Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 18 October 2026

J. Hause
Independent

16 April 2026

ASIP: AS-Structured Internet Protocol (128-bit)
draft-hause-asip-00

Abstract

ASIP (AS-Structured Internet Protocol; IP version 8 on the wire) is a 128-bit network protocol that defines a new address family alongside IPv4. ASIP is NOT a wire-level superset of IPv4: an unmodified IPv4 device cannot send, receive, or forward an ASIP packet. Interoperation between ASIP-aware endpoints and legacy IPv4 networks is provided by a defined transition mechanism (stateless translation at AS boundaries and encapsulation across non-upgraded transit), not by wire compatibility.

ASIP's addressing architecture arranges the 128-bit address as four 32-bit fields (ASN routing locator, zone, subnet, host). The ASN locator is used as a routing hint rather than a hard identity binding; multihoming, ASN transfer, and cross-ASN anycast remain possible through multi-address semantics defined in Sections 8, 11, and 14. This structure is intended to reduce operational friction during transition (familiar dotted-decimal notation, clean aggregation at the ASN boundary) rather than to achieve wire-level IPv4 compatibility.

This document specifies the core protocol: address format, packet header, address classes, routing behavior, transition mechanisms, and security considerations. The Zone Server reference architecture (Section 16) is Informative and non-normative. The Cost Factor routing metric (Section 17) is OPTIONAL and is specified in a separate companion document (draft-asip-cf-00); Section 17 of this document is a one-paragraph forward reference. Interplanetary realm reservations (Section 3.12) are reserved allocations only; delay-tolerant transport is out of scope for this document. Operators MAY deploy ASIP addressing without any of the above.

This specification extends the ASN-locator addressing model of [I-D.thain-ipv8] to a 128-bit four-field address format; see Section 1.3 for the relationship between the two proposals.

Status of This Memo

Copyright Notice

Table of Contents

1. 1. Introduction

1.1. 1.1. Requirements Language

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 BCP 14 RFC2119 [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. 1.2. Design Philosophy

ASIP is built on three premises.

First, ASIP is a new address family with a defined transition mechanism. It is not a wire-level superset of IPv4. Any packet carrying Version=8 in the IP header will be dropped by unmodified IPv4 routers, hosts, and middleboxes; conversely, ASIP-aware nodes receive IPv4 traffic only through the transition mechanisms defined in Section 12. The goal is not wire compatibility, which is unachievable without changing the framing layer. The goal is a transition path that minimizes operator-facing churn: familiar dotted-decimal notation, natural mapping of existing IPv4 host addresses into the ASIP host field, and stateless translation at AS boundaries so that legacy IPv4 endpoints remain reachable by ASIP-aware peers without application-layer changes.

Second, the address format should reflect routing topology so that aggregation is natural and deaggregation is visible. Arranging the 128-bit address as ASN/Zone/Subnet/Host makes the default aggregation boundary obvious and provides a single-lookup inter-AS forwarding path for the common case. The ASN field is a routing locator used as a forwarding hint, not a globally binding identity; Sections 8, 11, and 14 define how multihomed, anycast, and transferred ASNs are handled through multi-address semantics rather than through a one-ASN-per-host rule.

Third, a protocol specification must not exceed its scope. ASIP defines an address format, a packet header, and the routing behaviors those imply. Operational tooling, management platforms, authentication frameworks, and forward-looking realm designs are Informative only. An operator MAY adopt ASIP addressing without adopting any of them.

1.3. 1.3. Relationship to Prior Work

This specification extends the ASN-locator addressing concept of [I-D.thain-ipv8] from a 64-bit two-field address (ASN / Host) to a 128-bit four-field address (ASN / Zone / Subnet / Host). The dotted-decimal r.r.r.r notation for the ASN locator, the concept of encoding Autonomous System identity into the address prefix, and the Zone Server management architecture described in Section 16 (Informative) are derived from that prior work. The extensions contributed by this document are: the 128-bit address format with Zone and Subnet fields, the stateless translation mechanism defined in Section 12, the ingress-filter mechanism defined in Section 14.1, the byte-level encoding specifications in Section 12.7, and the security-considerations treatment in Section 14.

[I-D.thain-ipv8] remains a distinct proposal with different addressing semantics; this document does not supersede it and does not claim to. The two specifications MAY coexist as parallel proposals in the IETF process.

1.4. 1.4. What ASIP Is Not

ASIP is not a wire-compatible extension of IPv4. It is not a network management suite. It does not, at the protocol layer, mandate specific authentication mechanisms, logging formats, route-validation registries, or device firmware behaviors. Section 16 (Zone Server) is Informative, Section 17 (Cost Factor) is OPTIONAL and deferred to a companion document, and Section 3.12 (realm reservations beyond Earth) is reserved allocation only. The core protocol’s correctness does not depend on any of them; a compliant ASIP implementation can ignore all three.

The core address-layer specification stands alone: the packet header of 則5, the address format of 則3, and the link-local, mesh, and realm scopes of 則則3.10-3.12 are self-contained and do not depend on any companion protocol. Inter-AS routing, route-ownership validation, and transit across non-upgraded IPv4 networks require the extensions named in Sections 8 and 12, and those extensions are REQUIRED for the use cases that depend on them; they are not part of the core address-layer specification.

2. 2. Motivation and Problem Statement

2.1. 2.1. Address Exhaustion

IANA completed allocation of the IPv4 unicast address space in February 2011. Regional Internet Registries exhausted their pools between 2011 and 2020. CGNAT has extended IPv4's operational life at the cost of added latency, broken peer-to-peer protocols, complicated troubleshooting, and centralized failure domains.

The address exhaustion problem is architectural and cannot be resolved within a 32-bit address space.

2.2. 2.2. Routing Table Growth

The BGP4 global routing table exceeded 1,000,000 prefixes by 2024 and grows without architectural bound. Prefix deaggregation for traffic engineering purposes is the primary growth driver. No BGP4 mechanism structurally prevents it.

BGP4 has no binding relationship between what an ASN advertises and what it is authorized to advertise. Prefix hijacking, route leaks, and bogon injection remain possible because route ownership validation is not enforced as a condition of route acceptance. RPKI and BGPsec have seen limited deployment due to operational complexity and incomplete adoption incentives.

2.3. 2.3. Transition Model Failure

IPv6 required dual-stack operation: every device, application, and network supporting both IPv4 and IPv6 simultaneously. This model imposed cost with no incremental benefit to early adopters. Organizations that deployed IPv6 still required full IPv4 support for the foreseeable future. The absence of a forcing function allowed indefinite deferral.

Any viable successor must provide value to individual adopters before universal adoption occurs.

2.4. 2.4. Requirements for a Viable Successor

These are the goals ASIP sets for itself. Wire-level IPv4 compatibility and zero-modification deployment are not among them: both are unachievable at the framing layer once Version=8 is used, because unmodified IPv4 forwarding elements discard packets whose Version field is not 4.

* *R1.* Defined transition mechanism. ASIP-aware endpoints MUST be able to reach unmodified IPv4 endpoints through stateless translation and encapsulation mechanisms specified in Section 12. Wire-level IPv4 compatibility on the Version=8 code point is explicitly not claimed.

* *R2.* Single-stack operation on the end host. An ASIP-aware host MUST be able to reach both IPv4 and ASIP destinations through a single ASIP socket API; dual stacks in the application layer are not required. Network paths in transition will carry both IPv4 and ASIP frames; that is unavoidable and is not called "dual stack" here.

* *R3.* Expanded address space sufficient to eliminate exhaustion and CGNAT dependency and to reduce forced renumbering for the common case of upstream-provider change within a fixed ASN. The requirement is "reduce" rather than "eliminate" because the z:s:h identifier triple of 則3.1 is stable across ASN transfer and locator rewrite, but the r.r.r.r field of any affected address changes per 則8.3.2, and in that narrow sense some renumbering events persist. Residual renumbering events include (a) ASN transfer between organizations (則8.3.1), (b) multihoming add/drop where the set of r.r.r.r locators advertised for a host changes, (c) locator rewrite at an AS boundary (則8.3.2; not visible to most applications per 則3.1’s identifier/locator separation), (d) acquisition-driven merger of two ASNs into one, and (e) RIR policy revocation of an ASN. Common-case provider-change (an enterprise swapping upstream ISPs while keeping its own ASN) produces zero renumbering under ASIP, which is the benefit R3 claims.

* *R4.* Structurally bounded global routing table in the common case, with multihoming, anycast, and traffic engineering preserved through the mechanisms defined in Sections 8.3, 11.1, and 14.

* *R5.* Route ownership validation defined at the protocol layer (Section 8.4). Because route-ownership validation is a core guarantee, it is REQUIRED for any deployment that relies on the structural routing-table bound; it is RECOMMENDED but not REQUIRED for early deployments that accept BGP4-equivalent route hygiene.

* *R6.* Implementable via software update on existing forwarding hardware (line cards, NICs, kernels, middleboxes). Hardware that cannot be updated to parse a 40-byte Version=8 header is incompatible with ASIP and MUST be replaced or bypassed via tunnels at the deployment boundary.

* *R7.* Human-readable addressing consistent with IPv4 operator familiarity.

* *R8.* Incremental adoption value at the AS boundary. An AS that deploys ASIP internally and at its borders benefits before universal adoption; transit across non-upgraded IPv4 networks uses the mechanisms of Section 12.

* *R9.* No flag day for legacy IPv4 endpoints; they continue to operate unchanged and are reached via translation. There is, however, a per-device software-update flag day for any node that wishes to participate in ASIP forwarding or origination.

* *R10.* Clear separation between protocol-layer requirements and operational recommendations (Sections 16, 17).

3. 3. ASIP Address Format

3.1. 3.1. Structure

An ASIP address is a 128-bit value composed of four 32-bit fields:

r.r.r.r . z.z.z.z . s.s.s.s . h.h.h.h | | | | | | | ASN Zone Subnet Host Locator ID ID ID (32 bit) (32 bit) (32 bit) (32 bit)

* *r.r.r.r* -- ASN Routing Locator. A 32-bit value drawn from the Autonomous System Number space and used as the inter-AS forwarding key. The ASN locator is a routing hint; it is NOT a globally binding identity. A single host MAY be reached via multiple addresses with different r.r.r.r values, and a single ASN MAY be advertised from multiple locations. Rewriting of r.r.r.r at ingress or egress is permitted under the conditions defined in Section 8.3 and Section 11.1.

* *z.z.z.z* -- Zone Identifier. Identifies a network zone, region, or organizational division within the AS. Allocated by the AS operator. Locally significant.

* *s.s.s.s* -- Subnet Identifier. Identifies a subnet within a zone. Allocated by the zone operator. Locally significant.

* *h.h.h.h* -- Host Identifier. Identifies a host within a subnet. Its value space is the same as an IPv4 host address so that existing operational practice (well-known host addresses, DHCP pools, broadcast conventions) carries over without reinterpretation.

Identifier/locator separation: r.r.r.r is a routing locator only, not a host-identity field. Binding ASN identity into the address as a hard property of the host would break multihoming, ASN transfer, and cross-ASN anycast: a multihomed host reachable through two ASes

could not present a single identity, an ASN transfer would force every affected host to acquire a new identity, and a single anycast service could not be advertised from multiple ASNs. Under the locator-only semantics defined here, a multihomed host has one address per upstream AS it is reachable through; an anycast service has one address per ASN it is advertised from; an ASN transfer rewrites the r.r.r.r locator of the affected addresses without changing the underlying host. The z.z.z.z, s.s.s.s, and h.h.h.h fields collectively form a stable identifier within an operator's scope; r.r.r.r is allowed to change as reachability changes. Section 8 defines the routing-layer consequences; Section 14 defines the security consequences.

3.2. 3.2. Address Space

Total: 2^128 = 340,282,366,920,938,463,463,374,607,431,768,211,456 Per ASN: 2^96 = 79,228,162,514,264,337,593,543,950,336 Per Zone: 2^64 = 18,446,744,073,709,551,616 Per Subnet: 2^32 = 4,294,967,296

This is identical in total size to IPv6 (2^128) but with a fixed hierarchical structure that directly encodes routing topology.

3.3. 3.3. IPv4-Mapped ASIP Addresses

When the ASN Locator, Zone ID, and Subnet ID are all zero, the Host ID field encodes an IPv4 address:

0.0.0.0 . 0.0.0.0 . 0.0.0.0 . h.h.h.h

This form is analogous to IPv6's ::ffff:0:0/96 IPv4-mapped address block [RFC4291]. It is a representation inside the ASIP address space of an IPv4 destination; it is NOT a wire-level IPv4 packet. An ASIP-aware host that wishes to reach an unmodified IPv4 endpoint constructs an IPv4-mapped destination and passes it to the ASIP stack, which either:

1. Forwards the traffic as a Version=8 frame to an ASIP/IPv4 translator at the AS boundary (Section 12.2), which emits Version=4 on the downstream IPv4 network; or

2. Encapsulates the Version=8 frame in a Version=4 packet for transit across non-upgraded IPv4 infrastructure (Section 12.3).

An unmodified IPv4 device cannot send or receive a Version=8 frame directly. The IPv4-mapped address form exists so that the ASIP stack and application layer have a single address type for both ASIP and IPv4 destinations, not so that IPv4 hosts are silently reclassified as "ASIP compliant."

An ASIP implementation that receives a Version=8 packet where r.r.r.r = 0.0.0.0, z.z.z.z = 0.0.0.0, and s.s.s.s = 0.0.0.0 MUST treat the h.h.h.h field as an IPv4 destination and forward it toward the IPv4/ ASIP translator selected by Section 12.

r=0 with z≠0 or s≠0. The IPv4-mapped form above requires all three of r, z, s to be zero. The partial form (r.r.r.r = 0.0.0.0 with z.z.z.z or s.s.s.s non-zero) is not defined by this document and, consistent with the AS 0 reservation of [RFC7607], is not a valid origin or destination for ASIP traffic. ASIP implementations MUST drop any received Version=8 packet whose source or destination r.r.r.r = 0.0.0.0 is accompanied by a non-zero z.z.z.z or s.s.s.s, and SHOULD log the event.

Realm containment exception: Section 4.1 defines strict scope containment: an address in realm X is not visible outside realm X. IPv4-mapped addresses (r=z=s=0) are an explicit exception to the terrestrial-realm containment rule because they represent a legacy IPv4 destination that has no realm of its own. IPv4-mapped traffic is handled by translators at the terrestrial/realm boundary and is never forwarded natively into a non-terrestrial realm.

3.4. 3.4. ASN Encoding

The 32-bit ASN is encoded directly into the r.r.r.r field as a 32-bit unsigned integer in network byte order:

ASN 64496 = 0.0.251.240 (documentation, per RFC 5398) ASN 64497 = 0.0.251.241 (documentation, per RFC 5398) ASN 15169 = 0.0.59.65 (example: Google) ASN 13335 = 0.0.52.23 (example: Cloudflare)

3.5. 3.5. Internal Zone Prefix (127.0.0.0/8 in r.r.r.r)

The r.r.r.r range 127.0.0.0/8 is permanently reserved for internal zone prefixes. These addresses identify network zones within an organization’s private addressing space and are never routed externally.

127.1.0.0 . z.z.z.z . s.s.s.s . h.h.h.h (Internal zone 1) 127.2.0.0 . z.z.z.z . s.s.s.s . h.h.h.h (Internal zone 2) 127.3.0.0 . z.z.z.z . s.s.s.s . h.h.h.h (Internal zone 3)

Internal zone prefix rules:

* MUST NOT be routed beyond the organization’s AS boundary.

* MUST NOT appear on WAN interfaces or public internet links.

* MUST NOT appear in eBGP-ASIP advertisements.

* MAY be used freely within an organization’s routing infrastructure.

* Provides 2^(24+32+32+32) = 2^120 effective internal addresses per organization (24 free bits in the r.r.r.r field after the fixed 127/8 prefix, plus the full Zone, Subnet, and Host fields). The same 127.0.0.0/8 prefix space is reused independently inside every organization; the 2^120 figure is a per-organization capacity, not a globally partitioned one.

Organizations may build geographically distributed, multi-region private networks of arbitrary scale without external address coordination and with zero possibility of zone-to-zone address conflict.

ASN numbers that encode to the 127.0.0.0/8 range (ASN 2,130,706,432 through ASN 2,147,483,647) are reserved for internal zone use and MUST NOT be allocated by IANA for public internet routing.

3.6. 3.6. Inter-Organization Interop Prefix (127.127.0.0)

The prefix 127.127.0.0 in the r.r.r.r field is reserved as a standard inter-organization interconnect DMZ. When two organizations need to peer without exposing internal addressing:

Org A Org B ----- ----- 127.1.0.0.z.s.h <-> XLATE <-> 127.127.0.0.z.s.h <-> XLATE <-> 127.2.0.0.z.s.h

Neither organization exposes its internal zone topology to the other. Each controls exactly what it publishes into the shared 127.127.0.0 space.

3.7. 3.7. RINE Peering Prefix (100.0.0.0/8 in r.r.r.r)

The r.r.r.r range 100.0.0.0/8 is reserved for Regional Inter-Network Exchange (RINE) peering fabric addressing. RINE addresses are used exclusively for AS-to-AS peering links at IXPs and private interconnect facilities.

* MUST NOT be advertised in the global eBGP-ASIP routing table.

* MUST NOT be assigned to end devices.

* MUST be filtered at all eBGP-ASIP border routers.

3.8. 3.8. Interior Link Convention (222.0.0.0/8 in h.h.h.h)

The h.h.h.h range 222.0.0.0/8 is designated by convention for router-to-router interior link addressing within an AS. This is a soft convention, not a scope boundary.

Two distinct filtering behaviors apply and MUST NOT be conflated:

* *Route-advertisement filtering (Section 14.4):* eBGP-ASIP border routers MUST NOT advertise prefixes whose covered addresses have h.h.h.h in 222.0.0.0/8 as reachable interior links. This prevents interior-link addressing from being exported to other ASes.

* *Data-plane filtering:* Border routers MUST NOT filter data packets solely on the basis of a 222.0.0.0/8 h.h.h.h value. The host field is locally significant; other ASes MAY use 222.x.x.x host addresses for any purpose and their traffic must transit normally. Filtering h.h.h.h values at border routers would break legitimate inter-AS traffic from operators who use this range internally.

Conflating the two would cause legitimate external traffic to be dropped whenever a remote operator happened to use 222.x.x.x host values internally; the two behaviors are therefore kept distinct as a normative requirement.

3.9. 3.9. Private ASN Reservations

Consistent with RFC 6996:

* ASN 65534 (r.r.r.r = 0.0.255.254): Reserved for private inter-organization eBGP-ASIP peering.

* ASN 65533 (r.r.r.r = 0.0.255.253): Reserved for documentation and testing.

3.10. 3.10. Link-Local Scope (169.254.0.0/16 in r.r.r.r)

The r.r.r.r range 169.254.0.0/16 is reserved for link-local addressing, directly analogous to both IPv4’s APIPA (169.254.0.0/16) and IPv6’s fe80::/10. The familiar range was chosen deliberately so that operators immediately recognize the scope.

169.254.0.0:0.0.0.0:0.0.0.0:h.h.h.h Compressed: 169.254.0.0::h.h.h.h

Link-local rules:

* MUST NOT be forwarded by any router. Link-local addresses are valid only on the physical or virtual link where they originate.

* MUST be usable without any prior configuration, DHCP, or router contact. A device that has received no configuration of any kind MUST be able to generate and use a link-local address.

* MUST be used for neighbor discovery, router solicitation, and SLAAC-ASIP bootstrapping (Section 3.14).

* The h.h.h.h field is self-assigned by the device (see Section 3.14 for generation rules).

* Multiple link-local addresses MAY exist on a single interface.

Link-local addresses serve the same critical bootstrapping role as fe80:: in IPv6: they provide a communication channel that exists before any infrastructure (DHCP, DNS, routing) is available. Every ASIP interface MUST have at least one link-local address at all times.

3.11. 3.11. Mesh and Ad-Hoc Scope (240.0.0.0/8 in r.r.r.r)

The r.r.r.r range 240.0.0.0/8 is reserved for mesh and ad-hoc network addressing. This range is used by devices participating in self-organizing networks where no AS infrastructure exists: disaster recovery, battlefield mesh, sensor networks, and similar environments.

240.x.x.x:z.z.z.z:s.s.s.s:h.h.h.h

Mesh scope rules:

* MUST NOT be routed beyond the mesh boundary.

* MAY be bridged to AS-routed networks via a mesh gateway that performs address translation for unicast traffic. Multicast is NOT forwarded across a mesh gateway in either direction per § 10.6 rule 5; application-layer relay is the only sanctioned path for cross-mesh multicast delivery.

* Devices within a mesh self-assign addresses using SLAAC-ASIP (Section 3.14) with DAD.

* The x.x.x portion of 240.x.x.x MAY be used to identify distinct mesh domains.

3.12. 3.12. Realm Architecture and Interplanetary Addressing (Informative)

This subsection is INFORMATIVE. The reservations described here set aside r.r.r.r ranges and nothing more; no delay-tolerant transport, no inter-realm relay, and no non-terrestrial routing protocol is defined by this document. A compliant ASIP implementation MAY ignore this section entirely except to filter the reserved ranges as unallocated.

ASIP reserves ranges in the r.r.r.r field for network realms beyond the terrestrial internet. A realm is a physically or logically distinct routing domain where light-speed delay, intermittent connectivity, or governance boundaries make standard BGP convergence assumptions invalid.

The rationale is forward-looking: deep-space networking (DTN) is already standardized via the Bundle Protocol [RFC9171] but has no native IP-layer addressing scheme. Reserving realm prefixes now costs nothing and avoids a renumbering event later. The reservations are held unallocated until a companion specification defines how they are to be used.

Realm Allocations:

r.r.r.r Range	Realm	Status
0.0.0.1 - 96.255.255.255	Terrestrial (Earth)	Active
97.0.0.0/8	Near-Earth Infrastructure	Reserved
98.0.0.0/8	Cislunar / Lunar	Reserved
99.0.0.0/8	Martian	Reserved
241.0.0.0 - 249.255.255.255	Future Celestial Bodies	Reserved
250.0.0.0/8	Delay-Tolerant (DTN) Relay	Reserved

Table 1

Realm properties:

(The bullets below are descriptive guidance on operating characteristics; the RFC 2119 keywords within them are normative only for operators who voluntarily deploy into the named realm per a future companion DTN profile. None of these bullets imposes a conformance requirement on ASIP core implementations, which per the section opener MAY ignore §3.12 entirely.)

* *Near-Earth (97.0.0.0/8):* LEO and GEO satellite constellations, space stations, orbital infrastructure. Round-trip times up to ~600ms. Standard TCP is functional with tuning. eBGP-ASIP peering is viable with extended timers.

* *Cislunar/Lunar (98.0.0.0/8):* Earth-Moon communication. ~2.5 second round-trip. TCP is marginal; DTN overlays are RECOMMENDED within the scope of a companion profile. eBGP-ASIP peering within that companion profile would require hold timers of 30+ seconds.

* *Martian (99.0.0.0/8):* Earth-Mars communication. 4-24 minute one-way delay depending on orbital position. TCP is non-functional. Any companion deployment profile for this realm would necessarily use DTN store-and-forward; eBGP-ASIP is not applicable and a delay-tolerant routing protocol is needed (out of scope for this document).

* *DTN Relay (250.0.0.0/8):* Addresses for DTN relay nodes that bridge between realms. In a future companion profile these nodes would participate in multiple realms and perform store-and-forward routing across light-speed boundaries; this document defines no such behavior.

Each realm operates as an independent routing domain. Inter-realm traffic is forwarded by relay gateways that bridge the appropriate delay-tolerant transport. The realm prefix in r.r.r.r allows any router to determine the destination realm from a single 32-bit lookup and route to the appropriate relay gateway.

The remainder of this subsection is Informative guidance for operators who elect to experiment with realm-scoped deployments; normative MUSTs and MUST NOTs below are normative only within such deployments and have no effect on implementations that ignore §3.12 entirely per the opening paragraph of this subsection. The normative core-protocol rules that cover cross-realm scope violations for any deployment are in §14.11 and §10.6.

Realm identification is implicit in r.r.r.r. There is no separate realm-ID field in the ASIP header (§5.1). A border relay determines the destination realm by looking up the destination address's r.r.r.r value against the realm-allocation table above; the source realm is

determined the same way from the source address. No packet-format changes are required to carry realm identity, and no companion protocol defines a realm-ID explicit field in this document.

Mesh/realm and mesh-as-relay interaction. A mesh node (§ 3.11, 240.0.0.0/8 in r.r.r.r) is by construction scoped to its mesh domain and does not hold an address in any non-terrestrial realm range (97/8, 98/8, 99/8, 241/8249/8, 250/8). A realm relay is, by definition, a border node that holds addresses in both the terrestrial realm and the non-terrestrial realm it bridges; a mesh node holds neither (its r.r.r.r is in 240/8) and therefore cannot be a realm relay. Operators who wish a physically-mesh deployment to participate in realm-crossing SHOULD front that deployment with a terrestrial AS boundary (a mesh gateway per § 3.11) and attach that AS to a realm relay via standard terrestrial peering. A node SHOULD NOT simultaneously hold a mesh (240/8) and a non-terrestrial realm (any of 97/8, 98/8, 99/8, 241/8249/8, 250/8, as enumerated earlier in this paragraph) r.r.r.r address on the same interface; dual-homing across the two scopes is NOT RECOMMENDED and should use a gateway node with two interfaces if attempted.

Delay-tolerant profile for core ASIP machinery. Several pieces of core ASIP machinery have implicit sub-second-RTT assumptions that do not hold at interplanetary distances:

* *SLAAC-ASIP DAD (§ 3.14.3)* expects a probe response within RetransTimer (IPv6 default 1000 ms); at cislunar RTT (~2.5 s) DAD timers require at minimum 4x extension and at Martian RTT (minutes) DAD as specified is inoperable.

* *NDP neighbor-cache aging (§ 9.2)* inherits IPv6 defaults (ReachableTime on the order of 30 s); link-level reachability assumptions break when a "link" is a light-minute.

* *PMTUD (§ 12.8)* assumes ICMP-ASIP Packet Too Big arrives within a data-flow RTT; at realm scale PMTU discovery would stall.

* *Translator per-flow state (§ 14.15)* ages at TCP/UDP-scale defaults; per-flow state for a cross-realm flow would expire before the first round-trip completes.

* *eBGP-ASIP hold timers (§ 8.3)* default in the single-digit-minutes range; cislunar is tractable with tuning, Martian is not.

This document does NOT define a DTN profile that adjusts the above. Until a companion DTN profile is published, § 3.12 realm reservations 98/8, 99/8, 241/8249/8, and 250/8 are *reserved address space only* and should not be deployed over the public internet for real traffic.

97/8 (Near-Earth) operates at sub-second RTT and is the only realm range for which the core ASIP timers above are tractable with standard tuning. This restricts interplanetary realm use to reserved-address status until a DTN profile exists; that restriction is a scope choice, not a protocol defect.

Cross-realm authentication (Informative guidance; core filtering rule is normative in §14.11). A packet arriving at a terrestrial realm relay with a source r.r.r.r in a remote realm's range (e.g., 99.x.x.x claiming to originate on Mars) is trivially spoofable within the local realm unless the relay verifies the packet's origin through some mechanism outside the ASIP header. This document does not define a cross-realm authentication mechanism; the companion DTN profile should define one (e.g., BPSec [RFC9172] over the Bundle Protocol carrying ASIP as payload, or a relay-to-relay IPsec/ESP tunnel anchored to published realm-relay identity). Until that profile exists, a realm relay SHOULD treat packets sourced from a remote realm as authenticated only when received on a physical or cryptographic channel that is itself bound to the peer realm's relay (e.g., a preshared-key IPsec tunnel to a specific named Martian relay endpoint). Unauthenticated packets carrying remote-realm source addresses SHOULD be dropped at the terrestrial-realm ingress under this Informative guidance; the normative MUST-drop rule for scope-boundary violations in general lives in §14.11 and covers this case by the scope-containment principle.

Design note: These allocations are deliberately generous. Reserving entire /8 blocks for bodies that currently have zero networked devices is intentional. The cost of reserving address space now is zero. The cost of not having it when Mars has a permanent settlement is a protocol revision.

3.13. 3.13. Documentation and Testing Range (254.0.0.0/8 in r.r.r.r)

The r.r.r.r range 254.0.0.0/8 is reserved for documentation, examples, and testing. This is analogous to IPv4's 192.0.2.0/24 (TEST-NET-1), 198.51.100.0/24 (TEST-NET-2), and 203.0.113.0/24 (TEST-NET-3), but provides a substantially larger space suitable for complex multi-AS test scenarios.

Addresses within 254.0.0.0/8 MUST NOT appear on any production network and MUST be filtered at all border routers. Example:

254.0.0.1::192.168.1.1 (test ASN 1, host 192.168.1.1)
254.0.0.2:0.0.0.3::10.0.0.1 (test ASN 2, zone 3, host 10.0.0.1)

3.14. 3.14. Stateless Address Autoconfiguration (SLAAC-ASIP)

ASIP supports stateless address autoconfiguration, derived from IPv6 SLAAC [RFC4862] but adapted for the 32-bit host field.

3.14.1. 3.14.1. Overview

SLAAC-ASIP allows a device to configure a routable ASIP address without contacting a DHCP server. The device learns the network prefix (r.r.r.r:z.z.z.z:s.s.s.s, 96 bits) from Router Advertisements and generates the h.h.h.h host portion locally.

SLAAC-ASIP is OPTIONAL. Operators MAY require DHCP-ASIP-only addressing via a flag in Router Advertisements, identical to IPv6's M (Managed) flag. SLAAC-ASIP and DHCP-ASIP MAY coexist on the same network.

3.14.2. 3.14.2. Host Identifier Generation

The h.h.h.h field (32 bits) is generated by one of three methods, in order of preference:

Method 1: Stable Opaque Identifier (RECOMMENDED)

A stable, pseudorandom 32-bit host ID derived from a hash of the interface identifier, network prefix, a secret key, and a counter [RFC7217 adapted]:

h.h.h.h = truncate_32(SHA-256(prefix || interface_id || secret_key || counter))

This produces a stable address per network (the same device on the same network always gets the same address) without revealing hardware identity. This is the ASIP equivalent of IPv6's RFC 7217 stable privacy addresses.

Method 2: Temporary Random Identifier (Privacy Extension)

A cryptographically random 32-bit value, regenerated periodically (RECOMMENDED interval: 24 hours). Analogous to IPv6 temporary addresses [RFC8981]. Used for outbound connections where tracking prevention is desired.

Method 3: MAC-Derived Identifier (NOT RECOMMENDED)

The lower 24 bits of the MAC address (OUI stripped) with the upper 8 bits set to a hash of the full 48-bit MAC:

h.h.h.h = hash_8(MAC[0:6]) || MAC[3:6]

This method is NOT RECOMMENDED because it exposes hardware identity, analogous to the privacy concerns with IPv6's original EUI-64 scheme. It is defined for constrained devices that lack a cryptographic random number generator.

Reserved host values: h.h.h.h = 0.0.0.0 (network identifier) and h.h.h.h = 255.255.255.255 (subnet broadcast) are reserved and MUST NOT be generated by SLAAC-ASIP. If Method 1's hash output or Method 2's random draw produces one of these reserved values, the implementation MUST discard the candidate and regenerate (for Method 1, by incrementing the counter and re-hashing; for Method 2, by drawing a new random value). The probability of this event is 2/2^32 per draw and does not meaningfully affect the DAD retry budget.

3.14.3. 3.14.3. Duplicate Address Detection (DAD)

Before using a SLAAC-ASIP-generated address, a device MUST perform Duplicate Address Detection using ICMP-ASIP Neighbor Solicitation messages (analogous to IPv6 DAD [RFC4862]). The device sends an ICMP-ASIP Neighbor Solicitation to the tentative address using its link-local address as the source. If a response is received, the address is in use and the device MUST generate a new h.h.h.h (increment the counter for Method 1, regenerate for Method 2).

Collision bound and DHCP-ASIP fallback: The 32-bit host field is smaller than IPv6's 64-bit interface identifier. Birthday-paradox analysis gives the following collision probabilities on a single subnet with N concurrent SLAAC-generated host IDs drawn uniformly at random from 2^32: P(collision) ~= 1 - exp(-N^2 / 2^33). At N=10,000: P ~= 1.16%. At N=50,000: P ~= 25.3%. At N=65,536 (2^16): P ~= 39.3%. The 50%-collision point is N ~= sqrt(2 ln 2 * 2^32) ~= 77,162 hosts. Modern datacenter leaf subnets routinely approach the tens of thousands. Accordingly:

* On subnets expected to carry fewer than 10,000 concurrent devices, SLAAC-ASIP MAY be used as the sole addressing method.

* On subnets that may exceed 10,000 concurrent devices, Router Advertisements MUST set the M (Managed) flag and devices MUST obtain addresses via DHCP-ASIP. SLAAC-ASIP MAY still be used for the link-local bootstrap address.

* DAD retries MUST be bounded. After 3 collisions, a device MUST fall back to DHCP-ASIP or report failure to the operator rather than re-rolling indefinitely.

Operators deploying very large flat L2 domains (>10k hosts) SHOULD segment those domains into smaller subnets rather than relying on 32-bit SLAAC uniqueness. The trade-off narrative motivating the 10k threshold, including the birthday-paradox curve and the relation to IPv6's 64-bit interface identifier, is summarized in § 18.7.

3.14.4. 3.14.4. SLAAC-ASIP Sequence

1. Interface comes up 2. Generate link-local address: 169.254.0.0::h.h.h.h (SLAAC-ASIP Method 1 or 2) 3. Perform DAD on link-local address 4. Send Router Solicitation from link-local address 5. Receive Router Advertisement containing prefix (r:z:s) 6. Generate host ID h.h.h.h for the advertised prefix 7. Perform DAD on the full address r:z:s:h 8. Address is ready for use

The entire sequence completes in under 2 seconds on a typical wired network, comparable to IPv6 SLAAC.

3.15. 3.15. Address Usage Summary

r.r.r.r Range	Usage	Scope
0.0.0.0 (with z=0, s=0)	IPv4-mapped	Translated/encapsulated per Section 12
0.0.0.1 - 96.255.255.255	Terrestrial ASN Unicast	Global (eBGP-ASIP)
97.0.0.0/8	Near-Earth Infrastructure	Realm (reserved)
98.0.0.0/8	Cislunar / Lunar	Realm (reserved)
99.0.0.0/8	Martian	Realm (reserved)
100.0.0.0/8	RINE peering links	Link (IXP only)
101.0.0.0 - 126.255.255.255	Terrestrial ASN Unicast	Global (eBGP-ASIP)
127.0.0.0/8	Internal zone prefixes	Organization
128.0.0.0 - 169.253.255.255	Terrestrial ASN Unicast	Global (eBGP-ASIP)

169.254.0.0/16	Link-local	Link only
169.255.0.0 - 239.255.255.255	Terrestrial ASN Unicast	Global (eBGP-ASIP)
240.0.0.0/8	Mesh / Ad-hoc	Mesh domain
241.0.0.0 - 249.255.255.255	Future celestial bodies	Reserved
250.0.0.0/8	DTN Relay	Realm bridge
251.0.0.0 - 253.255.255.255	Reserved	Future use
254.0.0.0/8	Documentation / Testing	Never routed
255.0.0.0 - 255.255.254.255	Reserved	Future use; MUST NOT be routed
255.255.255.0 - 255.255.255.254	Cross-ASN Multicast	Per-value assignment; see Sections 4.2 and 10
255.255.255.255	Broadcast	L2 broadcast only

Table 2

Most devices on most networks use either internal zone addressing (127.x.x.x) or their organization’s public ASN. Link-local (169.254.x.x) is always present on every interface as a bootstrap channel.

4. 4. Address Classes

r.r.r.r Value	Class	Description
0.0.0.0 (z=0, s=0)	IPv4-Mapped	Legacy IPv4 destination; translated/encapsulated per Section 12
0.0.0.1 - 96.255.255.255	Terrestrial ASN Unicast	Earth internet, public routing via eBGP-ASIP
97.0.0.0/8	Near-Earth Realm	LEO/GEO satellite infrastructure

		(reserved)
98.0.0.0/8	Cislunar Realm	Earth-Moon infrastructure (reserved)
99.0.0.0/8	Martian Realm	Mars infrastructure (reserved)
100.0.0.0/8	RINE Peering	AS-to-AS link addressing only
101.0.0.0 - 126.255.255.255	Terrestrial ASN Unicast	Earth internet, public routing via eBGP-ASIP
127.0.0.0/8	Internal Zone	Organization-scoped, never routed externally
128.0.0.0 - 169.253.255.255	Terrestrial ASN Unicast	Earth internet, public routing via eBGP-ASIP
169.254.0.0/16	Link-Local	Single link only, never forwarded by routers
169.255.0.0 - 239.255.255.255	Terrestrial ASN Unicast	Earth internet, public routing via eBGP-ASIP
240.0.0.0/8	Mesh / Ad-Hoc	Self-organizing networks, mesh-scoped
241.0.0.0 - 249.255.255.255	Interplanetary (Reserved)	Future celestial body allocations
250.0.0.0/8	DTN Relay	Delay-tolerant relay nodes (reserved)
251.0.0.0 - 253.255.255.255	Reserved	Future use
254.0.0.0/8	Documentation / Testing	MUST NOT appear on production networks
255.0.0.0 - 255.255.254.255	Reserved	Future use; MUST NOT be routed
255.255.255.0 - 255.255.255.254	Cross-ASN Multicast	Per-value assignment; see Section 4.2 and Section 10

255.255.255.255	Broadcast	L2 broadcast, MUST NOT be routed

Table 3

4.1. 4.1. Scope Hierarchy

ASIP defines a formal scope hierarchy for addresses, drawn from IPv6's multicast scope model but applied universally:

Scope Level	Boundary	Example r.r.r.r Ranges
Link	Single physical/ virtual link	169.254.0.0/16 (link-local)
Mesh	Self-organizing network domain	240.0.0.0/8
Organization	AS boundary	127.0.0.0/8 (internal zones)
Terrestrial	Earth internet	0.0.0.1 - 96.x, 101.x - 126.x, 128.x - 239.x
Realm	Celestial body / region	97.x, 98.x, 99.x, 241.x - 249.x, 250.x
Universal	All realms	Not yet defined; requires inter-realm relay

Table 4

With the one exception called out immediately below for mesh, each strictly-nested scope level in the table contains the ones below it. A link-local address is never visible at organization scope. An organization-internal address is never visible at terrestrial scope. A terrestrial address is never visible in another realm without explicit relay. This containment is enforced by routers at each boundary.

Mesh scope (240.0.0.0/8) is orthogonal rather than strictly nested. A mesh domain is not contained within a terrestrial AS and is not a non-terrestrial realm; it is a self-organizing scope reachable only via the mesh-gateway bridge of § 3.11. For scope-boundary enforcement

(§ 14.11) it is treated as a peer of "terrestrial" and "realm": mesh addresses MUST NOT leave the mesh domain natively, and terrestrial or realm addresses MUST NOT enter a mesh domain natively; a mesh gateway that performs address translation is the only sanctioned bridge.

IPv4-mapped addresses (r=z=s=0, Section 3.3) are the single explicit exception: they represent a legacy IPv4 destination that has no realm, are handled by the translators defined in Section 12, and MUST NOT be forwarded natively into any non-terrestrial realm.

4.2. 4.2. Multicast Prefix Assignments (r.r.r.r = 255.255.255.x)

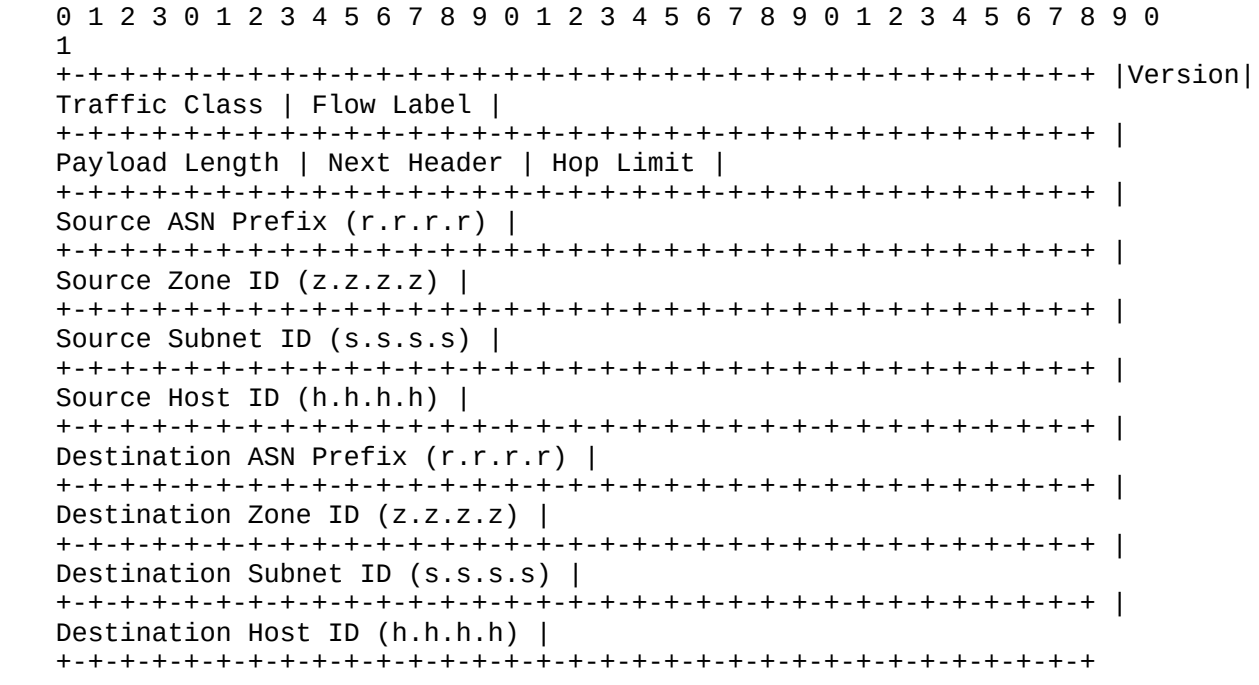
r.r.r.r	Assignment
255.255.255.0	General cross-ASN multicast (includes all well-known protocol groups; see § 10.4)
255.255.255.1	RESERVED for a future per-protocol OSPF-ASIP group encoding; MUST NOT be used by current implementations (see § 10.4)
255.255.255.2	RESERVED for a future per-protocol eBGP-ASIP peer-discovery group encoding; MUST NOT be used by current implementations (see § 10.4)
255.255.255.3	RESERVED for a future per-protocol IS-IS-ASIP group encoding; MUST NOT be used by current implementations (see § 10.4)
255.255.255.4 - 255.255.255.127	Available for IANA assignment
255.255.255.128 - 255.255.255.254	Reserved, future use
255.255.255.255	Broadcast

Table 5

5. 5. ASIP Packet Header

5.1. 5.1. Header Format

ASIP uses IP version number 8 in the Version field. The base header is fixed-length (40 bytes), drawing from IPv6's design decision to eliminate variable-length options from the base header. Additional functionality (fragmentation, routing options, hop-by-hop processing) is provided via extension headers identified by the Next Header field, identical in mechanism to IPv6 [RFC8200].



Field definitions:

Field	Bits	Description
Version	4	Always 8 for ASIP
Traffic Class	8	Differentiated services (DSCP + ECN), identical to IPv6 Traffic Class and backward compatible with IPv4 ToS/DSCP
Flow Label	20	Per-flow identifier for QoS and ECMP hashing, identical semantics to IPv6 [RFC6437]. Enables routers to identify packets belonging to the same flow without deep packet inspection. Source-assigned; zero if unused.
Payload Length	16	Length of payload after the base header, in octets. Does not include the 40-byte base header itself. (IPv6 semantics.)
Next Header	8	Identifies the type of the immediately following header. Values are drawn from the IPv6 Next Header registry for protocols that are identical under ASIP (6 = TCP, 17 = UDP, 43 = Routing, 44 = Fragment, 59 = No Next Header, 60 = Destination Options, 50/51 = ESP/AH). ASIP-specific protocols receive distinct assignments; see Section 5.3 and Section 15.5. ICMP-ASIP = 143 (to be assigned by IANA).
Hop Limit	8	Decremented by 1 at each forwarding node. Packet is discarded when it reaches 0. (Identical to IPv6 Hop Limit and IPv4 TTL in function.)
Source Address	128	Full 128-bit ASIP source address (r.r.r.r : z.z.z.z : s.s.s.s : h.h.h.h)
Destination Address	128	Full 128-bit ASIP destination address

Table 6

Total base header size: 40 bytes. This is identical to IPv6 and 20 bytes larger than IPv4’s minimum header.

5.2. 5.2. Design Rationale: What Was Dropped from IPv4

The ASIP header deliberately omits several IPv4 fields:

* *IHL (Internet Header Length):* Unnecessary. The ASIP base header is fixed at 40 bytes. Variable-length options are handled via extension headers, not inline.

* *Identification, Flags, Fragment Offset:* Moved to a Fragment Extension Header (Next Header = 44), identical to IPv6's approach. Fragmentation is performed only by the source, not by intermediate routers. This eliminates the performance penalty of router-path fragmentation and the reassembly-based attacks that plagued IPv4. *Fragmentation happens at the inner ASIP layer only, not at any outer encapsulation (§12.3).* ASIP-to-IPv4 encapsulators MUST set the outer IPv4 DF bit (§12.8) so that outer fragmentation is suppressed; any too-big condition is reported upstream and fragmentation, if needed, is performed by the ASIP source. This eliminates the ambiguity of where fragmentation occurs in a translate-plus-encapsulate path.

* *Header Checksum:* Dropped, identical to IPv6's rationale. Link-layer (L2) and transport-layer (L4) checksums provide integrity verification. The per-hop header checksum in IPv4 was a performance bottleneck on high-speed routers (recomputed at every hop due to TTL decrement) with no security benefit.

5.3. 5.3. Extension Headers

ASIP uses the same extension header mechanism as IPv6. Extension headers are chained via the Next Header field and processed in order. The following extension headers are defined:

Next Header Value	Extension Header	Description
0	Hop-by-Hop Options	Options examined by every node along the path
43	Routing	Source routing and related functions
44	Fragment	Fragmentation and reassembly
50	ESP	Encapsulating Security Payload (IPsec)
51	AH	Authentication Header (IPsec)
59	No Next Header	Nothing follows this header
60	Destination Options	Options examined only by the destination
143	ICMP-ASIP	ASIP ICMP messages (to be assigned by IANA; distinct from IPv6 NH=58 to prevent dispatcher ambiguity across header-stripping middleboxes)
(reserved, no assignment requested)	Mobility	Explicitly out of scope for this document. No Next Header value is requested in §15 for ASIP mobility. A future specification MAY define Mobile ASIP and request an IANA assignment at that time; this document does not reserve or request any code point for it.

Table 7

By reusing IPv6’s extension header numbering and semantics, ASIP benefits from the extensive implementation experience and security analysis already applied to IPv6 extension header processing [RFC8200, RFC7045].

IPsec (AH/ESP) over ASIP. AH (NH=51) and ESP (NH=50) inherit IPv6's semantics [RFC4301, RFC4302, RFC4303] verbatim, except that the AH Integrity Check Value computation covers the full 128-bit ASIP source and destination addresses as written on the wire. Because the r.r.r.r locator is allowed to be rewritten at AS boundaries (§8.3.2), AH's immutable-field assumption does not hold for r.r.r.r across a locator rewrite; AH-protected traffic MUST NOT traverse a boundary that rewrites r.r.r.r, or the ICV check will fail at the far end. ESP is unaffected because it does not sign the outer IP header. Operators who require locator rewriting on AH-protected flows MUST use tunnel-mode ESP or re-establish the SA after the rewriting boundary.

Enforcement at rewriting nodes. The "MUST NOT traverse" rule above is a prevention rule, not a suggestion. A router that would otherwise rewrite r.r.r.r per §8.3.2 MUST first parse the extension-header chain of the inbound packet; if any AH header (NH=51) is present anywhere in the chain, the router MUST NOT rewrite and MUST drop the packet, emitting ICMP-ASIP Destination Unreachable (Type=1, Code=10 "Administratively Prohibited by AH-rewrite policy"; exact Code per the §15.5 registry). Silently rewriting an AH-protected packet produces an ICV-failure symptom at the destination that is indistinguishable from an on-path attack and MUST be avoided. A middlebox that strips AH before rewrite to avoid this check is a downgrade attacker; endpoints that require AH integrity SHOULD additionally verify via an ESP-protected channel or out-of-band attestation that no rewrite-capable intermediary is on the path. This is a classical downgrade-attack concern inherited from IPsec and is not unique to ASIP.

Ordering with ingress filtering. At a border router that both implements §14.1 ingress filtering and performs §8.3.2 locator rewrite, the ingress filter MUST run on the received packet as it arrived on the wire, before any rewrite. The rewrite MUST occur after the ingress filter has validated the original source r.r.r.r against the peer's authorized-origin set. Performing rewrite before ingress filtering would either (i) validate the rewritten value against the wrong peer set, or (ii) fail to validate at all; both outcomes break BCP 38 semantics.

5.4. 5.4. Flow Label Usage

The 20-bit Flow Label enables per-flow identification without transport-layer inspection. Semantics are identical to IPv6 Flow Label [RFC6437]: the value is opaque to the network, source-assigned, and used as an input to hash-based path-selection functions.

* *ECMP hashing:* Routers MAY use the flow label (combined with source and destination addresses) as input to ECMP hash functions, ensuring all packets of a flow traverse the same path. This is the primary intended use.

* *QoS classification:* Middleboxes MAY classify traffic by flow label as a hint only.

A source that does not use flow-based path selection MUST set the flow label to zero. Routers MUST NOT parse structure inside the flow label, MUST NOT make security decisions based on its value, and MUST NOT rewrite it in the forwarding path.

Relationship to external path-selection metrics. Any path-selection metric external to this specification (e.g., the OPTIONAL Cost Factor of § 17 / draft-asip-cf-00, or a future operator-defined metric) MUST NOT be encoded into the flow label and MUST NOT cause a router to parse structure inside it. The flow label remains opaque regardless of whether such a metric is deployed. Where an external metric drives path selection, the flow label MAY be used as an ECMP hash input that pins a flow to whichever path that metric selected — this is an emergent property of hash-based forwarding, not a structural use of the flow label.

5.5. 5.5. Socket API Compatibility

Existing IPv4 applications use the standard BSD socket API with AF_INET and sockaddr_in. The ASIP compatibility layer intercepts these calls transparently. The application requires zero ASIP awareness.

New applications MAY use AF_ASIP with sockaddr_asip:

c struct sockaddr_asip { sa_family_t asip_family; /* AF_ASIP */ in_port_t asip_port; /* Port number */ uint32_t asip_asn; /* r.r.r.r ASN locator */ uint32_t asip_zone; /* z.z.z.z Zone ID */ uint32_t asip_subnet; /* s.s.s.s Subnet ID */ uint32_t asip_host; /* h.h.h.h Host ID */ uint32_t asip_flowlabel; /* Flow label (20 bits) */ uint32_t asip_scope_id; /* Interface index for link-local and mesh scopes; zero for all other scopes */ };

All numeric fields except asip_family are in network byte order. The flow label occupies the low-order 20 bits of asip_flowlabel; the upper 12 bits MUST be zero on transmission and MUST be ignored on reception. The asip_scope_id field is in host byte order and carries the interface index disambiguating which link a link-local (169.254.0.0/16 in r.r.r.r, § 3.10) or mesh (240.0.0.0/8 in r.r.r.r, § 3.11) address refers to; it serves the same role as

sockaddr_in6.sin6_scope_id [RFC4291]. asip_scope_id MUST be zero for non-link-local, non-mesh addresses and MUST be ignored by the stack for those scopes. An application operating on link-local or mesh addresses across multiple interfaces MUST set asip_scope_id to the correct interface index; a bind or sendto with asip_scope_id=0 on a multi-interface host for a link-local address is implementation-defined and MAY fail with EINVAL.

5.6. 5.6. IPv4/ASIP Coexistence Processing

When an ASIP-aware router receives a packet with Version = 4, it processes it as standard IPv4. When an ASIP-aware router receives a packet with Version = 8, it processes the 128-bit source and destination addresses as specified in Sections 3 and 8. If the destination has r=z=s=0 (IPv4-mapped form, Section 3.3), the packet's endpoint is an IPv4 host and the router forwards it toward an IPv4/ASIP translator at the AS boundary (Section 12.2) rather than attempting to dispatch it on h.h.h.h alone; the source may be either a native ASIP address (the common case, § 12.7.1) or itself IPv4-mapped (the IPv4-to-IPv4-over-ASIP relay case, rare). A Version=8 packet is always a Version=8 frame on the wire; the r=z=s=0 condition on either address is an addressing form, not an alternate forwarding mode.

Routers, hosts, and middleboxes that are not ASIP-aware see a Version=8 frame as an unrecognized IP version and drop it. This is not a quiet no-op; it is a hard break. Every forwarding element in the path between two ASIP-aware endpoints must either be ASIP-aware or be bypassed by the transition mechanisms in Section 12. This cost is the reason ASIP is specified as a new address family with a transition mechanism rather than as a wire-level IPv4 superset.

6. 6. Notation and Address Compression

ASIP defines a colon-separated field notation as the canonical text representation, with compression rules that eliminate consecutive all-zero fields. All compliant implementations MUST accept both compressed and uncompressed forms and MUST be able to produce compressed output.

6.1. 6.1. Canonical Notation

The canonical ASIP text representation uses colons to separate the four 32-bit fields. Each field is written in standard dotted decimal:

r.r.r.r:z.z.z.z:s.s.s.s:h.h.h.h

Examples (uncompressed): 0.0.59.65:0.0.0.1:0.0.0.10:192.0.2.1 (ASN 15169, Zone 1, Subnet 10, Host) 0.0.251.240:0.0.0.0:0.0.0.0:10.0.0.1 (ASN 64496, flat network) 0.0.0.0:0.0.0.0:0.0.0.0:8.8.8.8 (IPv4 compatible: 8.8.8.8) 127.1.0.0:0.0.0.5:0.0.1.0:10.0.0.1 (Internal zone 1, zone 5, subnet 256)

6.2. 6.2. ASN Integer Notation

The r.r.r.r field MAY be written as a plain decimal ASN integer instead of dotted decimal. This is the RECOMMENDED human-facing format for public addresses:

{ASN}:z.z.z.z:s.s.s.s:h.h.h.h

Examples (uncompressed): 15169:0.0.0.1:0.0.0.10:192.0.2.1 (same as first example above) 64496:0.0.0.0:0.0.0.0:10.0.0.1 (same as second example above) 0:0.0.0.0:0.0.0.0:8.8.8.8 (IPv4 compatible)

All compliant implementations MUST accept both dotted-decimal and integer forms for the ASN field. Implementations SHOULD produce ASN integer notation in user-facing output.

6.3. 6.3. Zero-Field Compression (:: Notation)

One or more consecutive fields whose value is 0.0.0.0 MAY be replaced by ::, analogous to IPv6 zero compression [RFC5952]. Since ASIP has exactly four fields, the compression rules are simpler and less ambiguous than IPv6.

Rules:

1. :: replaces one or more consecutive all-zero fields (a field is all-zero when its 32-bit value is 0x00000000, i.e., 0.0.0.0).

2. :: MUST NOT appear more than once in a single address.

3. When multiple runs of consecutive zero fields exist and are of equal length, :: SHOULD replace the leftmost run.

4. :: MAY replace a single all-zero field. (Unlike RFC 5952’s recommendation for IPv6, single-field compression is permitted in ASIP because each field is 32 bits and eliminating even one saves substantial notation length.)

5. The host field (h.h.h.h) MUST always be written explicitly. :: MUST NOT compress the rightmost field. (Rationale: the host field is the most operationally significant for debugging, and omitting it creates dangerous ambiguity with IPv4-compatible addresses.)

6. The ASN field (r.r.r.r) written as integer 0 is valid. Both 0::10.0.0.1 and ::10.0.0.1 expand to ASN=0, zone=0, subnet=0, host=10.0.0.1 and are semantically identical; the leading 0 before :: is an explicit-field form that the expansion algorithm (§6.5) collapses to the same wire value as the ::-only form. Implementations MUST treat the two as equivalent.

Compression table (all possible positions for four fields where h.h.h.h is never compressed):

Fields Present	Notation	Meaning
All four explicit	A:Z:S:H	No compression
Zone is zero	A::S:H	z = 0.0.0.0
Subnet is zero	A:Z::H	s = 0.0.0.0
Zone and Subnet are zero	A::H	z = 0.0.0.0, s = 0.0.0.0
ASN and Zone are zero	::S:H	r = 0, z = 0.0.0.0
ASN, Zone, Subnet are zero	::H	r = 0, z = 0.0.0.0, s = 0.0.0.0

Table 8

Note: Because :: can only appear once, an address where only the ASN and Subnet are zero (but the Zone is non-zero) cannot compress both. In this case, write the ASN explicitly as 0:Z:0.0.0.0:H or compress only the longer run. In practice this pattern is rare.

6.4. 6.4. Compression Examples

‘’’ UNCOMPRESSED COMPRESSED ----------- ----------

Public host, flat network: 64496:0.0.0.0:0.0.0.0:192.0.2.1
64496::192.0.2.1

Public host, zone 1, subnet 10: 15169:0.0.0.1:0.0.0.10:192.0.2.1
15169:0.0.0.1:0.0.0.10:192.0.2.1 (no zero fields, no compression)

IPv4 compatible: 0:0.0.0.0:0.0.0.0:8.8.8.8 ::8.8.8.8

Internal zone, flat: 127.1.0.0:0.0.0.0:0.0.0.0:10.0.0.1
127.1.0.0::10.0.0.1

Internal zone, with zone and subnet:
127.1.0.0:0.0.0.5:0.0.1.0:10.0.0.1 127.1.0.0:0.0.0.5:0.0.1.0:10.0.0.1
(no zero fields, no compression)

Internal zone, zone set, subnet zero:
127.2.0.0:0.0.0.3:0.0.0.0:172.16.0.1 127.2.0.0:0.0.0.3::172.16.0.1

RINE peering link: 100.0.0.1:0.0.0.0:0.0.0.0:10.255.0.1
100.0.0.1::10.255.0.1

Loopback (all zeros except host): 0:0.0.0.0:0.0.0.0:127.0.0.1
::127.0.0.1

Multicast: 255.255.255.0:0.0.0.0:0.0.0.0:224.0.0.1
255.255.255.0::224.0.0.1 ‘’’

6.5. 6.5. Expansion Algorithm

To expand a compressed address, an implementation:

1. Splits the string on :: to obtain a left part and a right part.

2. Splits each part on : to count explicit fields.

3. Inserts enough 0.0.0.0 fields between left and right to bring the total to exactly four (ASN, Zone, Subnet, Host).

4. If the ASN field is a plain integer, converts it to a 32-bit unsigned value in network byte order.

5. If no :: is present, all four fields must be explicit.

This algorithm is deterministic and requires no lookahead.

6.6. 6.6. Flat Dotted Notation (Wire/Debug Format)

For diagnostic output, packet dumps, and wire-level representations, implementations MAY use flat dotted notation with all 16 octets dot-separated:

0.0.59.65.0.0.0.1.0.0.0.10.192.0.2.1

This format is unambiguous but not human-friendly for routine use. It MUST be accepted by all parsers. Implementations SHOULD NOT produce this format in user-facing output; use canonical or compressed notation instead.

6.7. 6.7. CIDR Prefix Notation

ASIP CIDR notation appends a prefix length to the compressed address, separated by /:

15169::/32 (ASN 15169, all addresses within that ASN)
15169:0.0.0.1::/64 (ASN 15169, Zone 1, all subnets/hosts)
127.1.0.0::/32 (Internal zone 1, all addresses) ::0.0.0.0/0 (Default route)

The prefix length counts bits from the most significant bit of the r.r.r.r field. Common prefix boundaries:

Prefix Length	Boundary	Meaning
/32	After ASN	All addresses within one AS
/64	After Zone	All subnets/hosts within one zone
/96	After Subnet	All hosts within one subnet
/128	Full address	Single host
/0	None	Default route

Table 9

6.8. 6.8. Design Note on Notation Choices

The choice to retain dotted decimal and use colons only as field separators is deliberate. IPv6's hexadecimal colon notation (2001:0db8::1) is compact but unfamiliar to many operators and a common source of transcription errors. ASIP notation is immediately readable by anyone who can read an IPv4 address. The :: compression symbol is borrowed from IPv6 because it is the one piece of IPv6 notation that operators already understand and that solves a real readability problem.

The requirement that h.h.h.h always be explicit is a deliberate safety constraint. In IPv4 troubleshooting, the host address is the most commonly referenced field. Allowing it to be compressed away would make compressed ASIP addresses actively hostile to operational debugging. The upper fields (ASN, Zone, Subnet) are structural and rarely change during a troubleshooting session; the host field changes constantly.

This is an explicit trade-off: ASIP addresses are longer to write than IPv6 addresses but shorter to understand, and common compressed forms (like ::8.8.8.8 for IPv4-compatible addresses or 64496::192.0.2.1 for flat-network hosts) are concise enough for routine use.

7. 7. DNS Integration

7.1. 7.1. A-ASIP Record Type

A new DNS resource record type, A-ASIP, is defined for ASIP addresses.

* *Type:* A-ASIP (IANA assignment pending)

* *Wire format:* 128-bit ASIP address in network byte order

* *Presentation format:* Canonical notation with compression (Section 6)

7.2. 7.2. Resolution Behavior

An ASIP-aware resolver SHOULD request both A and A-ASIP records. Resolution behavior by address-family hint:

* *AF_INET (IPv4-only):* Return A records only. A-ASIP records MUST NOT be returned through an AF_INET query. Legacy applications that call gethostbyname() or getaddrinfo() with AF_INET receive the A record unchanged.

* *AF_ASIP (ASIP-only):* Return A-ASIP records preferentially. When no A-ASIP record exists but an A record does, synthesize an IPv4-mapped A-ASIP (r=z=s=0, h = the A record, § 3.3) and return it; the stack routes such traffic via § 12 translation or encapsulation. The stack does not synthesize an ASN locator for an IPv4-only destination.

* *AF_UNSPEC (both):* Return both A and A-ASIP records if present, with A-ASIP records ordered first. Applications using Happy Eyeballs [RFC8305] or equivalent MUST treat the returned set as an

ordered preference list and MAY attempt connections in parallel. The resolver MUST NOT silently drop A records when A-ASIP records exist; doing so breaks dual-availability semantics.

When only an A record is available for an AF_ASIP or AF_UNSPEC query, the synthesis described above applies. An A-ASIP record MUST NOT be returned to an AF_INET application.

RFC 1918 addresses MUST NOT be published as A-ASIP records in public DNS.

7.3. 7.3. Dual Record Example

ns1.example.com. IN A 192.0.2.1 ns1.example.com. IN A-ASIP 15169:0.0.0.1:0.0.0.10:192.0.2.1

8. 8. Routing Protocol Behavior

8.1. 8.1. Multi-Tier Routing Table

ASIP routing uses a tiered lookup model derived directly from the address structure:

Tier	Field	Scope	Function
1	r.r.r.r	Global (inter-AS)	Routes packet to correct AS border
2	z.z.z.z	AS-internal (inter-zone)	Routes packet to correct zone
3	s.s.s.s	Zone-internal (intra-zone)	Routes packet to correct subnet
4	h.h.h.h	Subnet-local	Delivers to host (identical to IPv4)

Table 10

When r.r.r.r = 0.0.0.0, z.z.z.z = 0.0.0.0, and s.s.s.s = 0.0.0.0, the packet is IPv4-mapped (§3.3) and the lookup yields a route toward an IPv4/ASIP translator (§12.2); the Version=8 frame remains a Version=8 frame on the wire until it reaches the translator (§5.6). The tier structure above describes the lookup ordering for ASIP-native packets, not an alternate wire format.

In practice, most backbone routers perform a single Tier 1 lookup (32-bit ASN match) and forward. The remaining tiers are resolved only within the destination AS. In the common single-origin case, the global routing table contains one entry per ASN; multihoming, anycast, and traffic-engineering more-specifics documented in § 8.3.1 add entries on top of that baseline, and § 8.3's "honest bound" paragraph states the realistic table size.

8.2. 8.2. Routing Protocol Extensions

ASIP extends existing routing protocols rather than replacing them:

Protocol	ASIP Extension	Scope	Status
BGP4	eBGP-ASIP	Inter-AS (Tier 1)	REQUIRED for internet-facing routers
iBGP	iBGP-ASIP	Inter-zone (Tier 2)	RECOMMENDED for multi-zone AS
OSPFv2	OSPF-ASIP	Intra-zone (Tier 3)	RECOMMENDED for intra-zone routing
IS-IS	IS-IS-ASIP	Intra-AS (Tier 2/3)	OPTIONAL alternative IGP
Static	Unchanged	All tiers	Always available

Table 11

Note on existing protocols: ASIP does not deprecate any existing routing protocol. Operators MAY continue to use RIPv2, EIGRP, or any other protocol within their AS for IPv4-compatible traffic. The "ASIP" extensions add awareness of the 128-bit address space. They do not remove any existing functionality. Operators MAY additionally deploy the OPTIONAL Cost Factor metric (§ 17 companion draft), which has no effect on conformance to this specification.

8.3. 8.3. eBGP-ASIP

eBGP-ASIP extends BGP4 [RFC4271] with:

* 128-bit address family support (AFI/SAFI for ASIP; see § 15.2).

* WHOIS-ASIP route validation integration (Section 8.4, REQUIRED where the structural routing-table bound is relied upon).

* An OPTIONAL Cost Factor path attribute specified in a separate companion document (§ 17); eBGP-ASIP conformance does not require CF.

eBGP-ASIP operates alongside BGP4 on the same peering session via multi-protocol extensions [RFC4760]. eBGP-ASIP does not replace BGP4 for IPv4 prefixes; the two AFIs coexist.

Prefix granularity and table bound. The nominal inter-AS aggregation boundary is /32 in the r.r.r.r field (one prefix per ASN locator). Prefixes more specific than /32 in r.r.r.r (i.e., subdividing a single ASN's address space at the inter-AS boundary) SHOULD NOT be advertised across AS boundaries; they MAY be advertised when operationally necessary to support the use cases in Section 8.3.1. Aggregation less specific than /32 (covering multiple ASN locators under a single prefix) MAY be performed down to /16 for internal backbone use but MUST NOT be used to obscure origin at peering boundaries.

Honest bound. The routing-table entry count under ASIP is not equal to the ASN count. Approximately 74,000 ASNs exist globally (2026); any bound that simply counts allocated or announced ASNs (e.g., a "~175,000-entry" figure derived from allocated-but-unannounced ASN counts) would conflate "ASN count" with "routing-table entry count" and ignore the reasons operators deaggregate today. The actual entry count under ASIP will be larger than the ASN count whenever:

* An AS is multihomed and injects separate more-specifics to steer traffic per upstream (Section 8.3.1);

* An AS uses anycast across multiple origin locations (Section 11.1);

* An AS splits traffic engineering announcements for failover;

* An AS has legitimately acquired address space from a transferred ASN and continues to announce the old locator.

Forbidding these use cases does not remove them; operators would respond by splitting into additional ASNs, which grows the locator count rather than shrinking the table. The realistic structural bound is "substantially smaller than today's 1M+ BGP4 table in the common case, with a long tail of more-specifics driven by the traffic-engineering use cases that ASIP cannot eliminate." The specification records the trade-off in that form because a single-number headline figure would omit the deaggregation drivers that an implementer must size for.

8.3.1. 8.3.1. Multihoming, Anycast, and ASN Transfer

Because r.r.r.r is a routing locator rather than a host identity, the following cases work without re-coupling identifier and locator:

* *Multihoming.* A host reachable through two upstream ASes is assigned one ASIP address per upstream (e.g., ASN_A:z:s:h and ASN_B:z:s:h). DNS-ASIP publishes both A-ASIP records. Applications use happy-eyeballs-style selection between them. The host's stable identity is the z:s:h triple under its own administrative scope; r.r.r.r varies with reachability. Server-side operational impact of a client presenting multiple addresses is addressed in § 18.5b.

* *Cross-ASN anycast.* An anycast service advertised from multiple ASNs is assigned one address per origin ASN. Each origin ASN advertises its own ASN_n::h.h.h.h prefix. Clients use DNS-ASIP to obtain one or more candidate anycast addresses and the routing system steers each packet to the nearest instance via normal BGP path selection. This is strictly equal in capability to current cross-ASN anycast (Cloudflare, Google) and is not artificially constrained to a single ASN.

* *ASN transfer.* When an ASN transfers between organizations, the addresses using that ASN as their r.r.r.r locator also transfer. Operators MAY renumber affected hosts by rewriting r.r.r.r at ingress (Section 8.3.2); the z:s:h portion remains stable and no application reconfiguration is required for hosts addressed through stable identifiers rather than bare ASIP literals.

8.3.2. 8.3.2. Locator Rewriting

An AS border router MAY rewrite the r.r.r.r field of packets on ingress or egress when the rewrite is a pure locator change (z, s, h unchanged) and the mapping is published in WHOIS-ASIP (Section 8.4) or equivalent. Locator rewriting is used for:

* Stateless IPv4/ASIP translation at AS boundaries (Section 12.2);

* ASN transfer renumbering (Section 8.3.1);

* Privacy rewriting of internal z.z.z.z and s.s.s.s as described in Section 14.8.

Locator rewriting MUST NOT alter the transport-layer payload or checksums in ways inconsistent with the translation rules of Section 12. A router performing locator rewrite MUST enforce the AH-rewrite prohibition defined in § 5.3 ("Enforcement at rewriting

nodes"): any inbound packet carrying an AH extension header (NH=51) MUST be dropped with the ICMP-ASIP administratively-prohibited code rather than silently rewritten. A router that rewrites AH-protected traffic produces an ICV-failure symptom indistinguishable from an on-path attack; the prohibition is normative in both § 5.3 (from the IPsec-interaction perspective) and here (from the rewrite-implementation perspective), and the two statements are semantically identical.

8.4. 8.4. WHOIS-ASIP Route Validation

eBGP-ASIP routers validate received route advertisements against WHOIS-ASIP, a route-ownership registry. A route advertisement that cannot be validated against a registered WHOIS-ASIP entry SHOULD NOT be installed in the routing table.

WHOIS-ASIP is functionally similar to RPKI [RFC6480] but is scoped to ASIP's locator model: the registry maps each r.r.r.r value (and any legitimate more-specifics permitted by Section 8.3.1) to the ASN authorized to originate it. Because the common case is one prefix per ASN, the registry is substantially smaller than RPKI's prefix space; the long tail of multihoming, anycast, and TE more-specifics is registered explicitly.

Normative status. Route-ownership validation is a core guarantee of this document (Section 2.4 R5). Accordingly:

* Deployments that rely on the structural routing-table bound or on the "one route = one authorized origin" property MUST deploy WHOIS-ASIP validation on all eBGP-ASIP sessions.

* Deployments willing to accept BGP4-equivalent route hygiene (no route validation, hijacks and leaks possible) MAY defer WHOIS-ASIP validation during transition. Such deployments MUST NOT claim the structural routing-table bound.

The structural routing-table guarantee does not exist without route-ownership validation: a deployment that accepts any eBGP-ASIP advertisement from any peer inherits the same hijack, leak, and bogon-injection pathology as BGP4, and no "one route = one authorized origin" property can be claimed. The two tiers above bind R5's REQUIRED/RECOMMENDED status to the property the deployment actually relies on.

Chicken-and-egg. As with RPKI, WHOIS-ASIP adoption provides meaningful protection only when a critical mass of originating ASes publish records. Early adopters SHOULD publish WHOIS-ASIP records even when few peers validate them, and SHOULD default to log-only for received advertisements that fail validation until peering partners converge on enforcement. See Section 18.6.

Bootstrap transport. WHOIS-ASIP services MUST be reachable over IPv4 during the transition period so that new ASNs can bootstrap their ASIP routing locators without a pre-existing ASIP path. After ASIP native reachability is established, WHOIS-ASIP services MAY be reached over ASIP. This avoids the chicken-and-egg failure mode where two new ASIP-native ASNs cannot discover each other's r.r.r.r locator because the registry they depend on is itself only reachable via ASIP.

Response authentication. WHOIS-ASIP responses MUST be authenticated: the structural routing-table bound and the "one route = one authorized origin" property both depend on a validator being able to trust that the WHOIS-ASIP record it retrieved reflects the registry's actual entry. A companion specification (draft-asip-whois-00, out of scope for this document) is expected to define the signature and trust-anchor model; an RPKI-style X.509 hierarchy [RFC6480] is the working assumption. Until that specification is published, eBGP-ASIP deployments relying on WHOIS-ASIP validation MUST obtain records over an authenticated channel (e.g., TLS with HPKP-equivalent pinning, or signed zone file distribution) and MUST NOT accept unauthenticated WHOIS-ASIP responses as a basis for route acceptance. Without authentication, an adversary in the IPv4 bootstrap path could inject forged locator-to-ASN bindings into a validator's cache and defeat the entire route-validation chain.

Service discovery. WHOIS-ASIP service endpoints MUST be published via DNS using the SRV record _whois-asip._tcp.{registry-domain} and MUST be reachable at the IPv4 address(es) to which that SRV resolves. Registry operators MUST publish their SRV records under at least one well-known registry domain delegated by IANA (see § 15.10). This resolves the bootstrap-discovery gap: a new ASN needs only a working IPv4 DNS resolver to locate its regional WHOIS-ASIP service; no pre-existing ASIP path, hard-coded endpoint, or out-of-band configuration is required.

8.5. 8.5. iBGP-ASIP and OSPF-ASIP

iBGP-ASIP distributes external routes within an AS. OSPF-ASIP extends OSPFv2 with 128-bit address support. Both are specified fully in companion documents. iBGP-ASIP and OSPF-ASIP conformance does not require CF awareness; where an operator deploys the OPTIONAL CF attribute of §17's companion draft, iBGP-ASIP MAY carry it transparently across the AS. Neither iBGP-ASIP nor OSPF-ASIP defines CF behavior in this document.

Neither is mandatory. An operator running a single-zone AS may use OSPF-ASIP alone. An operator running a flat network may use static routes alone with ASIP addressing. The protocol does not prescribe internal network architecture.

9. 9. ICMP-ASIP

ICMP-ASIP extends ICMP [RFC792] to support 128-bit addresses, incorporating the Neighbor Discovery functions from IPv6's ICMPv6 [RFC4443]. ICMP-ASIP is identified by Next Header value 143 (to be assigned by IANA; see Section 15.5). Both ICMPv4 and ICMP-ASIP MUST be supported simultaneously by ASIP implementations.

Why not NH=58. ICMP-ASIP MUST NOT reuse IPv6's ICMPv6 Next-Header value (58). Reuse would create a dispatcher ambiguity: a middlebox that strips or normalizes the outer ASIP header while preserving the NH chain would deliver the inner payload to an ICMPv6 handler that expects 128-bit ICMPv6 semantics, not ICMP-ASIP. Because the message-type registries, Neighbor Discovery option codes, and checksum pseudo-headers differ between ICMPv6 and ICMP-ASIP, the collision would silently produce wrong behavior rather than a clean error. A distinct Next-Header value (requested value 143, subject to IANA assignment) eliminates the ambiguity.

9.1. 9.1. Core Messages

ICMP-ASIP carries full 128-bit addresses in Echo Request/Reply, Destination Unreachable, Time Exceeded, Redirect, Packet Too Big, and Parameter Problem messages. Path MTU Discovery is extended for the ASIP header.

The initial ICMP-ASIP Type values used normatively elsewhere in this document (§12.7, §12.8) are listed here for implementer convenience; the full registry is established per §15.5 and follows ICMPv6 [RFC4443] numbering unless otherwise noted:

Type	Name	Normative use in this spec
1	Destination Unreachable	§12.7.3, §12.7.7, §5.3 (AH-rewrite drop)
2	Packet Too Big	§12.7.4, §12.8 (PMTUD)
3	Time Exceeded	Hop Limit = 0 drop
4	Parameter Problem	Header malformed
128	Echo Request	§12.7.2
129	Echo Reply	§12.7.2
133-137	Router/Neighbor Discovery (RS, RA, NS, NA, Redirect)	§9.2, §3.14

Table 12

Values not listed above follow the ICMPv6 registry [RFC4443] until §15.5's IANA action establishes a distinct ICMP-ASIP registry.

9.2. 9.2. Neighbor Discovery

ASIP adopts IPv6's Neighbor Discovery Protocol (NDP) [RFC4861] in full, adapted for the four-field address format. NDP replaces ARP for ASIP-native address resolution, providing:

* *Router Solicitation / Router Advertisement (RS/RA):* Used by hosts to discover routers and obtain network prefixes for SLAAC-ASIP (Section 3.14). RAs carry the 96-bit prefix (r:z:s), the M flag (managed addressing via DHCP-ASIP), the A flag (autonomous SLAAC-ASIP permitted), and prefix lifetime.

* *Neighbor Solicitation / Neighbor Advertisement (NS/NA):* Used for address resolution (replacing ARP), Duplicate Address Detection (DAD), and neighbor unreachability detection. All NS/NA messages use link-local source addresses (169.254.0.0::h.h.h.h).

* *Redirect:* Used by routers to inform hosts of a better first-hop router for a destination.

NDP messages are sent to solicited-node multicast addresses derived from the target's h.h.h.h field. The solicited-node group is formed from the low 24 bits of h.h.h.h, mirroring IPv6's 24-bit L2-multicast form (IEEE 802.3 33:33:xx:xx:xx:xx for Ethernet). Collisions on the high 8 bits of h.h.h.h cause two ASIP hosts to share a solicited-node group; the NA reply carries the full 32-bit h.h.h.h and the requester discriminates at L3.

Collision-rate difference versus IPv6. Because ASIP's h.h.h.h is 32 bits rather than IPv6's 64-bit interface identifier, only 8 bits of the host field are elided into the solicited-node group (versus 40 bits elided in IPv6). Both protocols use 24 bits of the host/IID to form the solicited-node group, so both partition into 2^24 groups; under uniformly-random SLAAC host IDs, the expected number of other hosts sharing a given host's group is (N-1) / 2^24 for both ASIP and IPv6, where N is the subnet's host count. At N=1000 this is approximately 6× 10^-5 on both protocols; at N=10,000 approximately 6× 10^-4; at N=1,000,000 approximately 0.06. The solicited-node collision rate is therefore not materially worse on ASIP than on IPv6 at realistic subnet sizes. The 32-bit host field's distinct collision concern is the full-identifier birthday collision (two hosts drawing the same h.h.h.h value); that concern is covered quantitatively in § 3.14.3 and addressed by § 3.14.3's DHCP-ASIP-at-10k threshold. The solicited-node-group concern and the full-identifier concern are separate collision mechanisms operating at different layers (L3 group membership vs. L3 uniqueness) and MUST NOT be conflated: the group-level collision is negligible at realistic subnet sizes, while the identifier-level collision becomes material at N~10,000 and above.

9.3. 9.3. ARP-ASIP Compatibility

For backward compatibility on mixed IPv4/ASIP L2 segments, an ARP-style resolution mechanism for 128-bit addresses (provisionally "ARP-ASIP") MAY be supported as a fallback. On pure ASIP links, NDP (§ 9.2) is the specified mechanism and MUST be used. An ASIP link on which neither NDP nor a specified ARP-ASIP is available is a misconfiguration; this document does not define ARP-ASIP wire format, and a companion specification would be required before ARP-ASIP is interoperable. ASIP-only deployments that depend on ARP-ASIP before such a companion document is published are NOT INTEROPERABLE and SHOULD NOT be built; operators needing L2 resolution on ASIP MUST deploy NDP per § 9.2.

10. 10. Multicast

10.1. 10.1. Scoped Multicast Model

ASIP adopts IPv6's scoped multicast architecture [RFC4291, RFC7346], adapted to the four-field address structure. Multicast scope determines how far a multicast packet is forwarded and is encoded directly in the address.

ASIP multicast uses the low-order 4 bits of the z.z.z.z field to encode scope (values 0-15), mirroring the IPv6 multicast scope field [RFC7346]. The remaining 28 bits of z.z.z.z MUST be zero for scoped multicast (r.r.r.r = 255.255.255.0/24). Packets received with non-zero values in those 28 bits MUST be dropped at ingress (not merely ignored by receivers), because forwarding a packet whose reserved bits are non-zero would create a covert-channel and scope-bypass surface that a future assignment of those bits may not resolve safely.

z.z.z.z Low Nibble	Scope	Boundary
0x0	Reserved	MUST NOT be used
0x1	Interface-local	Never leaves the originating interface
0x2	Link-local	Single physical/virtual link
0x3	Reserved	MUST NOT be used
0x4	Admin-local	Administratively defined (site, building)
0x5	Site-local	Single site / campus
0x6-0x7	Unassigned	Reserved for future scope levels
0x8	Organization-local	Single AS / organization
0x9-0xD	Unassigned	Reserved for future scope levels
0xE	Global	Entire terrestrial internet
0xF	Reserved	MUST NOT be used

Table 13

Routers MUST NOT forward a scoped multicast packet beyond its indicated scope boundary. Routers MUST drop scoped multicast packets whose z.z.z.z low nibble is a Reserved or Unassigned value, and SHOULD log the event. This is a hard enforcement, not a suggestion.

10.2. 10.2. Intra-ASN Multicast (IPv4 Compatible)

Packets with r.r.r.r = 0.0.0.0 (all upper fields zero) and h.h.h.h in the IPv4 multicast range (224.0.0.0/4) are treated as intra-ASN multicast and MUST NOT be forwarded beyond the local AS boundary. This preserves full compatibility with existing IPv4 multicast deployments.

10.3. 10.3. Cross-ASN Multicast

Cross-ASN multicast uses r.r.r.r values in the range 255.255.255.0 through 255.255.255.254. The single r.r.r.r value 255.255.255.255 is reserved for broadcast (§11.2) and MUST NOT be used for multicast. The z.z.z.z field encodes the scope. The s.s.s.s and h.h.h.h fields identify the multicast group, providing a 64-bit group space within each scope level and each r.r.r.r assignment.

255.255.255.0 : {scope} : {group-hi} : {group-lo}

10.4. 10.4. Well-Known Multicast Groups

The addresses below are the canonical well-known groups. They use the general cross-ASN multicast r.r.r.r prefix 255.255.255.0 of §4.2 with IPv4-familiar group values in h.h.h.h, so that operators who recognize 224.0.0.5 and 224.0.0.6 from IPv4 OSPF see the same values here. The per-protocol r.r.r.r values 255.255.255.1 (OSPF-ASIP), 255.255.255.2 (eBGP-ASIP), and 255.255.255.3 (IS-IS-ASIP) in the §4.2 table are reserved for future per-protocol use (e.g., a protocol-specific group-ID encoding that does not overlap the IPv4 224/4 values); an implementation MUST join the 255.255.255.0-prefixed groups below to participate in the listed protocols, and a 255.255.255.1/2/3 address MUST NOT be used in place of the addresses below until a companion specification defines those prefixes' group-ID encoding.

255.255.255.0:0.0.0.2::224.0.0.1 All ASIP routers (link-local) 255.255.255.0:0.0.0.2::224.0.0.2 All ASIP Zone Servers (link-local) 255.255.255.0:0.0.0.2::224.0.0.5 OSPF-ASIP all routers (link-local) 255.255.255.0:0.0.0.2::224.0.0.6 OSPF-ASIP designated routers (link-local) 255.255.255.0:0.0.0.8::224.0.0.10 iBGP-ASIP peer discovery (organization-local) 255.255.255.0:0.0.0.14::224.0.0.1 All ASIP routers (global)

10.5. 10.5. Multicast Listener Discovery

ASIP uses MLD-ASIP, adapted from IPv6's MLDv2 [RFC3810], for multicast group membership management. MLD-ASIP operates over ICMP-ASIP and uses link-local source addresses (169.254.0.0::h.h.h.h) for all messages, identical to IPv6's requirement that MLD use link-local sources. This ensures multicast membership management functions before any global addressing is configured.

10.6. 10.6. Composition of Multicast Scope with Realm and Mesh Scope

§10.1 defines a multicast scope nibble (link, subnet, admin, site, org, global, …) drawn from IPv6 precedent. §4.1 defines an orthogonal scope hierarchy including realm (§3.12) and mesh (§3.11). Neither section states how the two compose. The rules below are NORMATIVE for any router enforcing either scope:

1. *Scope = Interface-local or Link-local (0x1, 0x2).* Never cross any boundary, including realm and mesh boundaries. Confined to the originating link.

2. *Scope = Admin-local, Site-local, Organization-local (0x4, 0x5, 0x8).* Confined to the originating AS. MUST NOT cross an AS boundary, MUST NOT cross a realm boundary, and MUST NOT cross a mesh gateway. "Organization-local scope in realm A" does not imply reachability in realm B even if the two realms share an operator.

3. *Scope = Global (0xE).* Confined to the originating realm's terrestrial scope (or, if originated in a non-terrestrial realm, to that realm's internal global scope). "Global" is *realm-local-global*, not universal. A multicast packet with scope=0xE originating in realm A MUST NOT be forwarded into realm B by any relay. No "universal-across-realms" multicast scope is currently defined; the Universal row in §4.1's scope table is reserved for a future inter-realm-multicast specification and has no encoding today.

4. *Realm-boundary ingress rule.* A router that receives a multicast packet on an interface it classifies as being at a realm boundary (i.e., the packet's source r.r.r.r lies in a different realm from the router's local realm) MUST drop the packet regardless of the packet's multicast scope nibble. Realm boundaries are hard drops for multicast; no multicast scope permits realm crossing in this document.

5. *Mesh-boundary rule.* Multicast packets originated in a mesh domain (§3.11) MUST NOT be forwarded out of the mesh by a mesh gateway, regardless of scope nibble. Conversely, non-mesh multicast MUST NOT be forwarded into a mesh domain by a mesh gateway. If a use case requires a mesh to receive a copy of a terrestrial multicast, the gateway MUST terminate the terrestrial multicast and originate a new mesh-scoped multicast (application-level relay), not forward natively.

6. *Link-local unicast scope (§3.10) and multicast.* Link-local multicast (nibble = 0x2) is the multicast analog of link-local unicast. Both are bounded to the single link and are not affected by realm or mesh boundaries because they never reach a boundary node's forwarding plane.

Interaction matrix (informative summary):

Multicast scope	Crosses link?	Crosses AS?	Crosses realm?	Crosses mesh gateway?
0x0 Reserved	Drop	Drop	Drop	Drop
0x1 Interface-local	No	No	No	No
0x2 Link-local	Intra-link only	No	No	No
0x3 Reserved	Drop	Drop	Drop	Drop
0x4 Admin-local	Yes (admin scope)	No	No	No
0x5 Site-local	Yes (site)	No	No	No
0x6-0x7 Unassigned	Drop	Drop	Drop	Drop
0x8 Organization-local	Yes (AS)	No	No	No
0x9-0xD Unassigned	Drop	Drop	Drop	Drop
0xE Global	Yes	Yes	*No*	No
0xF Reserved	Drop	Drop	Drop	Drop

Table 14

Rule 4 is the load-bearing rule: no multicast packet crosses a realm boundary natively in this document, regardless of scope nibble. §10.1 multicast scope and §4.1 realm scope are orthogonal dimensions, and a packet may satisfy one scope test while violating the other; Rule 4 composes the two so that realm-boundary enforcement dominates every multicast scope nibble and no combination admits a realm crossing.

11. 11. Anycast and Broadcast

11.1. 11.1. Anycast

Anycast is a routing property, not a distinct address class. Two anycast cases are supported:

* *Same-ASN anycast.* Multiple origin sites within the same ASN advertise the same ASIP prefix. The inter-AS routing system steers each packet to the nearest origin by normal BGP path selection. This is the straightforward extension of IPv4 anycast.

* *Cross-ASN anycast.* An anycast service reachable from multiple different ASNs (the current Cloudflare/Google model) is assigned multiple ASIP addresses, one per origin ASN, and is published under a single DNS-ASIP name with all its A-ASIP records. Clients use normal DNS-ASIP resolution to obtain the candidate set; the routing system steers each flow to the instance best-reached through its chosen destination address. This preserves the operational capability that BGP4 anycast delivers today.

The cross-ASN case is the reason Section 3.1 defines r.r.r.r as a routing locator rather than a host identity. An anycast service host is not "owned" by one ASN; it is reachable through several, each with its own locator.

Ordering and selection among anycast instances uses standard BGP path-selection criteria [RFC4271]. If an operator separately deploys the OPTIONAL Cost Factor metric (§17 / draft-asip-cf-00), CF MAY additionally influence anycast instance selection per that companion draft; CF-based anycast selection is not required by this document.

11.2. 11.2. Broadcast

The r.r.r.r value 255.255.255.255 is permanently reserved for broadcast and maps to the Layer 2 broadcast address. Broadcast packets MUST NOT be routed beyond the local network segment.

Note: As with IPv6, ASIP RECOMMENDS using multicast with a scope no broader than required instead of broadcast for most use cases. Broadcast remains defined for IPv4 compatibility. Link-layer resolution on ASIP-only segments uses NDP (§9.2); any ARP-style fallback (§9.3) is undefined in this document and MUST NOT be assumed interoperable.

12. 12. Compatibility and Transition

12.1. 12.1. Deployment Model

ASIP is a new address family on the wire. An ASIP-aware end host runs a network stack that originates Version=8 frames for ASIP destinations. For IPv4-mapped destinations (Section 3.3) the stack emits a Version=8 frame whose r=z=s=0 source and destination are handled by a translator (Section 12.2); where no translator is reachable, the host MAY fall back to emitting a native Version=4 frame on the same interface. That fallback is a dual-stack kernel data path by any reasonable definition, and this document acknowledges it as such. The single-stack property claimed by ASIP is narrower and scoped explicitly: applications see one address family (AF_ASIP), and sockets above the kernel are not duplicated per version.

"Single-stack" in this document therefore refers to the socket API surface exposed to applications, not to the kernel’s forwarding behavior. During transition, an ASIP-aware host’s kernel will process both IPv4 and ASIP frames, and forwarding elements in the path will be IPv4-only, ASIP-aware, or both. That is unavoidable.

Networks that have not deployed ASIP continue to operate as pure IPv4 networks. They require no modification. They are also not reachable by ASIP-native traffic except through the translation and encapsulation mechanisms below.

12.2. 12.2. IPv4/ASIP Stateless Translation at AS Boundaries

ASIP-aware AS border routers MAY act as stateless translators between Version=8 frames carrying IPv4-mapped addresses (r=z=s=0) and Version=4 frames. The translation rules follow SIIT [RFC7915] adapted for the ASIP address format:

* An ASIP-to-IPv4 translation strips the r=z=s=0 address prefix, emitting an IPv4 packet with h.h.h.h as the IPv4 source/ destination.

* An IPv4-to-ASIP translation prepends r=z=s=0 on ingress, producing a Version=8 frame whose addresses are IPv4-mapped.

* Transport checksums are recomputed per SIIT.

Origin advertisement requirement (normative; see § 14.1). Operators deploying this section’s stateless translation MUST originate the 0.0.0.0::/96 IPv4-mapped prefix via eBGP-ASIP from the translator-owning ASN, so that the § 14.1 ingress filter at every downstream peer

treats r=z=s=0 sourced traffic as originating from a legitimate peer rather than as spoofed. Without this advertisement, reverse-direction reply traffic (§12.7.1, common case) will be dropped at the first non-translator-declared peering boundary. This is a deployment requirement, not a wire-format requirement. This requirement binds only to operators deploying §12.2 stateless translation; operators deploying only §12.3 encapsulation (ASIP-to-IPv4 tunneling) MUST NOT originate 0.0.0.0::/96, because they do not serve that prefix and originating an unserved prefix is exactly the hijack pattern WHOIS-ASIP (§8.4) rejects.

Translation is stateless in the data-plane forward direction: any Version=8 frame with r=z=s=0 can be emitted as Version=4 using only information from the packet itself. ICMP error translation and reverse-path locator reconstruction require a stable mapping between the IPv4 literal and the originating ASIP source's r.r.r.r value; see §12.7.1 and §12.7.3 for the mechanisms. That mapping MAY be per-flow state or an administratively configured static mapping; the forward-direction translation itself remains stateless.

h.h.h.h addressability constraint. An ASIP source whose traffic is stateless-translated to IPv4 emits IPv4 packets with Src = S8.h.h.h.h (§12.7.1). Reply traffic from the IPv4 destination therefore targets that h.h.h.h value as an IPv4 address on the public IPv4 internet. For stateless §12.2 translation to work without a NAT44-style rewrite, S8.h.h.h.h MUST be a globally routable IPv4 address owned by the translator operator and reachable via the translator on the return path. ASIP sources whose h.h.h.h falls in RFC 1918 private space, CGN shared space (100.64.0.0/10), or any other non-publicly-routable IPv4 range MUST either (i) be assigned a publicly-routable h.h.h.h from a translator-owned IPv4 pool for egress, with the translator performing h.h.h.h rewrite (stateful NAT44, as in §12.6 CGNAT behavior), or (ii) use §12.3 encapsulation instead of §12.2 translation. The constraint is load-bearing because the IPv4 return path has no ASIP locator information and can only reach the h.h.h.h literal; a non-publicly-routable h.h.h.h simply produces unreachable return traffic.

Path-asymmetric traffic and multihoming. The h.h.h.h-ownership rule above also constrains return-path routing for multihomed ASIP clients (§8.3.1). Because reply traffic is addressed to the IPv4 h.h.h.h that the forward-direction translator owns, and IPv4-side routing steers that reply back to the translator that announced the h.h.h.h, return traffic inherently lands at the same translator that created the forward-direction mapping. A stateless translator at a different ASN's border that receives an inbound IPv4 packet whose Dst is not in its own translator-owned pool MUST silently drop the packet and MUST NOT synthesize a forged ASIP source by guessing a reverse mapping.

When an ASIP client deliberately switches outbound traffic from ASN_A's translator to ASN_B's translator (e.g., uplink failover), the client's ASIP source address changes (new h.h.h.h from ASN_B's pool, per the rewrite or multihoming model); existing IPv4-side sessions bound to the ASN_A h.h.h.h do not survive the switch and MUST be reopened. Session-continuity mitigations are covered in § 18.5b. The translator's behavior is strictly "drop what you don't own, do not fabricate": fabricating a reverse mapping would inject forged ASIP source addresses into a downstream path and violate § 14.1 ingress filtering at every subsequent hop.

12.3. 12.3. ASIP-to-IPv4 Encapsulation Across IPv4 Transit

Where two ASIP-aware sites are separated by IPv4-only transit, they MAY tunnel Version=8 frames inside Version=4 packets. The normative encapsulation is GENEVE over UDP [RFC8926] using IANA-assigned port 6081, with ASIP frames carried as the inner protocol. An IANA-assigned ASIP-to-IPv4 protocol type is requested for GENEVE's protocol field (Section 15.9).

Rationale. GENEVE/UDP is the encapsulation of choice for modern virtual networks (VXLAN, NVGRE are similar but older): it is stateless, adds 16 bytes of UDP+GENEVE overhead on top of the outer IPv4 header, is supported by commodity silicon, does not introduce per-flow TCP congestion-control interactions between unrelated tenants, and does not require a handshake before the first packet. HTTPS (TCP/TLS) tunneling is NOT RECOMMENDED for general transit use: TCP-over-TCP introduces nested congestion control and head-of-line blocking across unrelated inner flows; TLS handshakes add round-trips on every tunnel re-establishment; per-packet bytes above the inner L4 payload are ASIP(40)+IPv4(20)+TCP(20)+TLS(~29) = 109 rather than the 76 of GENEVE/UDP (ASIP(40)+IPv4(20)+UDP(8)+GENEVE(8)). HTTPS tunneling is retained only as a last-resort traversal option (see below).

Operators MAY use alternative encapsulation methods:

* *GRE or IP-in-IP* for high-throughput backbone links where UDP NAT traversal is not a concern.

* *WireGuard or IPsec/ESP* where per-tunnel authentication and encryption are required.

* *HTTPS as a last-resort traversal option* for environments that block UDP and non-HTTPS TCP. HTTPS tunneling is NOT RECOMMENDED for general transit use because of the overhead and congestion-control pathology noted above. It is MAY, not SHOULD.

The 8TO4-ENDPOINT eBGP-ASIP attribute carries the IPv4 tunnel endpoint address automatically. Tunnel MTU and Path MTU Discovery behave as defined in [RFC8926]; operators MUST account for the encapsulation overhead when sizing MTU (see § 12.8).

GENEVE option TLVs. This document defines no ASIP-specific GENEVE option classes. ASIP-to-IPv4 encapsulation MUST use GENEVE with no options in the baseline defined here; the 8-byte GENEVE header in § 12.8's overhead budget assumes zero options. Implementations that receive ASIP-to-IPv4 GENEVE frames carrying unknown option TLVs MUST process them per [RFC8926] § 3.5 (critical-bit handling). Future ASIP deployments that wish to define option TLVs (e.g., for per-flow telemetry, inter-AS traffic-engineering hints) MUST do so in a companion specification and MUST request an option class from IANA's GENEVE registry; inline invention of TLVs by operators breaks interop and is NOT RECOMMENDED.

Tunnel endpoint authentication and reflection/amplification. UDP-based encapsulation on a well-known port is a reflection/amplification surface if decapsulators accept packets from any outer source. ASIP-to-IPv4 decapsulators MUST maintain a configured allowlist of peer tunnel endpoint IPv4 addresses (learned via the 8TO4-ENDPOINT eBGP-ASIP attribute or administratively configured) and MUST silently drop inbound GENEVE/UDP/6081 packets whose outer IPv4 source is not on that allowlist. Decapsulators MUST NOT emit ICMPv4 error messages in response to unauthorized outer sources (doing so converts the decapsulator into a reflector). Decapsulators SHOULD rate-limit per outer source. Where operators require cryptographic endpoint authentication rather than source-address allowlisting (e.g., when the outer IPv4 path is subject to spoofing), WireGuard or IPsec/ESP MUST be used instead of bare GENEVE/UDP as noted above.

12.4. 12.4. Transition Value

Unlike IPv6, which offered no incremental benefit to early adopters until a critical mass was reached, ASIP provides localized value at each adoption phase. These are not "phases of global deployment"; each benefits the individual adopter.

* *ISP backbone adopter:* Reduced routing-table churn on the subset of the table that is ASIP-native, assuming WHOIS-ASIP is deployed (Section 8.4). Early adopters will not see a full 1M-entry table reduction because they inherit a transit environment dominated by IPv4 prefixes, and the BGP4 table their routers carry is unaffected by a parallel ASIP table; the per-adopter benefit grows with ASIP adoption across their peering set.

* *Cloud providers:* ASN+Zone addressing eliminates VPC address overlap within and across cooperating ASIP-aware tenants; cross-region routing inside a cloud's own ASIP deployment is simpler than the IPv4 overlay-on-overlay model.

* *Enterprise:* Internal zone addressing (127.x.x.x) enables multi-region private networks without external address coordination or RFC 1918 conflict management. This benefit is realized entirely inside the enterprise and requires no upstream ASIP deployment.

* *Consumer ISPs:* For customers whose reachable destinations are ASIP-native, CGNAT can be bypassed. For customers whose destinations remain IPv4, CGNAT is unchanged.

Each role interoperates with non-upgraded peers via Section 12.2 translation and Section 12.3 encapsulation. There is no dependency between roles. Organizations adopt at their own pace, but the claim "each adopter benefits fully regardless of anyone else's deployment" is too strong; the honest claim is that each adopter obtains localized value and pays localized cost.

12.5. 12.5. IPv6 Coexistence

ASIP and IPv6 MAY coexist on the same infrastructure; neither supersedes the other. An operator already running IPv6 MAY continue to run it and adopt ASIP as a parallel address family, as a replacement at the AS boundary, or not at all. ASIP does not claim to be technically superior to IPv6; it offers an alternative transition path whose costs and benefits differ. See Section 18.5 for a candid discussion.

12.6. 12.6. CGNAT Behavior

CGNAT devices that are not ASIP-aware are IPv4-only middleboxes and operate unchanged on IPv4 traffic. They cannot process Version=8 frames. An ASIP-aware CGNAT MAY translate the h.h.h.h field of IPv4-mapped ASIP addresses in the same way it translates IPv4 source addresses today; it MUST NOT modify r.r.r.r, z.z.z.z, or s.s.s.s during translation. CGNAT operators without a publicly registered ASN MUST NOT originate non-mapped ASIP traffic onto the public internet; they MAY use internal zone prefixes (Section 3.5) for subscriber addressing and MUST translate at the AS boundary.

12.7. 12.7. Stateless Translation Reference: Packet-Level Walkthrough

This subsection is NORMATIVE for implementations that perform IPv4/ ASIP stateless translation per § 12.2. It supplements § 12.2 by specifying the byte-level behavior for the cases that translator implementations most often get wrong: ICMP errors carrying truncated inner headers, fragmented inner payloads, the IPv4 DF bit, and ICMPv4 Fragmentation Needed. Byte values are hexadecimal; offsets are zero-based.

Notation. ASIP source S8 = ASN 0.0.59.65, zone 0, subnet 0, host 192.0.2.1 (compressed: 15169::192.0.2.1). IPv4 destination D4 = 198.51.100.1. IPv4-mapped ASIP destination D8m = ::198.51.100.1. All checksums shown are illustrative placeholders denoted cksum; implementations MUST compute them per [RFC7915] (transport) and per the respective protocol (ICMP).

12.7.1. 12.7.1. Simple TCP Data Packet (ASIP -> IPv4)

Inbound to translator: Version=8 frame, 40-byte ASIP header + 20-byte TCP header + 100-byte payload; S = 15169::192.0.2.1, D = ::198.51.100.1.

ASIP header fields on the wire (big-endian):

Offset Bytes Field 0x00 80 00 00 00 Version=8, TC=0, FlowLabel=0 0x04 00 78 06 40 PayloadLen=120, NH=6(TCP), HopLimit=64 0x08 00 00 3B 41 00 00 00 00 S.rrrr=0.0.59.65, S.zzzz=0.0.0.0 0x10 00 00 00 00 C0 00 02 01 S.ssss=0.0.0.0, S.hhhh=192.0.2.1 0x18 00 00 00 00 00 00 00 00 D.rrrr=0.0.0.0, D.zzzz=0.0.0.0 0x20 00 00 00 00 C6 33 64 01 D.ssss=0.0.0.0, D.hhhh=198.51.100.1 0x28 [TCP header 20 bytes] 0x3C [payload 100 bytes]

After translation, the translator emits Version=4 frame:

Offset Bytes Field 0x00 45 00 00 8C V=4, IHL=5, ToS=0, TotLen=140 0x04 00 00 00 00 ID=0 (see note), Flags=0, FragOff=0 0x08 40 06 CC CC TTL=64, Proto=6(TCP), HdrChecksum (CC CC = computed) 0x0C C0 00 02 01 Src=192.0.2.1 (h.h.h.h of S) 0x10 C6 33 64 01 Dst=198.51.100.1 (h.h.h.h of D) 0x14 [TCP header 20 bytes, checksum recomputed per RFC 7915] 0x28 [payload 100 bytes]

The r.r.r.r=0.0.59.65 source ASN locator is dropped because the egress IPv4 network has no field to carry it. Reply traffic from D4 arrives at the translator as Version=4 with Src=198.51.100.1, Dst=192.0.2.1, and is translated back to Version=8 with S8′ = ::198.51.100.1, D8′ = 15169::192.0.2.1. *The translator MUST maintain sufficient state (or an administratively configured reverse mapping)

to recover the original S8.r.r.r.r = 0.0.59.65 on the return path*; without it, the reply is translated with S8’.r.r.r.r = 0 (the IPv4-mapped form) and arrives at the original client under a different source address than the one it sent to, breaking any per-address state at the client. This state is implicit in the SIIT [RFC7915] model when one side is IPv4-mapped; it is stated explicitly here because the r.r.r.r locator has no IPv4 analog and a translator that omits reverse-path reconstruction will produce source-address mismatch on every reply.

IPv4 ID field: set to 0 when DF is off and the packet is not fragmented. When DF is set (see §12.7.4) or fragmentation is needed, the translator assigns a 16-bit ID per RFC 7915.

ASIP pseudo-header for transport checksums. TCP, UDP, and ICMP-ASIP checksums over ASIP are computed using the IPv6 pseudo-header format [RFC8200 §8.1] with the 128-bit ASIP Source and Destination Addresses substituted verbatim for the IPv6 address fields, the 32-bit upper-layer packet length, 24 zero bits, and the 8-bit upper-layer protocol identifier (Next Header). The pseudo-header is never transmitted; it is a checksum input only. Translator implementations performing IPv4<->ASIP translation MUST recompute the transport checksum using the IPv4 pseudo-header on the IPv4 side and the ASIP pseudo-header above on the ASIP side; incremental update [RFC1624] is permitted only when both pseudo-headers are fully known to the translator.

TCP options. TCP options (Timestamp, SACK-Permitted, MSS, Window Scale, etc.) are part of the TCP header, not the IP layer. They pass through IPv4<->ASIP translation byte-for-byte unchanged; only the TCP checksum is recomputed. The MSS option value (if present on a SYN) is end-host-visible: the translator MUST NOT rewrite it, but the ASIP and IPv4 sides’ PMTU discovery (§12.8, §12.7.4) jointly determine the actual usable segment size, and an MSS advertised for a larger IPv4-side MTU will be clamped by the normal PMTU path.

12.7.2. 12.7.2. ICMP-ASIP Echo Request -> ICMPv4 Echo Request

An ICMP-ASIP Echo Request (NH=143, Type=128) from 15169::192.0.2.1 to ::198.51.100.1 is translated to an ICMPv4 Echo Request (Proto=1, Type=8). The identifier and sequence fields pass through unchanged; the payload passes through unchanged; the ICMP checksum is recomputed. Return direction reverses the type mapping (ICMPv4 Type=0 -> ICMP-ASIP Type=129).

Return-path locator reconstruction. The ICMPv4 Echo Reply arrives at the translator with IPv4 Src = the original D4 and IPv4 Dst = the h.h.h.h that was emitted as the IPv4 source on the forward Echo Request. To deliver the reply back to the originating ASIP host with

the full S8 = r:z:s:h preserved, the translator MUST apply the same r.r.r.r reconstruction state or administrative-mapping rule specified in §12.7.3 (forward direction is stateless; reverse direction requires mapping). Because Echo flows are often short-lived (single exchange) and high-volume (traceroute, PMTUD probes), translators that maintain per-flow mapping state SHOULD use an Echo-specific short timeout (RECOMMENDED: 60 seconds from the last forward Echo Request) to bound state growth independently of longer TCP/UDP flow state. See §14.15.

12.7.3. 12.7.3. ICMPv4 Destination Unreachable Carrying a Truncated IPv4 Inner Header (-> ICMP-ASIP)

This is the translation case implementers most frequently get wrong. An IPv4 router upstream of the translator emits ICMPv4 Type=3 (Destination Unreachable), which includes as its payload the first 28 bytes of the offending IPv4 packet (20-byte IPv4 header + 8 bytes of the next protocol, per RFC 792 semantics and RFC 1812 minimum). When this error arrives at the translator with inner Src=192.0.2.1 (the IPv4 form of the ASIP originator), the translator MUST synthesize an ICMP-ASIP error whose inner payload is a reconstructed ASIP header, not a pass-through of the IPv4 header.

Inbound ICMPv4 (bytes after IPv4 outer header):

Offset Bytes Field 0x00 03 01 [cksum] 00 00 00 00 Type=3, Code=1(host unreach), Unused 0x08 [inner IPv4 hdr 20 bytes: Src=192.0.2.1, Dst=198.51.100.1, Proto=6] 0x1C [inner IPv4 first 8 bytes of TCP: SrcPort, DstPort, SeqNum]

Outbound ICMP-ASIP (NH=143):

Offset Bytes Field 0x00 01 01 [cksum] 00 00 00 00 Type=1(Dst Unreach), Code=1, Unused 0x08 [reconstructed inner ASIP hdr 40 bytes] Version=8, PayloadLen=copied from original, NH=6(TCP), HopLimit=copied, S.rrrr=0.0.59.65, S.zzzz=0, S.ssss=0, S.hhhh=192.0.2.1 D.rrrr=0, D.zzzz=0, D.ssss=0, D.hhhh=198.51.100.1 0x30 [copied first 8 bytes of TCP: SrcPort, DstPort, SeqNum]

The inner ASIP header’s S.rrrr field MUST be reconstructed from the translator’s per-session state or administrative mapping (the r.r.r.r=0.0.59.65 of the original ASIP source). If the translator lacks that state, it emits S.rrrr=0 (IPv4-mapped form) and the originating ASIP host’s ICMP error-correlation will fail silently for any packet where the host tracks error association by full 128-bit source. The normative consequence is that *stateless translation is stateless in the forward direction only*; ICMP error translation requires either (i) per-flow ingress state or (ii) an administratively stable source-locator mapping. Implementations MUST document which approach they use.

The ICMPv4-to-ICMP-ASIP Type/Code mapping follows RFC 7915 § 4 for the common cases (Host Unreachable, Net Unreachable, Protocol Unreachable, Port Unreachable, TTL Expired, Parameter Problem).

12.7.4. 12.7.4. ICMPv4 Type 3 Code 4 (Fragmentation Needed / DF Set) -> ICMP-ASIP Packet Too Big

When the IPv4-side MTU cannot accommodate a translated packet whose inner originator set the ASIP "atomic fragment" intent (analogous to IPv4 DF), the translator converts the ICMPv4 Type=3 Code=4 error into an ICMP-ASIP Packet Too Big message. Because ASIP-to-IPv4 translation shrinks the packet by 20 bytes (ASIP base header is 40 bytes; IPv4 minimum header is 20 bytes), an ASIP packet of size X translates to an IPv4 packet of size X-20. To fit at the IPv4 next-hop MTU M, the ASIP origin may send packets up to M+20 bytes. The reported MTU to the ASIP originator is therefore IPv4_next_hop_MTU + (ASIP_hdr - IPv4_hdr) = IPv4_next_hop_MTU + 20. This is consistent with RFC 7915 § 5.1 MTU handling and symmetric with the IPv4->ASIP direction rule in § 12.7.6 (which subtracts 20).

Worked example: IPv4-side link reports next-hop MTU = 1500. Translator emits ICMP-ASIP Packet Too Big with MTU = 1520. The ASIP originator caches MTU=1520 for the destination; a 1520-byte ASIP packet becomes a 1500-byte IPv4 packet after translation and fits exactly. Implementations MUST NOT report the unadjusted IPv4 MTU (doing so causes a 20-byte per-packet efficiency loss) and MUST NOT report IPv4_next_hop_MTU - 20 (doing so causes a 40-byte per-packet loss and diverges from RFC 7915).

Fragment-needed-on-fragmented-inner corner case. If the ICMPv4 Type=3 Code=4 error arrives as a response to a translated ASIP packet that was already a fragment (NH=44 fragment header present), the translator emits ICMP-ASIP Packet Too Big and the originator MUST re-fragment at the lower MTU at the source, not via intermediate-router fragmentation (§ 5.2: ASIP fragments only at the source). This preserves IPv6/ASIP fragmentation semantics.

12.7.5. 12.7.5. Fragmented Inner Payload (-> IPv4)

An ASIP packet containing a Fragment Extension Header (NH=44) translates to an IPv4 packet with Flags.MF and FragOff set from the fragment header. The 32-bit fragment Identification in the ASIP fragment header is truncated to 16 bits for the IPv4 ID field; the low-order 16 bits are used. This is a lossy truncation but matches RFC 7915 §5.1.1 behavior. *The translator MUST NOT perform reassembly*; it forwards fragments as fragments. If the IPv4-side link MTU cannot carry the translated fragment, ICMPv4 Type=3 Code=4 is returned upstream and handled per §12.7.4.

Fragment-ID collision attack surface. Two ASIP fragmented flows whose 32-bit fragment IDs differ only in the upper 16 bits collapse to the same 16-bit IPv4 ID after truncation. On the IPv4 side, a reassembler may mis-associate fragments from the two flows, producing a fragmentation-based injection or corruption surface analogous to the RFC 7915 §5.1.1 caveat. Translators SHOULD assign the IPv4 ID field by hashing (source address, destination address, fragment ID) rather than by simple truncation when per-flow state is available, to reduce collision probability; implementations that use simple low-16-bit truncation MUST rate-limit the number of distinct fragmented flows translated concurrently, or MUST refuse to translate fragments above a configured flow-count threshold.

IPv4 -> ASIP direction (fragments). An IPv4 packet with Flags.MF=1 or FragOff!=0 arriving at the translator is a fragment of a larger original datagram. The translator MUST perform per-fragment translation without reassembly: each IPv4 fragment becomes an ASIP packet carrying a Fragment Extension Header (NH=44) whose 32-bit Identification is the zero-extended IPv4 16-bit ID (upper 16 bits set to 0), whose M and FragOff fields mirror the IPv4 Flags.MF and FragOff respectively, and whose payload is the fragmented upper-layer bytes verbatim. Reassembly occurs only at the ASIP destination, never at the translator. Fragments arriving out of order are translated in arrival order and forwarded independently; the translator MUST NOT buffer fragments waiting for earlier parts. This preserves the "stateless in the forward direction" property of §12.2 and matches RFC 7915 §4.1 behavior.

12.7.6. 12.7.6. IPv4 DF Bit Handling (IPv4 -> ASIP)

When translating an IPv4 packet with DF=1 to ASIP, the translator:

* If the IPv4 packet fits in the ASIP-side MTU with added overhead, forwards it as a single ASIP packet (no Fragment Extension Header).

* If the IPv4 packet does not fit, responds with ICMPv4 Type=3 Code=4 (Fragmentation Needed) to the IPv4 source with next-hop MTU = ASIP_MTU - 20. The IPv4 source reduces its sending size.

When DF=0, the translator MAY fragment the IPv4 packet inline on the IPv4 side before translation, or MAY translate-then-fragment on the ASIP side with a Fragment Extension Header. Either is compliant; implementations SHOULD prefer DF=1-respecting behavior for all IPv4 traffic because it matches modern IPv4-side deployment practice (PMTUD relies on DF=1).

Header-size delta with fragment extension header. When the translate-then-fragment path adds an ASIP Fragment Extension Header (NH=44, 8 bytes), the IPv4->ASIP header delta is +28 bytes (ASIP 40-byte base + 8-byte frag ext, minus IPv4 20-byte header), not the +20 bytes used elsewhere in § 12.7. A translator that emits fragmented ASIP output from a non-fragmented IPv4 input MUST account for the additional 8 bytes of fragment-extension overhead when deciding whether an output fragment will fit its ASIP-side MTU; the PMTU-reported value on the ASIP side is reduced by 8 additional bytes relative to the § 12.7.4 +20 formula in this case. This is a translator-local accounting concern; the ASIP originator sees only the end-to-end PMTU and does not need separate knowledge of the translator’s fragment-ext decision.

12.7.7. 12.7.7. ICMP-ASIP Error Back to IPv4 Originator (Reverse Direction)

When an IPv4 originator (S4 = 203.0.113.7) sends to an IPv4-mapped ASIP destination that reaches an unreachable ASIP host via the translator, the far-side ASIP router emits an ICMP-ASIP error (e.g., Type=1 Destination Unreachable) whose inner reconstructed header shows S8 = ::203.0.113.7 (the translator’s forward-direction synthesis) and some non-zero destination. That ICMP-ASIP error transits back toward the translator. The translator MUST synthesize an ICMPv4 error whose inner header is a reconstructed IPv4 header matching the packet S4 originally sent: inner Src = 203.0.113.7, inner Dst = the IPv4 h.h.h.h that S4 originally used as the destination. The outer ICMPv4 source address is the translator’s IPv4 address (standard ICMP-originating-router semantics). Without this reverse synthesis, S4 receives an ICMP error referencing an IPv4 destination it never addressed, and IPv4-side error correlation fails. The translation uses the same state or administrative-mapping requirement as the forward ICMP path (§ 12.7.3): the translator must recover the original IPv4 Dst from the ICMP-ASIP inner header’s D.hhhh field.

12.7.8. 12.7.8. Port-Restricted Inner ICMP

Some firewalls drop ICMPv4 Destination Unreachable messages whose inner TCP/UDP header indicates a port the firewall considers policy-blocked. When the translator receives such an error, it has already committed to forwarding the error back to the ASIP originator; the translator MUST NOT apply secondary port filtering on the inner payload of an ICMP error unless explicitly configured to do so. Conversely, if the IPv4-side firewall drops the error, the ASIP originator receives no feedback and its PMTU discovery or error correlation stalls. This is a pre-existing operational pathology inherited from IPv4; it is not created by ASIP and is not resolved here. Operators SHOULD disable aggressive ICMP filtering on translator-adjacent firewalls, identical to the advice given for IPv4 PMTUD.

12.8. 12.8. Path MTU Discovery and Encapsulation Budget

This subsection is NORMATIVE. ASIP PMTUD is semantically identical to IPv6 PMTUD [RFC8201] extended for the ASIP-to-IPv4 encapsulation of §12.3. Operators who deploy ASIP over non-upgraded IPv4 transit MUST account for the overhead below, or applications will experience silent truncation, black-holing, or PTB storms.

Encapsulation overhead budget (single-level GENEVE/UDP/IPv4 over Ethernet):

Ethernet frame payload (L2 MTU) 1500 bytes - Ethernet header -14 - Outer IPv4 header -20 - Outer UDP header -8 - GENEVE header (no options) -8 - Inner ASIP base header -40
-- Available for inner L4 / application 1410 bytes

On a standard 1500-byte L2 link, an ASIP flow traversing ASIP-to-IPv4 encapsulation has a *1410-byte inner payload budget*, not 1500. Applications and middleware that assume a 1500-byte MTU will emit 1500-byte inner packets; those packets will either be silently truncated or trigger Packet Too Big storms, depending on the encapsulator’s policy.

Jumbo-frame paths. On end-to-end 9000-byte jumbo-frame paths, the budget is 9000 - 14 - 20 - 8 - 8 - 40 = 8910 bytes inner payload. Jumbo-frame paths are not universal; any path segment that does not support jumbo frames reduces the budget to that segment’s MTU. Operators deploying ASIP-to-IPv4 across mixed-MTU paths MUST NOT assume jumbo MTU end-to-end.

Normative requirements:

* *(MUST)* ASIP-aware hosts MUST implement PMTUD per [RFC8201]: on receipt of an ICMP-ASIP Packet Too Big (Type=2), the host MUST cache the reported MTU per destination (and per source address, see below) and MUST NOT emit packets larger than the cached MTU to that destination until the cache entry expires.

* *(MUST)* ASIP-to-IPv4 encapsulators MUST set the outer IPv4 DF bit so that intermediate IPv4-only links trigger ICMPv4 Type=3 Code=4 rather than silently fragmenting the outer IPv4 packet. Translation of the ICMPv4 error back to ICMP-ASIP Packet Too Big is specified in §12.7.4.

* *(MUST)* The ASIP minimum link MTU is 1280 bytes (identical to IPv6's minimum per RFC 8200 §5). On receipt of an ICMP-ASIP Packet Too Big whose reported MTU is less than 1280, the ASIP originator MUST clamp the cached PMTU to 1280 for the affected destination and MUST use a Fragment Extension Header (NH=44) to carry larger application payloads in 1280-byte pieces. Implementations MUST NOT set a cached PMTU below 1280 regardless of the reported value, and MUST NOT emit ASIP packets smaller than 1280 bytes on the wire except as trailing fragments. The "< 48" degenerate case (ASIP 40-byte base header + 8-byte minimum upper-layer header) is equally covered by this rule.

* *(MUST, per-source-address cache)* Hosts that hold multiple ASIP source addresses (multihoming, §8.3.1) MUST cache PMTU state keyed by the tuple (source address, destination address), not by destination alone. The forward path differs per source address because r.r.r.r steers inter-AS forwarding; PMTU to the same destination via two different locators MAY differ. Flushing the cache only on destination change will cause PTB storms when the host switches source addresses.

* *(SHOULD)* Encapsulating gateways SHOULD probe path MTU proactively (e.g., using Packetization-Layer PMTUD per [RFC8899]) rather than waiting for the first ICMP Packet Too Big event in the data flow.

* *(MAY)* Operators with strict application MTU requirements MAY configure their ASIP-to-IPv4 transit to use GRE (-24 outer: IPv4(20)+GRE(4)) or IP-in-IP (-20 outer: IPv4(20) only) instead of GENEVE/UDP (-36 outer: IPv4(20)+UDP(8)+GENEVE(8)). IP-in-IP recovers 16 bytes of inner payload per packet versus GENEVE/UDP at the cost of losing UDP-based NAT traversal and per-flow port entropy for ECMP hashing. The choice is a trade-off between payload efficiency and outer-transport properties.

Deliberate non-resolution: outer PMTU vs. inner PMTU. When an ASIP-to-IPv4 tunnel reports a path MTU via ICMPv4 on the outer path, the encapsulator MUST translate that to an ICMP-ASIP Packet Too Big on the inner flow (§12.7.4). When an ICMP-ASIP Packet Too Big is received on the inner flow from beyond the decapsulator, the encapsulator MUST NOT re-generate an ICMPv4 error on the outer path; the inner error is propagated to the inner originator directly. This is the standard IP-in-IP tunneling behavior [RFC4459] and is restated here because translation-plus-encapsulation paths combine both semantics.

13. 13. Application Compatibility

13.1. 13.1. Legacy Applications

Existing IPv4 applications require no modification. The ASIP socket compatibility layer intercepts standard BSD socket calls (AF_INET, connect(), bind(), etc.) and transparently manages the upper 96 bits via DNS-ASIP resolution and routing table state. The application never sees an ASIP address.

13.2. 13.2. New Applications

Applications that wish to leverage ASIP addressing directly MAY use the AF_ASIP address family and sockaddr_asip structure defined in Section 5.5. Libraries SHOULD provide helper functions for converting between ASIP canonical/compressed notation strings and sockaddr_asip structures.

13.3. 13.3. URL and URI Representation

ASIP addresses in URLs use canonical or compressed notation enclosed in brackets, consistent with IPv6 convention [RFC3986]:

https://[15169:0.0.0.1:0.0.0.10:192.0.2.1]:443/path (full, no compression needed) https://[15169::192.0.2.1]:443/path (flat network, compressed) https://[::8.8.8.8]:443/path (IPv4 compatible, compressed) https://[64496::10.0.0.1]:8080/api (documentation ASN, compressed)

Parsers MUST handle both compressed and uncompressed forms within brackets. The host field h.h.h.h is always present per Section 6.3 Rule 5, which prevents ambiguity with bare IPv4 addresses in URLs.

14. 14. Security Considerations

14.1. 14.1. ASN Locator Spoofing

ASIP border routers MUST implement ingress filtering validating that the source r.r.r.r of received packets is a legitimate origin for the eBGP-ASIP session over which the packet arrived. The legitimate-origin set is the union of the peer's own ASN locators and any ASN locators the peer has validly announced (including multihoming and anycast more-specifics per Section 8.3.1).

This is consistent with BCP 38 [RFC2827]. Because the locator is carried in the address rather than inferred from prefix ownership, the check is a direct field match rather than a lookup. Note that the check validates the locator, not the host identity: an operator who rewrites r.r.r.r at an internal boundary (Section 8.3.2) MUST ensure that the rewritten locator still passes the ingress-filter check at the next external boundary.

IPv4-mapped source (r=z=s=0) handling at ingress. An incoming Version=8 frame with source r.r.r.r=0 is by definition a packet emitted (or re-emitted) by a §12.2 translator: either (a) IPv4-to-ASIP forward, where the translator prepended r=z=s=0 on ingress, or (b) IPv4-to-ASIP reply to a native ASIP client, where the IPv4 source gets lifted to ::S4 (§12.7.1 reverse direction, the common case of every reply from an IPv4 destination to an ASIP-native originator). Both cases produce legitimate downstream traffic whose source locator is 0.0.0.0. The filter below treats them uniformly. A border router MUST accept r=z=s=0 sourced Version=8 frames on an interface if and only if *both* of the following hold: (i) the peer's legitimate-origin set on that eBGP-ASIP session (or administratively configured adjacency) contains the 0.0.0.0::/96 IPv4-mapped prefix; and (ii) the receiving interface satisfies a uRPF-style reception check for 0.0.0.0::/96 in the sense of [RFC3704]. Condition (ii) prevents the one-bit-flip attack in which a transit peer that merely re-advertises 0.0.0.0::/96 thereby unlocks blanket acceptance of r=z=s=0 source packets across _all_ its customer-facing ingress without applying BCP 38 to its own customers; condition (ii) restricts acceptance to interfaces consistent with the route for the IPv4-mapped prefix. Transit ASes that re-advertise 0.0.0.0::/96 MUST themselves apply §14.14's intra-AS BCP 38 rule on their customer-facing ingress, so that a downstream customer cannot spoof r=z=s=0 into a transit AS that is itself trusted by its upstream peers. Carriage of the 0.0.0.0::/96 prefix is expected to be propagated by normal eBGP-ASIP advertisement from the translator-owning ASN outward, exactly as any other originated prefix; transit ASes re-advertise it and MUST include it in the outbound legitimate-origin set they present to their own downstream peers. A receiving border router that is a transit hop several ASNs away from the originating translator will therefore see 0.0.0.0::/96 in its upstream peer's advertised set and,

provided the packet arrives on the interface associated with that upstream route (uRPF condition (ii)), will correctly accept r=z=s=0 reply traffic. This preserves the §12.7.1 reverse-direction reply path across arbitrary-length multi-AS transit.

uRPF mode selection (normative). Condition (ii) admits two modes from [RFC3704]: strict (packet accepted only if the best-path route for 0.0.0.0::/96 in the router's RIB points out the same interface on which the packet arrived) and feasible-path/loose (packet accepted if any route for 0.0.0.0::/96 exists via the receiving interface, whether or not best-path; loose mode accepts if any route exists in the RIB at all, regardless of interface). Leaving the mode unspecified would produce implementation divergence: one vendor strict-mode-drops legitimate asymmetric reply traffic, while another vendor loose-mode-accepts exactly the one-bit-flip pattern condition (ii) was written to block. Accordingly: - On single-homed edge eBGP-ASIP interfaces facing stub customers or transit-free peers, routers MUST apply strict-mode uRPF for 0.0.0.0::/96; strict mode is the mode that actually closes the one-bit-flip gap. - On multihomed-customer-facing interfaces and on transit/core interfaces where asymmetric return paths are expected (e.g., a customer multihomed to two upstreams advertising via primary but receiving replies via backup), routers MUST apply feasible-path uRPF per [RFC3704] §2.2 (packet accepted if the receiving interface is _any_ reverse path for 0.0.0.0::/96 in the RIB, not only the best path). Feasible-path uRPF tolerates legitimate asymmetry while still requiring a per-interface route association, which continues to block the one-bit-flip attack from an interface that has no 0.0.0.0::/96 route at all. - Loose-mode uRPF (the "any route anywhere" variant) MUST NOT be used for 0.0.0.0::/96 reception on any eBGP-ASIP interface; loose mode degenerates condition (ii) into condition (i) and reintroduces the one-bit-flip acceptance gap that condition (ii) was written to block. - During a transient RIB state where 0.0.0.0::/96 is absent (BGP flap, session reset), condition (ii) will fail and legitimate r=z=s=0 replies will be dropped for the duration of the absence. This is the same transient-drop behavior that RFC 3704 uRPF applies to IPv4 BCP 38 and is accepted as a standard operational cost; operators SHOULD minimize 0.0.0.0::/96 churn by configuring the prefix with eBGP-ASIP dampening disabled and with a stable originator. The mode selection above is enforceable at configuration time: the operator knows which of its interfaces are single-homed-edge vs. multihomed/transit, and per-interface mode configuration is standard in every eBGP-ASIP-capable router. An implementation that cannot determine the correct mode per-interface MUST default to strict mode and reject r=z=s=0 traffic on any interface where strict uRPF fails; this is the safe default. On any eBGP-ASIP session whose legitimate-origin set does NOT contain 0.0.0.0::/96, or where the uRPF reception check fails, packets with r=z=s=0 source MUST be dropped: the all-zero locator is

not a valid origin for a peer that has not propagated the IPv4-mapped originator prefix, and an attacker on such a link cannot bypass the §14.1 locator-match filter by forging r=z=s=0 (which would otherwise evade the filter because no non-zero ASN is present to compare against). Operators who deploy stateless §12.2 translation at their AS boundary MUST originate the 0.0.0.0::/96 prefix via eBGP-ASIP so that downstream reply traffic flows are not black-holed by peer ingress filters; a translator deployment that silently serves §12.2 without this advertisement will see its reverse-direction replies dropped at the first non-translator-declared boundary. Operators deploying only §12.3 encapsulation (ASIP-to-IPv4 tunneling) without §12.2 translation MUST NOT originate 0.0.0.0::/96, because they do not serve that prefix; originating a prefix one does not serve is a hijack pattern WHOIS-ASIP (§8.4) is specifically designed to reject.

14.2. 14.2. Internal Zone Prefix Protection

The 127.x.x.x internal zone prefix MUST NOT appear on WAN interfaces. Border routers MUST drop packets with 127.x.x.x as source or destination r.r.r.r on external interfaces and SHOULD log each violation.

14.3. 14.3. RINE Prefix Protection

The 100.x.x.x RINE prefix MUST NOT appear in eBGP-ASIP advertisements or on non-peering interfaces. Border routers MUST filter these prefixes and SHOULD log violations.

14.4. 14.4. Interior Link Convention Protection

Border routers MUST filter received eBGP-ASIP route advertisements whose announced reachability would treat the 222.0.0.0/8 h.h.h.h range as an externally routable interior-link address. This is a control-plane filter applied to route advertisements only.

Border routers MUST NOT filter data-plane packets based solely on the h.h.h.h value being in 222.0.0.0/8. The h.h.h.h field is locally significant and other ASes may legitimately use any h.h.h.h value for any purpose. Per Section 3.8, route-advertisement filtering and data-plane filtering are distinct and MUST NOT be conflated.

14.5. 14.5. RFC 1918 Address Privacy

RFC 1918 private addresses in h.h.h.h remain non-routable on the public internet, consistent with IPv4 behavior.

14.6. 14.6. Prefix Granularity Enforcement

More-specifics than /32 in r.r.r.r. Prefixes more specific than /32 in the r.r.r.r field subdivide a single ASN locator’s address space across multiple route advertisements. Such announcements SHOULD NOT be accepted at external eBGP-ASIP boundaries. They MAY be accepted when they are explicitly listed in the originating ASN’s WHOIS-ASIP record (Section 8.4) as legitimate multihoming, anycast, or TE more-specifics (Section 8.3.1). A blanket "no deaggregation ever" rule would force operators to split into additional ASNs to achieve the same traffic-engineering granularity, which grows the locator count and does not shrink the table; the WHOIS-ASIP-authorized-more-specifics model supports the real use cases BGP4 already supports without this perverse incentive.

Less-specifics than /32 in r.r.r.r. Aggregation covering multiple ASN locators (less specific than /32) MAY be performed for internal backbone routing but MUST NOT be advertised across peering boundaries, because such an aggregate obscures the per-ASN origin that WHOIS-ASIP validation depends on.

This resolves the tension between § 8.3 aggregation-to-/16 and the "one route = one ASN" property: aggregation is a routing-table optimization internal to an operator, not a peering-advertisement practice.

14.7. 14.7. Header Overhead and DDoS

The 40-byte ASIP header is 20 bytes larger than IPv4’s minimum header and identical in size to IPv6. In volumetric DDoS scenarios, this increases per-packet overhead for both attacker and defender. The net effect is approximately equivalent; the additional header bytes do not create a meaningful amplification vector. Rate limiting and traffic scrubbing operate identically to IPv4 and IPv6 at the transport layer.

14.8. 14.8. Privacy Considerations

The r.r.r.r field reveals the origin ASN of every packet. This is functionally equivalent to existing BGP prefix-to-AS mapping via whois lookups, but encoded directly in the address rather than requiring a secondary lookup. Operators concerned about origin AS privacy at the packet level may use VPN or tunnel encapsulation to mask the outer ASIP header, identical to existing practice with IPv4/IPv6.

The z.z.z.z and s.s.s.s fields reveal internal network topology to external observers. Operators who consider this unacceptable SHOULD use XLATE-ASIP (address translation) at their AS boundary to replace internal z.z.z.z and s.s.s.s values with a public-facing address before packets exit the AS. This is recommended for all operators as a baseline privacy practice.

14.9. 14.9. SLAAC-ASIP Security

SLAAC-ASIP inherits the security properties and risks of IPv6 SLAAC [RFC4862]:

* *Rogue Router Advertisements:* A malicious device on a link may send forged Router Advertisements containing incorrect prefixes, redirecting traffic or causing denial of service. RA Guard [RFC6105] adapted for ASIP SHOULD be deployed on all access switches.

* *DAD Attacks:* An attacker may respond to every DAD probe, preventing legitimate devices from configuring addresses. This is mitigated by limiting DAD attempts and falling back to DHCP-ASIP.

* *Host ID Tracking:* MAC-derived host identifiers (SLAAC-ASIP Method 3) enable device tracking across networks. Implementations SHOULD default to Method 1 (stable opaque) or Method 2 (temporary random) to prevent this.

* *Temporary Address Rotation:* Method 2 temporary addresses SHOULD be regenerated on a configurable interval (default 24 hours) and MUST be regenerated when moving between networks.

14.10. 14.10. Link-Local Scope Enforcement

Link-local addresses (169.254.0.0/16 in r.r.r.r) MUST NOT be forwarded by any router under any circumstances. Routers MUST silently drop packets with link-local source or destination addresses received on any interface other than the originating link. This enforcement prevents link-local addresses from being used as a side channel to bypass scope boundaries.

14.11. 14.11. Scope Boundary Enforcement

Each scope level (Section 4.1) represents a security boundary. Routers at scope boundaries MUST enforce the containment hierarchy:

* Link-local traffic MUST NOT exit the link.

* Mesh-scoped traffic MUST NOT exit the mesh domain. Conversely, non-mesh traffic MUST NOT enter a mesh domain natively (§ 3.11; the only sanctioned entry is a mesh-gateway-performed translation).

* Organization-scoped traffic (127.x.x.x) MUST NOT exit the AS.

* Terrestrial traffic MUST NOT be forwarded into a non-terrestrial realm without explicit relay configuration. Conversely, a terrestrial router receiving a packet whose source r.r.r.r lies in a non-terrestrial realm range (97/8, 98/8, 99/8, 241/8249/8, 250/8) on an interface that is not a configured realm-relay ingress MUST drop the packet. A terrestrial router MUST NOT accept such a packet on general peering links; realm-sourced traffic is accepted only on a physical or cryptographic channel configured as a realm-relay ingress. Without this ingress-side restriction, an attacker in a terrestrial peer's path could spoof a non-terrestrial source r.r.r.r and bypass realm-boundary controls that § 3.12 authorizes only at designated relay points.

* Cross-realm multicast is governed by § 10.6; no multicast scope crosses a realm boundary natively in this document.

Violation of scope boundaries MUST be dropped as specified above and SHOULD be logged as a security event.

14.12. 14.12. Extension Header Security

ASIP reuses IPv6's extension header mechanism and therefore inherits the same security considerations [RFC7045, RFC8200]. Implementations MUST:

* Limit the number of extension headers per packet to at most 8 distinct headers and the total length of the extension-header chain to at most 256 bytes. Packets exceeding either bound MUST be dropped at the ingress node and SHOULD be counted. These limits make the "long chain of extension headers" IDS-evasion attack [RFC7112] ineffective while preserving realistic legitimate use (at most one Hop-by-Hop, one Destination Options before routing, one Routing, one Fragment, one AH or ESP, one Destination Options after routing per RFC 8200 § 4.1).

* Require the full ASIP header chain, through and including the upper-layer protocol header, to be contained in the first fragment of a fragmented packet. Packets that violate this requirement MUST be dropped. This matches [RFC7112] and prevents first-fragment-only inspection bypass.

* Drop packets with unrecognized extension header types at intermediate nodes (unless the header is a Destination Options header, which is only processed at the final destination).

* Implement rate limiting on packets with Hop-by-Hop Options headers, which require processing at every node.

14.13. 14.13. Flow Label Security

The flow label is source-assigned and opaque to the network. Routers MUST NOT trust flow labels for security decisions. An attacker may set arbitrary flow labels to manipulate ECMP distribution or QoS classification. Flow labels SHOULD be used as hints for performance optimization, never as security-relevant identifiers.

14.14. 14.14. STRIDE Summary

The preceding §§14.1-14.13 address individual threats. This subsection consolidates coverage against the STRIDE categories (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) so that gaps are visible in one place. Normative statements in this subsection that are not already covered elsewhere are load-bearing.

Spoofing.

* ASN-locator spoofing at peer ingress: covered normatively in §14.1.

* Intra-AS source-address spoofing (a host on the internal side of r.r.r.r faking a different z:s:h): NOT covered at the protocol layer. Operators MUST deploy BCP 38-equivalent ingress filtering on internal AS-access segments; this is a standard operational control and is not specific to ASIP, but is named here so that implementers do not assume §14.1's border-only filter suffices.

* DNS-ASIP response spoofing: mitigated by DNSSEC at the DNS layer; ASIP imposes no additional requirements. A-ASIP records MUST be served over DNSSEC-validated zones where the deploying operator relies on A-ASIP-vs-A preference for policy decisions.

* WHOIS-ASIP response spoofing: covered normatively by §8.4 "Response authentication."

* ICMP-ASIP error-source spoofing: an on-path or off-path attacker can forge an ICMP-ASIP Packet Too Big or Destination Unreachable to poison a host's PMTU cache or abort a connection. Hosts MUST validate that the inner header of an ICMP-ASIP error corresponds

to a packet the host recently sent (source address, destination address, and at least 8 bytes of upper-layer header, identical to RFC 5927 guidance for ICMPv4); errors that do not match MUST be discarded.

* NDP / RA spoofing on link-local: mitigated by RA Guard (§ 14.9) and by NDP cache entry validation; no new normative text needed.

Tampering.

* In-transit modification of unprotected traffic: not addressable at the IP layer; relies on upper-layer integrity (TLS, QUIC, AH, ESP). § 14 makes no additional claim.

* Extension-header chain tampering: bounded by § 14.12 (max 8 headers, max 256 bytes, first-fragment inclusion).

* GENEVE option tampering: mitigated by § 12.3's "no options in the baseline" rule; future option-class additions MUST define their own integrity story in the companion specification that introduces them.

* Flow-label manipulation by intermediate nodes: forbidden by § 5.4 ("Routers MUST NOT … rewrite it in the forwarding path"); an on-path tamperer can manipulate it, but doing so is classified as a QoS/ECMP impact only and never a security-relevant change, per § 14.13.

Repudiation.

* Translator logging: § 14.15 (below) requires an observable counter of failed ICMP-error translations; operators SHOULD additionally log stateful translation session establishment and teardown at rates compatible with their incident-response workflow. This document does not mandate a specific log schema.

* Locator-rewrite attribution gaps: when § 8.3.2 rewrites r.r.r.r, downstream logs show the rewritten locator, not the original. Operators performing rewrite MUST retain a (timestamp, pre-rewrite, post-rewrite, peer-session) audit record for the period required by applicable policy; failing this, post-incident attribution across a rewrite boundary is impossible.

* CF telemetry attribution: CF is specified in a separate companion document (§ 17 / draft-asip-cf-00); attribution of CF values to specific peers is a CF-companion concern and is not pinned down here.

Information disclosure.

* Flow-label as a side-channel: covered normatively in § 14.13.

* Extension-header chain as a fingerprinting vector: a responder's choice of extension-header ordering (AH vs. ESP placement, presence of Destination Options) can identify the implementation. Implementations SHOULD NOT emit optional extension headers that are not required by the packet's function; this is an RFC 8200 best practice restated for ASIP.

* Translator state as a timing oracle: a per-flow translator that differentiates response latency between "first packet of a flow" (state creation) and "subsequent packets" (state hit) exposes an observable that an attacker can use to probe another user's flow existence. Translators SHOULD minimize the timing delta between state-miss and state-hit, or SHOULD apply a constant-time pad to external-facing response latency.

* Multicast-scope network-topology disclosure: a receiver that observes which scope nibbles reach it learns the network's scope-boundary topology. This is not a new threat class (IPv6 MLD has the same property) and is accepted.

* Internal zone (127.x) address leakage: covered normatively in § 14.2. A packet with 127.x source arriving on a WAN interface is a configuration error that leaks internal topology; the MUST-drop rule in § 14.2 prevents propagation.

Denial of service.

* Translator state exhaustion: covered in § 14.15 (below).

* GENEVE reflection: covered in § 12.3 "Tunnel endpoint authentication and reflection/amplification."

* ICMP-ASIP amplification: ICMP-ASIP Echo Reply is 1:1 in size with Echo Request and does NOT amplify on its own. However, ICMP-ASIP Destination Unreachable messages whose inner payload is set to the maximum permitted 1232-byte quotation ([RFC4443]-equivalent) do amplify a small triggering packet to ~1232 bytes plus headers. Routers emitting ICMP-ASIP errors MUST apply rate limits per (source, destination) pair, as required of ICMPv6 routers by [RFC4443] § 2.4. Unsolicited Neighbor Advertisements and Router Advertisements are link-scoped and do not amplify across routers.

* SLAAC DAD storms: a malicious host that responds positively to every DAD probe can starve a subnet. Mitigated by §3.14.3's bounded-retry rule and DHCP-ASIP fallback, plus RA Guard per §14.9.

* WHOIS-ASIP query amplification: WHOIS-ASIP responses can be substantially larger than queries (especially for prefix-to-ASN enumerations). WHOIS-ASIP servers MUST apply per-client rate limits and MUST NOT answer unauthenticated bulk-enumeration queries over UDP. The authenticated-channel requirement of §8.4 ("Response authentication") restricts large-response queries to TLS-authenticated clients.

* Flow-label-driven router-CPU exhaustion via ECMP hashing: flow labels are source-assigned and an attacker can choose labels to maximize cache-line contention in an ECMP hash-bucket table. Routers SHOULD include a random per-router seed in ECMP hashing so that an attacker cannot target a specific hash bucket without knowing the seed.

Elevation of privilege.

* Multicast scope escape: covered by §10.6 composition rules and by §14.11 scope-boundary enforcement.

* Realm-boundary escape: covered by §3.12 "Cross-realm authentication" and by §14.11; supplemented by §10.6 rule 4 for multicast.

* Internal-zone (127.x) addresses leaking externally: covered by §14.2.

* eBGP-ASIP route injection by a non-authorized AS: covered by §8.4's WHOIS-ASIP validation where deployed; where WHOIS-ASIP is not enforced, operators accept BGP4-equivalent route hygiene (§8.4 "Normative status") and an injection remains possible, same as today's BGP4.

14.15. 14.15. Translator State and ICMP Error Attribution

§12.7.1 and §12.7.3 require translators to maintain a mapping between IPv4 literals and the originating ASIP source locator in order to reconstruct ICMP errors and return-path source addresses. This mapping is an attack surface:

* *Memory exhaustion via ICMP flood.* If the mapping is per-flow session state, an attacker can exhaust translator memory by opening many short-lived flows and solicit ICMP errors that force state allocation. Translators SHOULD bound per-session state, apply LRU eviction, and log state-table pressure events.

* *Stale-mapping misattribution.* If the mapping is administrative/static, a long-lived mapping entry can survive past the originating ASIP host's address changes (locator transfer, ASN renumbering). Misattributed ICMP errors are a low-severity information leak but not a traffic-injection vector. Operators SHOULD age out administrative mappings on ASN-transfer events.

* *Observability requirement.* Translators MUST expose a counter of ICMP-error translations that failed for lack of mapping state; a sudden increase in this counter indicates either an attack or a misconfigured mapping and SHOULD be monitored.

* *Short-flow state lifetime.* Echo and single-probe flows (§ 12.7.2) generate high-rate, low-value mapping entries that, if aged at the same timeout as TCP/UDP flow state, amplify the memory-exhaustion surface above. Translators that maintain per-flow mapping state SHOULD apply a distinct short timeout (RECOMMENDED: 60 seconds) to Echo/ping-class traffic and MAY apply the longer TCP/UDP default only to flows carrying a transport connection. This caps Echo-based state amplification without harming correlation of legitimate traceroute and PMTU probes.

15. 15. IANA Considerations

15.1. 15.1. IP Version Number

IANA is requested to assign version number 8 in the IP Version Number registry to ASIP (AS-Structured Internet Protocol), the 128-bit four-field protocol specified in this document.

15.2. 15.2. Address Family

IANA is requested to assign:

* An Address Family Identifier (AFI) for ASIP in the Address Family Numbers registry. This is the AFI that eBGP-ASIP (§ 8.3), iBGP-ASIP, and any Multiprotocol BGP [RFC4760] usage MUST carry to announce ASIP reachability.

* A Subsequent Address Family Identifier (SAFI) for ASIP unicast.

* A Subsequent Address Family Identifier (SAFI) for ASIP multicast (for MP-BGP carriage of the cross-ASN multicast prefixes defined in §10.3 and §4.2).

eBGP-ASIP conformance (§8.3) requires the AFI/SAFI assignment to exist before interoperable inter-AS ASIP routing can be deployed. Until IANA has assigned these values, experimental deployments MAY use a pair of Private Use AFI/SAFI values agreed between peers; such deployments MUST NOT advertise prefixes over assigned-value peering until the formal assignment is completed.

15.3. 15.3. Reserved ASN Ranges

IANA is requested to reserve the following ASN ranges for ASIP use. Each /8 in r.r.r.r spans 2^24 = 16,777,216 ASN values; a /16 spans 2^16 = 65,536. Rows below that span a single /8 or /16 carry the corresponding 2^24 or 2^16 Count; rows that span multiple /8 ranges (241.0.0.0249.255.255.255; 251.0.0.0253.255.255.255) carry the sum, and the partial-/8 row 255.0.0.0255.255.254.255 excludes the 256 values reserved for cross-ASN multicast and broadcast per §15.6 and §15.7:

ASN Range (decimal)	r.r.r.r Equivalent	Count	Purpose
1,627,389,952 - 1,644,167,167	97.0.0.0/8	16,777,216	Near-Earth infrastructure realm (Informative, Section 3.12)
1,644,167,168 - 1,660,944,383	98.0.0.0/8	16,777,216	Cislunar / Lunar realm (Informative, Section 3.12)
1,660,944,384 - 1,677,721,599	99.0.0.0/8	16,777,216	Martian realm (Informative, Section 3.12)
1,677,721,600 - 1,694,498,815	100.0.0.0/8	16,777,216	RINE peering fabric
2,130,706,432 - 2,147,483,647	127.0.0.0/8	16,777,216	Internal zone prefixes

2,851,995,648 - 2,852,061,183	169.254.0.0/16	65,536	Link-local scope
4,026,531,840 - 4,043,309,055	240.0.0.0/8	16,777,216	Mesh / ad-hoc scope
4,043,309,056 - 4,194,303,999	241.0.0.0 - 249.255.255.255	150,994,944	Future celestial bodies (Informative)
4,194,304,000 - 4,211,081,215	250.0.0.0/8	16,777,216	DTN relay realm (Informative)
4,211,081,216 - 4,261,412,863	251.0.0.0 - 253.255.255.255	50,331,648	Reserved for future use
4,261,412,864 - 4,278,190,079	254.0.0.0/8	16,777,216	Documentation and testing
4,278,190,080 - 4,294,967,039	255.0.0.0 - 255.255.254.255	16,776,960	Reserved for future use (non-multicast/ broadcast)

Table 15

15.4. 15.4. DNS A-ASIP Record Type

IANA is requested to assign a DNS resource record type number for the A-ASIP record type defined in Section 7.

15.5. 15.5. ICMP-ASIP Next-Header Value and Type Registry

IANA is requested to assign a previously-unassigned value in the IP Protocol Numbers / IPv6 Extension Header registry for ICMP-ASIP. The requested value is 143; any unassigned value in the range 143-252 that IANA prefers is acceptable. The assigned value MUST NOT be 58 (ICMPv6) to prevent dispatcher ambiguity across header-stripping middleboxes.

IANA is also requested to establish a registry for ICMP-ASIP message types, initially populated with types corresponding to ICMPv4 [RFC792] and ICMPv6 [RFC4443] equivalents including Neighbor Solicitation, Neighbor Advertisement, Router Solicitation, and Router Advertisement for SLAAC-ASIP and neighbor discovery.

15.6. 15.6. Cross-ASN Multicast Registry

IANA is requested to establish a registry for ASIP cross-ASN multicast prefix assignments within r.r.r.r = 255.255.255.0/24, with scope values as defined in Section 10.1.

15.7. 15.7. Broadcast Reservation

IANA is requested to reserve r.r.r.r = 255.255.255.255 as the ASIP broadcast address.

15.8. 15.8. Next Header Values

ASIP draws from the IPv6 Next Header registry for shared protocols (TCP, UDP, ESP, AH, Routing, Fragment, Destination Options, No Next Header). It requests a distinct value for ICMP-ASIP (Section 15.5) rather than reusing NH=58, to eliminate dispatcher ambiguity at header-stripping middleboxes.

15.9. 15.9. GENEVE Inner-Protocol Assignment for ASIP-to-IPv4

IANA is requested to assign a GENEVE [RFC8926] inner-protocol type identifying encapsulated ASIP frames for use by the ASIP-to-IPv4 transit encapsulation defined in Section 12.3.

15.10. 15.10. WHOIS-ASIP Registry Domain Delegation

IANA is requested to delegate one or more well-known registry domains for WHOIS-ASIP service discovery (§ 8.4 "Service discovery"). Each delegated domain MUST carry an _whois-asip._tcp SRV record resolvable over IPv4 so that a bootstrapping ASN can locate its regional WHOIS-ASIP service without a pre-existing ASIP path. The specific delegation structure (single global domain vs. per-RIR domains) is left to IANA and the RIR community; this specification requires only that the delegation exist and that the SRV record be IPv4-resolvable during transition.

16. 16. Zone Server Reference Architecture (Informative)

This section is INFORMATIVE. Nothing in this section is a protocol-layer requirement. Operators MAY deploy ASIP addressing without any Zone Server infrastructure.

16.1. 16.1. Concept

The Zone Server is a reference architecture for unified network service delivery. A Zone Server is a paired active/active platform that consolidates address assignment (DHCP-ASIP), name resolution (DNS-ASIP), time synchronization (NTP8), telemetry collection (NetLog8), authentication caching, route validation, access control, and IPv4/ASIP address translation (XLATE-ASIP) into a single operational platform.

The motivation is operational: modern networks require a minimum of 6-8 independently configured, independently authenticated, independently monitored services before a device is operational. The Zone Server consolidates these into a single platform with a shared identity model and a single configuration surface.

16.2. 16.2. Service Delivery Model

A device connecting to a Zone Server-equipped network sends one DHCP-ASIP Discover and receives a single response containing every service endpoint it requires. No subsequent manual configuration is needed. The device is fully operational (addressed, authenticated, time-synchronized, logged) before its first user interaction.

16.3. 16.3. Authentication Model

The Zone Server reference architecture recommends OAuth2 JWT [RFC7519] as the universal authentication mechanism. Tokens are validated locally by the Zone Server without round trips to external identity providers. This enables continued operation when upstream identity providers are unreachable.

Operators MAY use alternative authentication mechanisms. The Zone Server architecture is a recommendation, not a protocol-layer mandate.

16.4. 16.4. Why This Is Informative, Not Normative

Bundling operational architecture into a protocol specification is a category error. Protocols define wire formats and interoperability requirements. Operational architecture defines how organizations deploy and manage their networks. These are different concerns with different rates of change, different stakeholders, and different consensus requirements.

The Zone Server architecture is published as a companion specification to enable operators who want unified management to deploy it. Operators who do not want it lose nothing. ASIP addressing works without it.

17. 17. Cost Factor Routing Metric (Informative)

The Cost Factor (CF) routing metric is specified in a separate companion document (draft-asip-cf-00). CF is intended as an OPTIONAL multi-dimensional path-selection input beyond the standard BGP attributes (LOCAL_PREF, AS_PATH, ORIGIN, MED). CF is NOT a protocol-layer requirement of this document: ASIP eBGP-ASIP sessions MUST function correctly using only standard BGP path-selection criteria [RFC4271], and no normative behavior defined in §§315 of this document depends on CF. An operator MAY deploy ASIP without any CF awareness; conformance to this specification does not require parsing, emitting, or acting on a CF path attribute. Implementers seeking the CF component set, accumulation semantics, wire-format encoding, and physics-floor rules MUST consult the companion draft; partial CF machinery is intentionally not reproduced here because a partial reproduction could be treated as specification despite not being a complete one, which would fork CF semantics between the two documents.

18. 18. Acknowledgment of Trade-offs

Every protocol design involves trade-offs. This section makes them explicit rather than pretending they don’t exist. Items marked UNRESOLVED describe cases where the design has no clean answer; they are documented here rather than hidden.

18.1. 18.1. Header Overhead

The 40-byte ASIP base header is 20 bytes larger than IPv4 (20 bytes) and identical in size to IPv6 (40 bytes). By adopting IPv6’s fixed-header design (dropping IHL, header checksum, and inline fragmentation), ASIP achieves header parity with IPv6 while carrying the same 128-bit address space in a structured four-field format. On bandwidth-constrained links (satellite, IoT, cellular), the 20-byte overhead versus IPv4 is non-trivial. Header compression (ROHC or equivalent adapted for ASIP) is RECOMMENDED for such environments.

Per-packet overhead in ASIP-to-IPv4 transit is higher because of the outer IPv4 and UDP headers. The total bytes above the inner L4 payload are ASIP(40) + IPv4(20) + UDP(8) + GENEVE(8) = 76 bytes, of which the inner ASIP header contributes 40 bytes and the encapsulation framing (IPv4 + UDP + GENEVE) contributes 36 bytes. An HTTPS (TCP/TLS) tunneling alternative costs 109 total bytes per

packet (ASIP(40) + IPv4(20) + TCP(20) + TLS(~29)) and introduces TCP congestion-control interactions; GENEVE/UDP is preferred because it avoids both the extra 33 bytes and the TCP-over-TCP pathology (see § 12.3 rationale).

18.2. 18.2. Rigid Hierarchy

The fixed four-field address structure (ASN/Zone/Subnet/Host) imposes a specific network topology model. Organizations whose networks do not map naturally to this hierarchy (highly flat networks, mesh topologies, multi-ASN single-logical-network deployments) may find the structure constraining. The Zone and Subnet fields are locally significant and operator-assigned, which provides flexibility within an organization. The r.r.r.r field is drawn from the global ASN number space and is constrained by external allocation.

18.3. 18.3. ASN Dependency

Every ASIP-routable address requires an ASN locator. Organizations that do not currently hold an ASN must obtain one before deploying ASIP with public-facing addresses. The ASN allocation process is well-established via RIRs but adds a bureaucratic step that IPv4 and IPv6 do not require for basic internet connectivity. Organizations using only internal zone addressing (127.x.x.x) or link-local addressing (169.254.x.x) do not require a public ASN.

18.4. 18.4. Transition Realism

ASIP is not wire-compatible with IPv4. Every router, OS kernel, NIC driver, firewall, IDS/IPS, load balancer, and middlebox in the forwarding path of an ASIP-aware endpoint must receive a software update to process Version=8 frames natively; hardware that cannot be updated must be replaced or bypassed via the Section 12.3 encapsulation. Until that happens for each deployment segment, ASIP-to-IPv4 encapsulation handles transit, but encapsulation adds overhead (36 bytes of encapsulation framing per packet: outer IPv4(20) + UDP(8) + GENEVE(8); total bytes above the inner L4 payload are 76 with the 40-byte inner ASIP header included) and operational complexity (tunnel MTU, PMTU discovery, endpoint discovery). The transition is incremental, not instantaneous, and not free. The abstract and Section 1.2 state this plainly rather than claiming "no modification required."

18.5. 18.5. Relationship to IPv6

ASIP does not claim that IPv6 is technically deficient. IPv6 is a well-designed protocol that solves the address exhaustion problem, and ASIP borrows extensively from its design: the fixed 40-byte header, extension header mechanism, flow label, traffic class, SLAAC model, scoped multicast, and the elimination of header checksums and router-path fragmentation are all IPv6 innovations adopted wholesale. ASIP's residual contribution is the structured four-field address format (ASN locator / Zone / Subnet / Host), the IPv4-mapped address form, and a more structured relationship between the routing locator and the registered ASN. Whether this narrower contribution justifies a new protocol version rather than a set of IPv6 deployment guidelines is a legitimate question that this document does not claim to settle. Both protocols MAY coexist indefinitely.

18.6. 18.5a. UNRESOLVED: Multihoming Under an ASN-Shaped Address

Section 8.3.1 handles multihoming by assigning one address per upstream ASN. This preserves capability but has a real cost: a multihomed host presents multiple stable addresses to applications, DNS-ASIP, and any system that identifies a host by its address literal rather than by name. Applications that hash or pin on IP address (legacy configuration files, certain rate-limit systems, session-affinity middleware) will see the same host as two hosts. Happy-eyeballs-style selection helps on the client side but does not solve the server-side identification problem. Alternatives considered and rejected were: (i) a separate non-locator host identifier field, which would have required a header redesign; (ii) identifier/locator split via HIP or LISP-style mapping, which would have added a mapping-service dependency on the critical path. Neither was compatible with the "familiar dotted-decimal, no new control plane" goal. The chosen trade-off is documented here as a cost the design accepts, not a cost it resolves.

18.7. 18.5b. Server-Side Multi-Address Operational Impact

The multi-address model in § 8.3.1 creates concrete operational consequences for any server-side system that derives identity or state from the client's source IP literal. This subsection names the affected systems, prescribes mitigations, and states the normative status of each mitigation. It converts § 18.5a's acknowledgment of the multihoming trade-off into deployable operator guidance.

Affected systems (non-exhaustive):

System	Failure mode under multi-address
HTTP session stickiness keyed on source IP	One user is routed to different backends across requests if the client switches between its per-ASN addresses.
Rate limiters keyed on source IP	Effective per-user limit is multiplied by the number of upstream ASNs; DoS defenses under-count.
Stateful firewalls with per-IP state tables	Return traffic on a different source address than the forward flow is dropped or mismatched to a different state entry.
IP allow-lists / IP block-lists	A client reachable via ASN_A is allow-listed by its ASN_A address; the same client arriving via ASN_B is not recognized.
TLS session resumption keyed on peer IP	Resumption tickets bound to the ASN_A address fail when the client reconnects from its ASN_B address, forcing full handshakes.
Abuse-reporting, audit logs, SIEM correlation	A single user action appears in logs as multiple distinct sources; incident investigation must cross-reference.
CAPTCHA / risk-score systems keyed on IP	Reputation does not aggregate across a client’s addresses.
Path-bound PMTU caches (see § 12.8)	A host must track PMTU per source address, not per destination, because the forward path differs per address.

Table 16

Mitigations. The following mitigations address the above failure modes. Normative status is assigned per mitigation; operators deploying ASIP servers in the public internet SHOULD implement the normative ones.

* *(Normative, SHOULD) Prefer session-token stickiness over IP stickiness.* HTTP load balancers SHOULD use cookie-based or JWT-based stickiness rather than source-IP hashing when serving ASIP

clients. This is the single most effective mitigation and the one operators can implement without any ASIP-specific support from clients.

* *(Normative, SHOULD) Group by ASN in rate limiters.* Rate limiters SHOULD include the r.r.r.r ASN locator as an aggregation key in addition to the full 128-bit address, so that clients from the same enterprise ASN aggregate into one bucket. This is effective for the common multihoming case where a client's two addresses share the enterprise's two transit ASNs. For clients whose multiple addresses come from disjoint ASNs (e.g., a home user multihomed across two retail ISPs), ASN-grouping does not aggregate and token-based identity (cookies, JWTs) SHOULD be used instead.

* *(Normative, SHOULD) Export z:s:h as the stable-identity triple in logs.* Where logs must correlate a user across addresses, the z:s:h 96-bit triple under the same administrative scope (§ 3.1 identifier/locator separation) is the stable portion. Logging systems SHOULD record both the full address and the z:s:h triple so downstream correlation is possible.

* *(Advisory, MAY) DNS-based address preference rules.* Authoritative DNS-ASIP servers MAY order A-ASIP records in a consistent preference order per client (by ASN geography, by peering cost, etc.) so that a given client preferentially selects the same server-side address most of the time, reducing but not eliminating the multi-address fan-out. This is advisory because it depends on DNS resolver and client cache behavior that the server cannot control.

* *(Normative, MUST -- see § 12.8) Per-address PMTU cache.* Hosts holding multiple ASIP source addresses MUST cache PMTU state keyed by (source address, destination address), not by destination alone, because the forward path is a function of the chosen source address. The MUST status is defined in § 12.8 and applies identically here; this subsection does not weaken it.

* *(Advisory, MAY) TLS session resumption across addresses.* TLS server implementations MAY accept resumption tickets regardless of peer-IP continuity when the ticket includes a client-identity binding (PSK or channel ID). This is advisory because RFC 8446 does not require IP continuity and most existing implementations already accept resumption across address changes; the note is included for implementations that currently check peer IP.

* *(Advisory, MAY) Firewall session reconciliation.* Stateful firewalls MAY use a client-identity tag (TLS channel ID, QUIC connection ID, or application cookie) as an auxiliary state key so that return traffic on a different source address is matched to the existing forward-flow state. This is advisory because the firewall must be in the application's trust domain to read such identifiers.

Deliberate non-resolution. The normative/advisory split above is deliberate: mitigations that a single operator can deploy without cooperation from clients, DNS, or other operators are normative (SHOULD); mitigations that require cross-party coordination or depend on application semantics outside the firewall/LB boundary are advisory (MAY). This distinction is not a design escape hatch; it reflects what a deploying operator can actually control. Applications that cannot tolerate multi-address semantics at all SHOULD pin the client to a single address in DNS-ASIP (publish only one A-ASIP record per client) at the cost of losing multihoming failover — this is a deployment choice, not a protocol defect.

18.8. 18.6. WHOIS-ASIP Adoption

WHOIS-ASIP route validation provides meaningful security only if widely adopted. An eBGP-ASIP router that enforces WHOIS-ASIP validation in a network where most peers do not publish WHOIS-ASIP records will reject legitimate routes. The same chicken-and-egg problem that limited RPKI deployment applies here. The simplification from one-route-per-ASN reduces but does not eliminate this problem.

18.9. 18.7. 32-Bit Host Field and SLAAC Constraints

The 32-bit host field (h.h.h.h) is substantially smaller than IPv6's 64-bit interface identifier. This limits SLAAC-ASIP to methods that produce 32-bit identifiers rather than the full EUI-64 derivation available in IPv6. Birthday-paradox analysis (§3.14.3) gives approximately 39% collision probability at N=65,536 and 50% at N~77,000; the 10,000-host DHCP-ASIP-fallback threshold sits well below this point (~1% collision probability) to give DAD retries a large safety margin. Modern datacenter leaf-subnet populations routinely approach or exceed 10,000, so the threshold is operationally material: the collision region is reached by real fabrics, not by theoretical corner cases. Section 3.14.3 specifies the consequence: deployments expected to exceed ~10,000 concurrent hosts per subnet MUST use DHCP-ASIP rather than relying on SLAAC-ASIP uniqueness, and DAD retries are bounded to prevent indefinite re-rolling.

18.10. 18.8. Interplanetary Address Reservation

Reserving large ASN ranges for celestial bodies that currently have zero networked devices may appear wasteful. The counter-argument is that address space reservation costs nothing today but prevents painful renumbering later. IPv4's failure to reserve space for mobile devices, IoT, and cloud infrastructure before they existed is a cautionary example. The reservations are deliberately generous and can always be narrowed later; they cannot easily be widened once allocated.

18.11. 18.9. Complexity Budget

ASIP is more complex than either IPv4 or IPv6 individually. It combines IPv4's addressing familiarity, IPv6's header and extension mechanism, a new hierarchical address structure, SLAAC, scoped multicast, and forward-looking realm allocations. Each feature is independently justified, but the aggregate complexity is a real adoption barrier. Implementors must understand three protocol generations to build a compliant stack. This specification attempts to mitigate complexity by making most features OPTIONAL or RECOMMENDED rather than REQUIRED, but the specification itself is long. A novel routing metric (the OPTIONAL Cost Factor, § 17) is deliberately excised to a companion draft so that the ASIP address-family specification is not burdened with a metric it does not require.

19. References

19.1. 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/rfc/rfc2119>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/rfc/rfc2827>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/rfc/rfc3704>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/rfc/rfc4271>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/rfc/rfc4443>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/rfc/rfc4760>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/rfc/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/rfc/rfc4862>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/rfc/rfc6437>.

[RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <https://www.rfc-editor.org/rfc/rfc7112>.

[RFC7607] Kumari, W., Bush, R., Schiller, H., and K. Patel, "Codification of AS 0 Processing", RFC 7607, DOI 10.17487/RFC7607, August 2015, <https://www.rfc-editor.org/rfc/rfc7607>.

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <https://www.rfc-editor.org/rfc/rfc7915>.

[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/rfc/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/rfc/rfc8200>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/rfc/rfc8201>.

[RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <https://www.rfc-editor.org/rfc/rfc8926>.

19.2. Informative References

[I-D.thain-ipv8] Thain, J., "Internet Protocol Version 8 (IPv8)", Work in Progress, Internet-Draft, draft-thain-ipv8-00, April 2026, <https://datatracker.ietf.org/doc/draft-thain-ipv8/>.

[RFC1624] Rijsinghani, A., Ed., "Computation of the Internet Checksum via Incremental Update", RFC 1624, DOI 10.17487/RFC1624, May 1994, <https://www.rfc-editor.org/rfc/rfc1624>.

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <https://www.rfc-editor.org/rfc/rfc1812>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/rfc/rfc1918>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/rfc/rfc2328>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/rfc/rfc3810>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/rfc/rfc4291>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/rfc/rfc4301>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/rfc/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/rfc/rfc4303>.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, DOI 10.17487/RFC4459, April 2006, <https://www.rfc-editor.org/rfc/rfc4459>.

[RFC5398] Huston, G., "Autonomous System (AS) Number Reservation for Documentation Use", RFC 5398, DOI 10.17487/RFC5398, December 2008, <https://www.rfc-editor.org/rfc/rfc5398>.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <https://www.rfc-editor.org/rfc/rfc5927>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/rfc/rfc5952>.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <https://www.rfc-editor.org/rfc/rfc6105>.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <https://www.rfc-editor.org/rfc/rfc6480>.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <https://www.rfc-editor.org/rfc/rfc6996>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/rfc/rfc7045>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/rfc/rfc7217>.

[RFC7346] Droms, R., "IPv6 Multicast Address Scopes", RFC 7346, DOI 10.17487/RFC7346, August 2014, <https://www.rfc-editor.org/rfc/rfc7346>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.

[RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/rfc/rfc792>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/rfc/rfc8305>.

[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/rfc/rfc8446>.

[RFC8899] Fairhurst, G., Jones, T., Txen, M., Rngeler, I., and T. Vlker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/rfc/rfc8899>.

[RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <https://www.rfc-editor.org/rfc/rfc8981>.

[RFC9171] Burleigh, S., Fall, K., and E. Birrane, III, "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <https://www.rfc-editor.org/rfc/rfc9171>.

[RFC9172] Birrane, III, E. and K. McKeever, "Bundle Protocol Security (BPSec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <https://www.rfc-editor.org/rfc/rfc9172>.

Author’s Address

Jordan Hause
Independent
Email: truixprojects@gmail.com