0-RTT requests that would not be safe to replay at the application level.

Authenticity alone is insufficient for transport-authorising artefacts. A well-signed Grant, resumption token, or related control artefact MUST also be evaluated for freshness, scope, nonce or sequence state, and current policy eligibility before the endpoint or RN relies on it.

If Pantheon policy specifies "no-replay" for a given Grant, endpoints MUST NOT use 0-RTT for any traffic under that Grant, and RNs MUST drop any early data tagged with that policy.

10. Post-Quantum Profiles and Crypto Agility

Given the long-term horizon of identity-centric networking, UZP is designed to support post-quantum (PQ) algorithms:

* Cipher suites define both classical and PQ-capable KEMs and signature schemes.

* Pantheon policy can enforce that certain tenants or epochs require PQ-only or hybrid key exchange.

* The handshake format allows negotiation of PQ suites in parallel with classical ones, similar to ongoing PQ-TLS efforts [NIST-PQC].

Transport endpoints SHOULD support at least one PQ KEM and one PQ-capable signature algorithm as they become standardised.

11. Rendezvous Node Behaviour

RNs are designed to be minimally trusted intermediaries. An RN:

* Accepts outbound connections from EPs and validates their Pantheon-issued credentials and Grants.

* Matches Join requests between EPs and stitches them into ZPITs according to policy.

* Relays encrypted blocks between stitched endpoints, potentially applying rate limits, traffic shaping, and policy enforcement on encrypted metadata.

Importantly, an RN:

* MUST NOT terminate end-to-end UZP AEAD protection.

* MUST NOT hold long-term secrets for EP identities.

* MAY observe and act on limited metadata (e.g., CIDs, block counts, timing) comparable to the exposure in QUIC’s on-path model [RFC9000].

An RN is therefore a relay and policy-constrained coordination point, not a sovereign source of identity, Grant, revocation, or other authorisation truth.

For high-assurance deployments, multi-RN topologies similar to onion routing may be used [TOR2004], with attestation chains that prove that each RN conforms to specified software and configuration baselines.

11.1. RN Set Discovery and Eligibility

UZP endpoints that require resilient rendezvous SHOULD maintain one or more currently valid RN Set Statements, or an equivalent accepted RN set carried by bootstrap or recovery artefacts, as defined by UZPIF ([UZPIF]). RN referral by a currently attached RN MAY be used as a hint, but it is not sufficient by itself to make an alternate RN eligible for authorisation-relevant use.

At baseline, before first use or failover use, the endpoint MUST validate the RN Set Statement's signature set, scope, validity interval, and epoch or sequence state under the UZPIF common envelope rules and prefer the highest eligible non-conflicting statement for the relevant scope. Where policy permits, endpoints SHOULD keep multiple eligible alternate RNs rather than a single spare path.

An endpoint MAY attempt failover only to an alternate RN that is currently policy-eligible for the relevant transport role, trust-domain scope, and session class. Eligibility MUST be derived from externally verifiable artefacts rather than from one RN's private control view.

A previously successful RN does not remain eligible merely because it stitched earlier traffic. The endpoint MUST re-evaluate alternate-RN eligibility at failover time against current RN Set Statements, current policy scope, and current distrust or revocation state.

11.2. RN Failover Triggers

Endpoints SHOULD treat at least the following conditions as baseline failover triggers:

* loss of RN reachability, handshake timeout, or repeated Join failure;
* authenticated or otherwise policy-accepted overload, maintenance, or capacity-exhaustion signalling;
* signed or otherwise locally accepted evidence of RN misbehaviour;
* detected inconsistency between the RN's control claims and the endpoint's currently accepted RN Set Statement, Grant state, or transparency evidence; and
* suspected RN partition or divergence that prevents safe continuation of authorisation-relevant operation.

Triggering failover changes path selection and rendezvous handling; it does not, by itself, revoke identity, Grants, or other externally verifiable authority. RN loss is an availability event, not an authorisation event.

11.3. Endpoint Failover Procedure

When a baseline failover trigger occurs, the endpoint MUST at least:

1. transition the association into "failover / rebind" or "closed" and stop treating RN-local control state as authorisation truth;

2. freeze new authorisation-expanding actions until required revalidation succeeds;

3. retain only the portable state allowed by Section 11.5;

4. select an alternate RN only from a currently accepted RN Set Statement or equally trusted bootstrap or recovery artefact;

5. establish a new RN attachment and perform a new Join evaluation at the alternate RN;

6. present current authorisation artefacts by default, or bounded continuity state only when non-expanding continuity is permitted by policy; and

7. return to "stitched" only after alternate-RN eligibility and required revalidation checks succeed; otherwise remain unavailable or terminate cleanly.

If no eligible alternate RN can be validated, the association remains unavailable. That is an availability outcome, not an implicit revocation of externally verifiable authority.

11.4. Re-Stitching and Continuity State

Re-stitching at an alternate RN requires a new RN attachment and Join evaluation. Fresh presentation of current authorisation artefacts is the default baseline. An alternate RN MAY accept bounded continuity state instead of a fully fresh authorisation presentation only for non-expanding continuity when local policy permits it.

Bounded continuity state in UZP MUST be endpoint-presented and cryptographically bound to the authenticated peer identities, the still-valid Grant object identifiers or digests, the accepted RN set, the relevant session or resumption context, and a short expiry. It MUST NOT be an opaque RN-issued assertion that another RN is required to trust blindly.

An alternate RN MAY reuse bounded continuity state only when the resulting session scope is unchanged or narrower, the referenced Grant and revocation state remain valid, freshness and replay checks succeed, and no unresolved RN disagreement would make the continuity claim ambiguous. Otherwise the endpoint MUST perform a fresh authorised re-stitch.

Before moving from "failover / rebind" back to "stitched", the endpoint and alternate RN MUST revalidate at least the alternate RN's eligibility for the requested role and scope, the currently accepted RN Set Statement or equally trusted bootstrap or recovery artefact, peer identity binding and current Grant or equivalent authorisation state, current revocation or transparency state relevant to the requested action, and the binding, freshness, replay constraints, and expiry of any continuity state presented.

11.5. Portable Versus RN-Local State

The baseline portability rule for UZP failover is:

* resumable or re-usable state MAY include peer identity bindings, current Grant references, accepted RN Set Statement identifiers, validated revocation or transparency checkpoints, and bounded continuity state accepted under policy;

* endpoint-local continuity markers MAY include application mapping state or block-delivery checkpoints only when they remain bound to the same peer identities, Grant scope, and continuity window; and

* RN-local state such as stitch handles, relay queue placement, pacing and congestion state, per-RN replay allocations, path observations, and unverified relay buffers MUST be treated as non-portable and MUST be renegotiated or conservatively rebuilt.

Session continuity therefore does not imply trust continuity. A resumed or re-stitched transport path MUST still revalidate the authorisation artefacts relevant to the new RN attachment.

The fact that some state is portable does not remove the need to revalidate it. Portable state MAY be resumed only after its bindings, scope, freshness, and policy eligibility are rechecked for the new RN attachment.

11.6. RN Disagreement and Partition Behaviour

If a currently attached RN and an alternate RN present conflicting control views for the same session, scope, or Grant state, endpoints MUST treat the conflict as an availability or continuity issue rather than as proof that either RN is the mandatory truth source.

Under RN disagreement or suspected partition, endpoints MUST suspend new authorisation-expanding actions such as new peer scopes, broader Grants, QoS expansion, or 0-RTT resumption that depends on disputed state until externally verifiable artefacts are revalidated. Endpoints MAY continue narrowly scoped, already-established non-

expanding traffic only within an explicit continuity window allowed by local policy and MUST cease or cleanly terminate once that window expires or required revalidation fails.

When disagreement cannot be resolved safely, implementations SHOULD prefer clean session termination or fresh re-stitching over silent failover that preserves apparent continuity without revalidation.

If only one RN remains reachable during disagreement or partition, endpoints MUST still treat that RN as a relay path rather than as a truth anchor. They MUST ignore RN-local claims that would expand authority unless those claims are independently supported by externally verifiable artefacts.

12. Proof-of-Reachability (PoR)

Informal Ping or heartbeat semantics are replaced by Proof-of-Reachability (PoR) for transport liveness validation.

When PoR challenges, responses, or derived liveness evidence are retained beyond the immediate exchange, they MUST be represented as common signed artefacts using the envelope defined by UZPIF ([UZPIF]) with "object_type" set to "por-evidence" when interoperable verification or audit is required. That object type inherits the UZPIF common envelope unchanged, including canonical serialisation, exact signature coverage, object_id derivation, unknown-extension handling, signature ordering, algorithm identifier matching, epoch-versus-sequence precedence, and the rule that detached signatures are not part of baseline interoperability.

Base UZP defines PoR as an authenticated reachability proof bound to the peer identity, a fresh nonce, the RN identifier, an expiry condition, and session context. The proof is produced by a profile-defined authenticated responder and MUST be verifiable by the client without trusting RN assertions.

Base UZP does not require a single proof primitive for PoR. A profile may define the responder proof value as a signature, MAC, exporter-derived authenticator, or equivalent cryptographic proof, but the verifier MUST be able to determine exactly which bytes were authenticated and which authenticated responder construction was used for the challenged context.

A PoR succeeds only when the verifier accepts cryptographic proof material emitted by the endpoint-side authenticated responder for the challenged context. RN observation of relay traffic, queue activity, timing, packet return, or successful forwarding is not equivalent to endpoint liveness proof and MUST NOT be treated as such.

12.1. PoR Object Fields

When a PoR challenge, response, or retained liveness artefact is represented as an object, it MUST use the common signed artefact envelope defined by UZPIF ([UZPIF]) with "object_type" set to "por-evidence". In addition to the common envelope, a minimal PoR object MUST carry:

* the challenger identity, typically mapped to "audience_id" or an equivalent field;
* the responder identity, typically mapped to "subject_id" or an equivalent field;
* the RN identifier;
* session context sufficient to prevent replay across unrelated sessions, tunnels, or transport bindings;
* the PoR nonce;
* an issuance time and an expiry time;
* a proof-profile identifier indicating which authenticated responder construction was used;
* the responder proof value;
* a responder key identifier or exporter-label reference when needed for verification; and
* optional grant, transport-binding, or transparency references when the deployment needs them for audit or replay tracking.

These fields populate the PoR-specific body only. UZP inherits the UZPIF common envelope unchanged, including canonical serialisation, exact signature coverage, object identifiers, unknown extension handling, signature ordering, algorithm identifier matching, epoch-versus-sequence precedence, and the rule that detached signatures are not part of baseline interoperability.

When a PoR artefact is retained as a common signed object, the envelope signature set authenticates the retained artefact representation for exchange or audit. It does not replace the responder proof value as the liveness proof itself. Current reachability still depends on validating the responder proof against the profile-defined authenticated responder for the challenged context.

The responder proof value MUST authenticate the exact PoR input used by the selected proof profile. At a minimum, that authenticated input MUST cover the responder identity, the challenger identity or equivalent audience restriction, the RN identifier, the session context, the PoR nonce, the issuance and expiry values, the proof-profile identifier, and any grant, transport-binding, or transcript-binding values that the verifier is expected to enforce. If a profile authenticates a derived challenge representation rather than the retained PoR object verbatim, the verifier MUST still be able to determine exactly which PoR fields were authenticated.

12.2. PoR Exchange

A PoR exchange is defined as follows:

1. The client generates a fresh nonce for a specific PoR transaction. The nonce MUST be unique within the verifier’s active replay window for the relevant responder, RN context, and session class.

2. The PoR challenge MUST bind: the peer identity being tested, a fresh nonce, an RN identifier, an explicit expiry window, and session context sufficient to prevent replay across unrelated sessions or tunnels.

3. The RN relays the nonce to the target peer without modification.

4. The peer produces proof material using the profile-defined authenticated responder for the current session, covering the bound challenge fields and the authenticated session or transport context required by that profile.

5. The RN returns the responder proof to the client.

6. The client validates the returned proof against the exact challenged context, including nonce freshness, expiry, session-context binding, proof-profile identifier, and peer identity match, before accepting liveness.

Freshness evaluation MUST reject a PoR response if the nonce was not generated for the claimed transaction, if that nonce has already been accepted for the same responder or context, if the issuance or expiry values fall outside the verifier’s acceptance window, or if the authenticated session, resumption, or transport context is no longer current.

A verifier MUST also reject a response if the challenge expiry has elapsed before proof validation completes, if the authenticated RN identifier does not match the RN attachment under evaluation, or if the proof binds to an older session, resumption context, or continuity window than the one for which present reachability is being claimed.

12.3. PoR Across Re-Stitch and RN Failover

PoR evidence is path- and context-bound. A PoR response accepted under one RN attachment, session context, or continuity window MUST NOT by itself be treated as fresh liveness proof for a different RN attachment or re-stitched transport path.

After re-stitch or RN failover, endpoints MUST obtain a new PoR response bound to the new RN identifier and the currently authenticated session or continuity context before relying on PoR for current reachability on that path. Earlier PoR artefacts MAY be retained for audit, replay tracking, or continuity diagnostics, but they do not satisfy current liveness verification for the new attachment.

A previous PoR response MUST NOT be reinterpreted as current liveness merely because the same peer identities, Grant references, or application flow continue across failover. The verifier MUST obtain a fresh proof bound to the new RN attachment and current session or continuity context before treating the peer as presently reachable on that path.

A bounded continuity decision under Section 11.4 MAY allow traffic to continue briefly while revalidation proceeds, but it does not convert an older PoR bound to a previous RN or session into proof of present reachability.

12.4. TLS-DPA PoR Profile

The TLS-DPA profile instantiates the authenticated responder using TLS-DPA-authenticated identity material [TLS-DPA]. A deployment MAY use either:

* a dedicated PoR responder key derived from exporter material bound to the authenticated TLS-DPA or UZP session; or

* a proof directly generated with TLS-DPA-authenticated identity key material.

The TLS-DPA profile SHOULD prefer an exporter-derived PoR responder key for steady-state or frequent liveness checks because it reduces operational dependence on repeated use of long-term identity keys while preserving RN non-forgeability. Direct long-term identity-key signing MAY be used where policy explicitly requires identity-key participation or where exporter-derived responder material is unavailable.

In the TLS-DPA profile, exporter-derived responder material is the preferred baseline for ordinary PoR use. Long-term identity-key signatures are an exception path for deployments that explicitly require them, for recovery of exporter state, or for other policy-defined cases where exporter-derived proof material cannot be used.

Regardless of which method is used, the client MUST validate that the responder proof is cryptographically bound to a TLS-DPA-authenticated peer identity and to the relevant session context. The authenticated input MUST cover the PoR nonce, the asserted identities, the RN identifier, validity bounds, and the session or exporter binding required by the selected mode.

This draft intentionally does not define a byte-level PoR wire encoding for the TLS-DPA profile. Profile documents or future revisions may define concrete serialisations for PoR challenges, PoR responses, and retained PoR artefacts while preserving the object fields and validation rules specified here.

12.5. Replay Prevention

Replay prevention requirements for PoR are:

* Peers and verifiers MUST reject expired nonce values or challenges whose validity window has elapsed.

* Verifiers MUST accept a PoR response at most once for a given responder identity, RN identifier, nonce, and session-context tuple.

* RNs MUST NOT treat replayed forwarding, repeated observation, or cached PoR responses as fresh liveness, and MUST NOT cache PoR responses beyond their expiry window.

Replay prevention for PoR is therefore based on nonce uniqueness, bounded validity, exact context binding, and verifier-side single-acceptance tracking rather than on RN observation of traffic.

12.6. RN Non-Forgeability Requirement

The transport liveness model MUST ensure an RN cannot fabricate peer reachability.

* The peer PoR proof MUST cover the original nonce, peer identity, RN identifier, expiry condition, and session context.
* The client MUST validate the proof against the profile-defined authenticated responder associated with the peer identity, and MUST NOT rely on RN assertions alone for liveness.
* RNs MUST NOT generate synthetic PoR acknowledgements that are not produced by the authenticated responder bound to the peer identity and session context.
* RN-local evidence such as successful forwarding, packet observation, queue drain, or timing correlation MUST NOT be substituted for endpoint-generated PoR proof.

12.7. Liveness Proof Logging

RNs MAY publish aggregated PoR statistics in transparency logs. Publication is OPTIONAL but RECOMMENDED for deployments that use merit-based scoring or external audit signals.

Where PoR-related transparency artefacts are published for interoperable audit, they MUST use the common signed artefact envelope defined by UZPIF ([UZPIF]), with sequence or log linkage sufficient for independent audit.

13. Security Considerations

UZP’s security properties derive from:

* The zero-port architecture of UZPIF ([UZPIF]) (no open listeners).
* The use of modern AEAD algorithms with at least 96-bit tags.
* Identity-first addressing via CIDs and EIDs, influenced by HIP [RFC7401].
* Exporters that bind higher-layer security directly to transport identities and parameters.
* Strong replay controls via Grants and endpoint tracking of nonces and tickets.

* PoR liveness verification with profile-authenticated nonce evidence that limits RN forgery.

Residual risks include traffic analysis, RN compromise (drop/delay behaviour), and application-layer weaknesses. These are mitigated through:

* Multi-RN topologies and attestation.

* Pantheon policy that can revoke identities and Grants quickly.

* Integration with Zero Trust principles [NIST-SP800-207].

14. IANA Considerations

This document makes no requests of IANA at this time because it does not yet define final wire encodings or complete registries. Future revisions of UZP may define registries for protocol parameters (for example cipher suites, frame types, and exporter labels), but such actions are out of scope for this transport-semantics revision.

* A registry for UZP cipher suites and AEAD algorithms.

* A registry for frame and block types.

* A registry for exporter labels used by UZP and UZPIF ([UZPIF]).

15. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

16. Informative References

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/info/rfc7401>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[UZPIF] Fisher, B. A., "Universal Zero-Port Interconnect Framework (UZPIF)", Work in Progress, Internet-Draft, draft-dpa-uzpif-framework, <https://datatracker.ietf.org/doc/html/draft-dpa-uzpif-framework>.

[TLS-DPA] Fisher, B. A., "TLS-DPA: An Identity-Bound Security Protocol for Traditional, Overlay, and Zero-Port Transports", Work in Progress, Internet-Draft, draft-dpa-tls-dpa, <https://datatracker.ietf.org/doc/html/draft-dpa-tls-dpa>.

[NIST-SP800-207] Rose, S., Borchert, O., Mitchell, S., and S. Connelly, "Zero Trust Architecture", NIST SP 800-207, 2019, <https://doi.org/10.6028/NIST.SP.800-207>.

[NIST-PQC] Technology, N. I. O. S. A., "NIST Post-Quantum Cryptography Standardization: Fourth Round Candidate Algorithms", 2022, <https://csrc.nist.gov/Projects/post-quantum-cryptography>.

[TOR2004] Dingledine, R., Mathewson, N., and P. Syverson, "Tor: The Second-Generation Onion Router", USENIX Security Symposium, 2004.

Acknowledgements

The author thanks colleagues and early reviewers for discussions on identity-centric networking, rendezvous transports, and post-quantum transition considerations. Any errors or omissions remain the author’s responsibility.

Author’s Address

Benjamin Anthony Fisher
DPA R&D Ltd (https://www.dpa-cloud.co.uk)
Email: b.fisher@dpa-cloud.co.uk
URI: https://orcid.org/0009-0004-4412-2269