

---

Stream: Internet Engineering Task Force (IETF)  
RFC: [9758](#)  
Updates: [6260](#), [7116](#), [9171](#)  
Category: Standards Track  
Published: March 2025  
ISSN: 2070-1721  
Authors: R. Taylor E. Birrane  
*Aalyria Technologies* *JHU/APL*

# RFC 9758

## Updates to the ipn URI Scheme

---

### Abstract

This document updates the specification of the ipn URI scheme previously defined in RFC 6260 and the IANA registries established in RFC 7116. It also updates the rules for the encoding and decoding of these URIs when used as an Endpoint Identifier (EID) in the Bundle Protocol version 7 (BPv7) as defined in RFC 9171. These updates clarify the structure and behavior of the ipn URI scheme, define new encodings of ipn scheme URIs, and establish the registries necessary to manage this scheme.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9758>.

### Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction	4
2. Conventions and Definitions	5
3. Core Concepts	5
3.1. The Null ipn URI	5
3.2. Allocator Identifiers	6
3.2.1. Allocator Identifier Ranges	6
3.2.2. The Default Allocator	7
3.3. Node Numbers	8
3.3.1. Fully-Qualified Node Numbers	8
3.4. Special Node Numbers	8
3.4.1. The Zero Node Number	8
3.4.2. LocalNode ipn URIs	9
3.4.3. Private Use Node Numbers	9
3.5. Service Numbers	9
4. Textual Representation of ipn URIs	9
4.1. ipn URI Scheme Text Syntax	10
5. Usage of ipn URIs with BPv7	10
5.1. Uniqueness Constraints	11
5.2. The Null Endpoint	11
5.3. BPv7 Node ID	11
5.4. LocalNode ipn EIDs	11
5.5. Private Use ipn EIDs	12
5.6. Well-Known Service Numbers	12
5.7. Administrative Endpoints	12
6. CBOR Representation of ipn URIs with BPv7	13
6.1. ipn EID CBOR Encoding	13
6.1.1. Two-Element Scheme-Specific Encoding	13

---

6.1.2. Three-Element Scheme-Specific Encoding	14
6.2. ipn EID CBOR Decoding	14
6.3. ipn URI Scheme CBOR Syntax	15
6.4. ipn EID Matching	15
7. Special Considerations	16
7.1. Scheme Compatibility	16
7.2. CBOR Representation Interoperability	16
7.3. Text Representation Compatibility	17
7.4. Bundle Protocol Version 6 Compatibility	17
7.5. Late Binding	17
8. Security Considerations	18
8.1. Reliability and Consistency	18
8.2. Malicious Construction	18
8.3. Back-End Transcoding	18
8.4. Local and Private Use ipn EIDs	18
8.5. Sensitive Information	19
8.6. Semantic Attacks	19
9. IANA Considerations	19
9.1. 'ipn' Scheme URI Allocator Identifiers Registry	19
9.1.1. Guidance for Designated Experts	20
9.2. 'ipn' Scheme URI Default Allocator Node Numbers Registry	20
9.3. 'ipn' Scheme URI Well-Known Service Numbers for BPv7 Registry	21
9.3.1. Guidance for Designated Experts	22
10. References	22
10.1. Normative References	22
10.2. Informative References	23
Appendix A. ipn URI Scheme Text Representation Examples	23
A.1. Using the Default Allocator	24
A.2. Using a Non-Default Allocator	24
A.3. The Null ipn URI	24

---

<a href="#">A.4. The LocalNode ipn URI</a>	24
<a href="#">Appendix B. ipn URI Scheme CBOR Encoding Examples</a>	25
<a href="#">B.1. Using the Default Allocator</a>	25
<a href="#">B.2. Using a Non-Default Allocator</a>	25
<a href="#">B.3. The 'null' Endpoint</a>	26
<a href="#">Acknowledgments</a>	26
<a href="#">Authors' Addresses</a>	26

## 1. Introduction

The ipn URI scheme was originally defined in [RFC6260] and [RFC7116] as a way to identify network nodes and node services using concisely encoded integers that can be processed faster and with fewer resources than other verbose identifier schemes. The scheme was designed for use with the experimental Bundle Protocol version 6 (BPv6) [RFC5050], and "IPN" was defined as an acronym for the term "InterPlanetary Network" in reference to its intended use for deep-space networking. Since then, the efficiency benefits of integer identifiers make ipn scheme URIs useful for any network operating with limited power, bandwidth, and/or compute budget. Therefore, the term "IPN" is now used as a non-acronymous name.

Similar to the experimental BPv6, the standardized Bundle Protocol version 7 (BPv7) [RFC9171] codifies support for the use of the ipn URI scheme for the specification of bundle Endpoint Identifiers (EIDs). The publication of BPv7 has resulted in operational deployments of BPv7 nodes for both terrestrial and non-terrestrial use cases. This includes BPv7 networks operating over the terrestrial Internet and BPv7 networks operating in self-contained environments behind a shared administrative domain. The growth in the number and scale of deployments of BPv7 has been accompanied by a growth in the usage of the ipn URI scheme, which has highlighted areas to improve the structure, moderation, and management of this scheme.

By updating [RFC7116] and [RFC9171], this document updates the specification of the ipn URI scheme in a backwards-compatible way, in order to provide needed improvements both in the scheme itself and in its usage to specify EIDs with BPv7. Specifically, this document:

- introduces a hierarchical structure for the assignment of ipn scheme URIs,
- clarifies the behavior and interpretation of ipn scheme URIs,
- defines efficient encodings of ipn scheme URIs, and
- updates/defines the registries associated with this scheme.

Although originally developed by the deep-space community for use with the Bundle Protocol, the ipn URI scheme is sufficiently generic to be used in other environments where a concise unique representation of a resource on a particular node is required.

It is important to remember that, like most other URI schemes, the ipn URI scheme defines a unique identifier of a resource, and it does not include any topological information describing how to route messages to that resource.

## 2. Conventions and Definitions

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.

For the remainder of this document, the term "ipn URI" is used to refer to a URI that uses the ipn scheme.

## 3. Core Concepts

Every ipn URI, no matter whether it is expressed with a textual representation or a binary encoding, **MUST** be considered as a tuple of the following three components:

- The Allocator Identifier
- The Node Number
- The Service Number

The Allocator Identifier indicates the entity responsible for assigning Node Numbers to individual resource nodes, maintaining uniqueness whilst avoiding the need for a single registry for all assigned Node Numbers. See "[Allocator Identifiers](#)" ([Section 3.2](#)).

The Node Number is a shared identifier assigned to all ipn URIs for resources co-located on a single node. See "[Node Numbers](#)" ([Section 3.3](#)).

The Service Number is an identifier to distinguish between resources on a given node. See "[Service Numbers](#)" ([Section 3.5](#)).

The combination of these three components guarantees that every correctly constructed ipn URI uniquely identifies a single resource. Additionally, the combination of the Allocator Identifier and the Node Number provides a mechanism to uniquely identify the node on which a particular resource is expected to exist. See "[Fully-Qualified Node Numbers](#)" ([Section 3.3.1](#)).

### 3.1. The Null ipn URI

It has been found that there is value in defining a unique 'null' ipn URI to indicate "nowhere". This ipn URI is termed the "Null ipn URI" and has all three components (the Allocator Identifier, Node Number, and Service Number) set to the value zero (0). No resource identified by the Null ipn URI exists, and any destination identified by such a resource is therefore by definition unreachable.

## 3.2. Allocator Identifiers

An Allocator is any organization that wishes to assign Node Numbers for use with the ipn URI scheme and has the facilities and governance to manage a public registry of assigned Node Numbers. The authorization to assign these numbers is provided through the assignment of an Allocator Identifier by IANA. Regardless of other attributes of an Allocator (such as a name, point of contact, or other identifying information), Allocators are identified by Allocator Identifiers: unique, unsigned integers in the range 0 to  $2^{32}-1$ .

The Allocator Identifier **MUST** be the sole mechanism used to identify the Allocator that has assigned the Node Number in an ipn URI. An Allocator may have multiple assigned Allocator Identifiers, but a given Allocator Identifier **MUST** only be associated with a single Allocator.

A new IANA registry, "'ipn' Scheme URI Allocator Identifiers", is defined for the registration of Allocator Identifiers; see "['ipn' Scheme URI Allocator Identifiers Registry](#)" ([Section 9.1](#)). Although the uniqueness of Allocator Identifiers is required to enforce the uniqueness of ipn URIs, some identifiers are explicitly reserved for experimentation or future use.

Each Allocator assigns Node Numbers according to its own policies, without risk of creating an identical ipn URI, as permitted by the rules in "[Node Numbers](#)" ([Section 3.3](#)). Other than ensuring that any Node Numbers it allocates are unique amongst all Node Numbers it assigns, an Allocator does not need to coordinate its allocations with other Allocators.

If a system does not require interoperable deployment of ipn scheme URIs, then the [Private Use Node Numbers](#) ([Section 3.4.3](#)) range, reserved by the [Default Allocator](#) ([Section 3.2.2](#)) for this purpose, are to be used.

### 3.2.1. Allocator Identifier Ranges

Some organizations with internal hierarchies may wish to delegate the allocation of Node Numbers to one or more of their sub-organizations. Rather than assigning unique Allocator Identifiers to each sub-organization on a first-come, first-served basis, there are operational benefits in assigning Allocator Identifiers to such an organization in a structured way. This allows an external observer to detect that a group of Allocator Identifiers is organizationally associated.

An Allocator Identifier range is a set of consecutive Allocator Identifiers associated with the same Allocator. Each individual Allocator Identifier in a given range **SHOULD** be assigned to a distinct sub-organization of the Allocator. Assigning identifiers in this way allows external observers to both associate individual Allocator Identifiers with a single organization and usefully differentiate amongst sub-organizations.

The practice of associating a consecutive range of numbers with a single organization is inspired by the Classless Inter-Domain Routing (CIDR) assignment of Internet addresses described in [[RFC4632](#)]. In that assignment scheme, an organization (such as an Internet Service Provider (ISP)) is assigned a network prefix such that all addresses sharing that same prefix are considered to be associated with that organization.

Each Allocator Identifier range is identified by the first Allocator Identifier in the range and the number of consecutive identifiers in the range.

Allocator Identifier ranges differ from CIDR addresses in two important ways:

1. Allocator Identifiers are used to identify organizations and are not, themselves, addresses.
2. Allocator Identifiers may be less than 32 bits in length.

In order to differentiate between Allocator Identifier ranges using efficient bitwise operations, all ranges **MUST** be of a size  $S$  that is a power of 2, and for a given range of length  $N$  bits, with  $S = 2^N$ , the least-significant  $N$  bits of the first Allocator Identifier **MUST** be all 0.

An example of the use of Allocator Identifier ranges for four organizations (A, B, C, and D) is as follows:

Organization	Range (dec)	Range (hex)	Range Length (Bits)
Org A	974848 .. 974975	0xEE000 .. 0xEE07F	7 bits
Org B	974976 .. 974991	0xEE080 .. 0xEE08F	4 bits
Org C	974992 .. 974993	0xEE090 .. 0xEE091	1 bit
Org D	974994	0xEE092	0 bits

*Table 1: Allocator Identifier Range Assignment Example*

With these assignments, any Allocator Identifier whose most-significant 25 bits match 0xEE000 belong to organization A. Similarly, any Allocator Identifier whose most-significant 28 bits match 0xEE080 belong to organization B, and any Allocator Identifier whose most-significant 31 bits are 0xEE090 belong to organization C. Organization D has a single Allocator Identifier, and hence a range of bit-length 0.

### 3.2.2. The Default Allocator

As of the publication of [RFC7116], the only organization permitted to assign Node Numbers was the Internet Assigned Numbers Authority (IANA), which assigned Node Numbers via the "CBHE Node Numbers" registry. This means that all ipn URIs created prior to the addition of Allocator Identifiers are assumed to have Node Number allocations that comply with the "CBHE Node Numbers" registry.

The presumption that Node Numbers are allocated by IANA from a specific registry, unless otherwise specified, is formalized in this update to the ipn URI scheme by designating IANA as the Default Allocator and by assigning the Allocator Identifier zero (0) in the "'ipn' Scheme URI Allocator Identifiers" registry (Section 9.1) to the Default Allocator. In any case where an encoded ipn URI does not explicitly include an Allocator Identifier, an implementation **MUST** assume that the Node Number has been allocated by the Default Allocator.

A new IANA registry, "'ipn' Scheme URI Default Allocator Node Numbers", is defined to control the allocation of Node Number values by the Default Allocator. This new registry inherits behaviors and existing assignments from the "CBHE Node Numbers" registry, and it reserves some other values as defined in "[Special Node Numbers](#)" ([Section 3.4](#)) below.

### 3.3. Node Numbers

A Node Number identifies a node that hosts a resource in the context of an Allocator. A Node Number is an unsigned integer. A single Node Number assigned by a single Allocator **MUST** refer to a single node.

All Node Number assignments, by all Allocators, **MUST** be in the range 0 to  $2^{32}-1$ .

It is **RECOMMENDED** that Node Number zero (0) not be assigned by an Allocator to avoid confusion with the [Null ipn URI](#) ([Section 3.1](#)).

#### 3.3.1. Fully-Qualified Node Numbers

One of the advantages of ipn URIs is the ability to easily split the identity of a particular service from the node upon which the service exists. For example, a message received from one particular ipn URI may require a response to be sent to a different service on the same node that sent the original message. Historically, the identifier of the sending node has been colloquially referred to as the "node number" or "node identifier".

To avoid future confusion, when referring to the identifier of a particular node, the term "Fully-Qualified Node Number" (FQNN) **MUST** be used to refer to the combination of the Node Number component and Allocator Identifier component of an ipn URI that uniquely identifies a particular node. In other words, an FQNN is the unique identifier of a particular node that supports services identified by ipn URIs.

In the examples in this document, FQNNs are written as (Allocator Identifier, Node Number). For example, (977000, 100) is the FQNN for a node assigned Node Number 100 by an Allocator with Allocator Identifier 977000.

### 3.4. Special Node Numbers

Some special-case Node Numbers are defined by the Default Allocator; see "['ipn' Scheme URI Default Allocator Node Numbers Registry](#)" ([Section 9.2](#)).

#### 3.4.1. The Zero Node Number

The Default Allocator assigns the use of Node Number zero (0) solely for identifying the [Null ipn URI](#) ([Section 3.1](#)).

This means that any ipn URI with a zero (0) Allocator Identifier and a zero (0) Node Number, but a non-zero Service Number component, is invalid. Such ipn URIs **MUST NOT** be composed, and processors of such ipn URIs **MUST** consider them as the Null ipn URI.

### 3.4.2. LocalNode ipn URIs

The Default Allocator reserves Node Number  $2^{32-1}$  (0xFFFFFFFF) to specify resources on the local node, rather than on any specific individual node.

This means that any ipn URI with a zero (0) Allocator Identifier and a Node Number of  $2^{32-1}$  refers to a service on the local bundle node. This form of ipn URI is termed a "LocalNode ipn URI".

### 3.4.3. Private Use Node Numbers

The Default Allocator provides a range of Node Numbers that are reserved for Private Use, as defined in [RFC8126].

Any ipn URI with a zero (0) Allocator Identifier and a Node Number reserved for Private Use is not guaranteed to be unique beyond a single administrative domain. An administrative domain, as used here, is defined as the set of nodes that share a unique allocation of FQNNs from the Private Use range. These FQNNs can be considered to be functionally similar to private address space IPv4 addresses, as defined in [RFC1918].

Because of this lack of uniqueness, any implementation of a protocol using ipn URIs that resides on the border between administrative domains **MUST** have suitable mechanisms in place to prevent protocol units using such Private Use Node Numbers to cross between different administrative domains.

## 3.5. Service Numbers

A Service Number is an unsigned integer that identifies a particular service operating on a node. A service in this case is some logical function that requires its own resource identifier to distinguish it from other functions operating on the same node.

## 4. Textual Representation of ipn URIs

All ipn scheme URIs comply with [RFC3986] and are therefore represented by a scheme identifier and a scheme-specific part. The scheme identifier is `ipn`, and the scheme-specific parts are represented as a sequence of numeric components separated with the `'.'` character. A formal definition is provided below (see "[ipn URI Scheme Text Syntax](#)" (Section 4.1)) and can be informally considered as:

```
ipn:[<allocator-identifier>.<node-number>.<service-number>
```

To keep the text representation concise, the following rules apply:

1. All leading `0` characters **MUST** be omitted. A single `'0'` is valid.
2. If the Allocator Identifier is zero (0), then the `<allocator-identifier>` and `'.'` **MAY** be omitted.

3. If the Allocator Identifier is zero (0), and the Node Number is  $2^{32-1}$  (i.e., the URI is a [LocalNode ipn URI \(Section 3.4.2\)](#)), then the character '!' **SHOULD** be used instead of the digits 4294967295, although both forms are valid encodings.

Examples of the textual representation of ipn URIs can be found in [Appendix A](#).

#### 4.1. ipn URI Scheme Text Syntax

The text syntax of an ipn URI **MUST** comply with the following ABNF syntax from [\[RFC5234\]](#) and reiterate the core ABNF syntax rule for DIGIT defined by that specification:

```
ipn-uri = "ipn:" ipn-hier-part
ipn-hier-part = fqnn "." service-number
fqnn = "!" / allocator-part
allocator-part = [allocator-identifier "."] node-number
allocator-identifier = number
node-number = number
service-number = number
number = "0" / non-zero-number
non-zero-number = (%x31-39 *DIGIT)
DIGIT = %x30-39
```

## 5. Usage of ipn URIs with BPv7

From the earliest days of experimentation with the Bundle Protocol, there has been a need to identify the source and destination of a bundle. The IRTF BPv6 experimental specification termed the logical source or destination of a bundle as an "Endpoint" identified by an "Endpoint Identifier" (EID). BPv6 EIDs are formatted as URIs. This definition and representation of EIDs was carried forward from the IRTF BPv6 specification to the IETF BPv7 specification. BPv7 additionally defined an IANA registry called the "Bundle Protocol URI Scheme Types" registry, which identifies those URI schemes that might be used to represent EIDs. The ipn URI scheme is one such URI scheme.

This section identifies the behavior and interpretation of ipn scheme URIs that **MUST** be followed when using this URI scheme to represent EIDs in BPv7. An ipn URI used as a BPv7 or BPv6 EID is termed an "ipn EID".

## 5.1. Uniqueness Constraints

An ipn EID **MUST** identify a singleton endpoint. The bundle processing node that is the sole member of that endpoint **MUST** be the node identified by the [Fully-Qualified Node Number \(Section 3.3.1\)](#) of the node.

A single bundle processing node **MAY** have multiple ipn EIDs associated with it. However, all ipn EIDs that share any single FQNN **MUST** refer to the same bundle processing node.

For example, `ipn:977000.100.1`, `ipn:977000.100.2`, and `ipn:977000.100.3` **MUST** all refer to services registered on the bundle processing node identified with FQNN (977000, 100). None of these EIDs could be registered on any other bundle processing node.

## 5.2. The Null Endpoint

[Section 3.2](#) of [\[RFC9171\]](#) defines the concept of the 'null' endpoint, which is an endpoint that has no members and is identified by a special 'null' EID.

Within the ipn URI scheme, the 'null' EID is represented by the [Null ipn URI \(Section 3.1\)](#). This means that the URIs `ipn:none` ([Section 4.2.5.1.1](#) of [\[RFC9171\]](#)), `ipn:0.0`, and `ipn:0.0.0` all refer to the BPv7 'null' endpoint.

## 5.3. BPv7 Node ID

[Section 4.2.5.2](#) of [\[RFC9171\]](#) introduces the concept of a "Node ID" that has the same format as an EID and uniquely identifies a bundle processing node.

Any ipn EID can serve as a "Node ID" for the bundle processing node identified by its [Fully-Qualified Node Number \(Section 3.3.1\)](#). That is, any ipn EID of the form `ipn:A.B.C` may be used as the Source Node ID of any bundle created by the bundle processing node identified by the FQNN (A, B).

## 5.4. LocalNode ipn EIDs

When a [LocalNode ipn URI \(Section 3.4.2\)](#) is used as a BPv6 or BPv7 EID, it is termed a "LocalNode ipn EID".

Because a LocalNode ipn EID only has meaning on the local bundle node, any such EID **MUST** be considered non-routable. This means that any bundle using a LocalNode ipn EID as a bundle source or bundle destination **MUST NOT** be allowed to leave the local node. Equally, all externally received bundles featuring LocalNode EIDs as a bundle source or bundle destination **MUST** be discarded as invalid.

LocalNode ipn EIDs **MUST NOT** be present in any other part of a bundle that is transmitted off of the local node. For example, a LocalNode ipn EID **MUST NOT** be used as a Bundle Protocol Security (BPSec) [\[RFC9172\]](#) security source for a bundle transmitted from the local bundle node, because such a source EID would have no meaning at a downstream bundle node.

LocalNode ipn EIDs **MUST NOT** be published in any node identification directory (such as a DNS registration) or presented as part of dynamic peer discovery, as the EID has no valid meaning for other nodes. For example, a LocalNode ipn EID **MUST NOT** be advertised as the peer Node ID during session negotiation in [RFC9174].

## 5.5. Private Use ipn EIDs

Bundles destined for EIDs that use an ipn URI with a [Fully-Qualified Node Number](#) (Section 3.3.1) that is within the Private Use range of the [Default Allocator](#) (Section 3.2.2) are not universally unique; therefore, they are only valid within the scope of the current administrative domain. This means that any bundle using a Private Use ipn EID as a bundle source or bundle destination **MUST NOT** be allowed to cross administrative domains. All implementations that could be deployed as a gateway between administrative domains **MUST** be sufficiently configurable to ensure that this is enforced, and operators **MUST** ensure correct configuration.

Private Use ipn EIDs **MUST NOT** be present in any other part of a bundle that is destined for another administrative domain when the lack of uniqueness prevents correct operation. For example, a Private Use ipn EID **MUST NOT** be used as a BPSec [RFC9172] security source for a bundle when the bundle is destined for a different administrative domain.

## 5.6. Well-Known Service Numbers

It is convenient for BPv7 services that have a public specification and wide adoption to be identified by a pre-agreed default Service Number, so that unless extra configurations are applied, such services can be sensibly assumed to be operating on the well-known Service Number on a particular node.

If a different service uses the number, or the service uses a different number, BPv7 will continue to operate, but some configuration may be required to make the individual service operational.

A new IANA registry, "'ipn' Scheme URI Well-Known Service Numbers for BPv7", is defined for the registration of well-known BPv7 Service Numbers; see ["'ipn' Scheme URI Well-Known Service Numbers for BPv7 Registry"](#) (Section 9.3). This registry records the assignments of Service Numbers for well-known services and also explicitly reserves ranges for both experimentation and Private Use.

## 5.7. Administrative Endpoints

The service identified by a Service Number of zero (0) **MUST** be interpreted as the Administrative Endpoint of the node, as defined in Section 3.2 of [RFC9171].

Non-zero Service Numbers **MUST NOT** be used to identify the Administrative Endpoint of a bundle node in an ipn EID.

## 6. CBOR Representation of ipn URIs with BPv7

Section 4.2.5.1 of [RFC9171] requires that any URI scheme used to represent BPv7 EIDs **MUST** define how the scheme-specific part of the URI scheme is encoded with Concise Binary Object Representation (CBOR) [RFC8949]. To meet this requirement, this section describes the CBOR encoding and decoding approach for ipn EIDs. The formal definition of the CBOR representation is specified; see "ipn URI Scheme CBOR Syntax" (Section 6.3).

### 6.1. ipn EID CBOR Encoding

Generic URI approaches to encoding ipn EIDs are unlikely to be efficient because they do not consider the underlying structure of the ipn URI scheme. Since the creation of the ipn URI scheme was motivated by the need for concise identification and rapid processing, the encoding of ipn EIDs maintains these properties.

Fundamentally, ipn EIDs from [RFC9171] are represented as a sequence of identifiers. In the text syntax, the numbers are separated with the '.' delimiter; in CBOR, this ordered series of numbers can be represented by an array. Therefore, when encoding ipn EIDs for use with BPv7, the scheme-specific part of an ipn URI **MUST** be represented as a CBOR array of either two (2) or three (3) elements. Each element of the array **MUST** be encoded as a single CBOR unsigned integer.

The structure and mechanisms of the two-element and three-element encodings are described below, and examples of the different encodings are provided in [Appendix B](#).

#### 6.1.1. Two-Element Scheme-Specific Encoding

In the two-element scheme-specific encoding of an ipn EID, the first element of the array is an encoding of the [Fully-Qualified Node Number](#) (Section 3.3.1), and the second element of the array is the ipn EID Service Number.

The FQNN encoding **MUST** be a 64-bit unsigned integer constructed in the following way:

1. The least significant 32 bits **MUST** represent the Node Number associated with the ipn EID.
2. The most significant 32 bits **MUST** represent the Allocator Identifier associated with the ipn EID.

For example, the ipn EID of ipn:977000.100.1 has an FQNN of (977000, 100), which would be encoded as 0xEE868\_00000064. The resulting two-element array [0xEE868\_00000064, 0x01] would be encoded in CBOR as the following 11-octet sequence:

```

82          # 2-Element Endpoint Encoding
 02        # uri-code: 2 (IPN URI scheme)
 82        # 2 Element ipn EID scheme-specific encoding
    1B 000EE86800000064 # Fully-Qualified Node Number
    01     # Service Number

```

The two-element scheme-specific encoding provides backwards compatibility with the encoding provided in [Section 4.2.5.1.2](#) of [\[RFC9171\]](#). When used in this way, the encoding of the FQNN replaces the use of the Node Number that was specified in [\[RFC9171\]](#). When the Node Number is allocated by the [Default Allocator](#) ([Section 3.2.2](#)), the encoding of the FQNN and the encoding of the Node Number from [\[RFC9171\]](#) are identical.

### 6.1.2. Three-Element Scheme-Specific Encoding

In the three-element scheme-specific encoding of an ipn EID:

1. the first element of the array is the Allocator Identifier,
2. the second element of the array is the Node Number, and
3. the third element of the array is the Service Number.

For example, the ipn EID of `ipn:977000.100.1` would result in the three-element array of `[977000, 100, 1]`, which would be encoded in CBOR as the following 9-octet sequence:

```
82          # 2-Element Endpoint Encoding
02          # uri-code: 2 (IPN URI scheme)
83          # 3 Element ipn EID scheme-specific encoding
  1A 000EE868 # Allocator Identifier
  64          # Node Number
  01          # Service Number
```

The three-element scheme-specific encoding allows for a more efficient representation of ipn EIDs using smaller Allocator Identifiers, and implementations are **RECOMMENDED** to use this encoding scheme unless explicitly mitigating for interoperability issues; see "[Scheme Compatibility](#)" ([Section 7.1](#)).

When encoding an ipn EID using the [Default Allocator](#) ([Section 3.2.2](#)) with this encoding scheme, the first element of the array is the value zero (0). In this case, using the equivalent [two-element scheme-specific encoding](#) ([Section 6.1.1](#)) will result in a more concise CBOR representation; therefore, it is **RECOMMENDED** that implementations use that encoding instead.

## 6.2. ipn EID CBOR Decoding

The presence of different scheme-specific encodings does not introduce any decoding ambiguity.

An ipn EID CBOR decoder can reconstruct an ipn EID using the following logic. In this description, the term `enc_eid` refers to the CBOR-encoded ipn EID, and the term `ipn_eid` refers to the decoded ipn EID.

```
if enc_eid.len() == 3
{
  ipn_eid allocator_identifier := enc_eid[0];
  ipn_eid node_number := enc_eid[1];
  ipn_eid service_number := enc_eid[2];
}
else if enc_eid.len() == 2
{
  ipn_eid allocator_identifier := enc_eid[0] >> 32;
  ipn_eid node_number := enc_eid[0] & (2^32-1);
  ipn_eid service_number := enc_eid[1];
}
```

### 6.3. ipn URI Scheme CBOR Syntax

When encoded in CBOR [RFC8949], a BPv7 endpoint identified by an ipn URI **MUST** comply with the following Concise Data Definition Language (CDDL) [RFC8610] specification:

```
eid = $eid .within eid-structure

eid-structure = [
  uri-code: uint,
  SSP: any
]

; ... Syntax for other uri-code values defined in RFC 9171 ...

$eid /= [
  uri-code: 2,
  SSP: ipn-ssp2 / ipn-ssp3
]
ipn-ssp2 = [
  fqnn: uint, ; packed value
  service-number: uint
]
ipn-ssp3 = [
  allocator-identifier: uint .lt 4294967296,
  node-number: uint .lt 4294967296,
  service-number: uint
]
```

Note: The node-number component will be the numeric representation of the concatenation of the Allocator Identifier and Node Number when the two-element encoding scheme has been used.

### 6.4. ipn EID Matching

Regardless of whether the two-element or three-element scheme-specific encoding is used, ipn EID matching **MUST** be performed on the decoded EID information itself. Different encodings of the same ipn EID **MUST** be treated as equivalent when performing EID-specific functions.

For example, the ipn EID of `ipn:977000.100.1` can be represented as either the two-element encoding of `0x821B000EE8680000006401` or the three-element encoding of `0x831A000EE868186401`. While message integrity and other syntax-based checks may treat these values differently, any EID-based comparisons **MUST** treat these values the same: as representing the ipn EID `ipn:977000.100.1`.

## 7. Special Considerations

The ipn URI scheme provides a compact and hierarchical mechanism for identifying services on network nodes. There is a significant amount of utility in the ipn URI scheme approach to identification. However, implementers should take into consideration the following observations on the use of the ipn URI scheme, particularly in regard to interoperability with implementations that pre-date this specification.

### 7.1. Scheme Compatibility

The ipn scheme update that has been presented in this document preserves backwards compatibility with any ipn URI scheme going back to the provisional definition of the ipn scheme in the experimental specification "Compressed Bundle Header Encoding (CBHE)" [RFC6260] in 2011. This means that any ipn URI that was valid prior to the publication of this update remains a valid ipn URI.

Similarly, the [two-element scheme-specific encoding](#) (Section 6.1.1) is also backwards compatible with the encoding of ipn URIs provided in [RFC9171]. Any existing implementation compliant with [RFC9171] will produce an ipn URI encoding in compliance with this specification.

The introduction of optional non-default Allocator Identifiers and a three-element scheme-specific encoding does not make this ipn URI scheme update forwards compatible. Existing implementations for which support of this update is desired **MUST** be updated to be able to process non-default Allocator Identifiers and three-element scheme-specific encodings. It is **RECOMMENDED** that BPv7 implementations upgrade to process these new features to benefit from the scalability provided by Allocator Identifiers and the encoding efficiencies provided by the three-element encoding.

### 7.2. CBOR Representation Interoperability

Care must be taken when deploying implementations that default to using the three-element encoding in networks that include implementations that only support the two-element encoding [RFC9171]. Because the existing implementations will reject bundles that use the three-element encoding as malformed, correct forwarding of semantically valid bundles will fail. The used mitigation for this issue depends on the nature of the interoperability required by the deployment. Techniques can include:

- A configuration option indicating when an implementation must use the two-element encoding for all ipn EIDs when processing bundles destined to a given endpoint. This would be suitable when adding a newer implementation to a network of existing implementations.

- Selective bundle encapsulation, whereby bundles that are known to originate from implementations that do not support the three-element encoding are tunneled across regions of the network that require the three-element encoding. This would utilize specially configured "gateway nodes" to perform the tunnel encapsulation and decapsulation and would be suitable when joining an existing network to a larger network.

Techniques that do not mitigate the problem include:

- Heuristic determination of the correct encoding to use when responding to a bundle by examining the incoming bundle. It is not possible to determine whether the two-element encoding is required by the destination when composing a new bundle in response to the receipt of a bundle, such as a status report, because ipn EIDs assigned by the Default Allocator use the two-element encoding, whether or not the implementation supports the three-element encoding.
- Transcoding bundles at intermediate nodes. [RFC9171] requires the bundle primary block to be immutable, and even if ipn EIDs in the primary block do not require rewriting, other blocks including the payload block may include ipn EIDs of which the transcoding node is unaware. Additionally, bundle blocks may be covered by bundle security blocks or bundle integrity blocks [RFC9172], making them immutable.

### 7.3. Text Representation Compatibility

The textual representation of ipn URIs is not forwards compatible with [RFC9171]. Therefore, care must be taken when deploying implementations or tooling that use the textual representation of ipn URIs and support for non-default Allocator Identifiers is required. For example, Section 4.6 of [RFC9174] specifies that the session initialization message "...SHALL contain the UTF-8 encoded node ID of the entity that sent the SESS\_INIT message." In such cases, the considerations that apply to the use of the three-element CBOR encoding also apply to the text representation when a non-default Allocator Identifier is present.

### 7.4. Bundle Protocol Version 6 Compatibility

This document updates the use of ipn EIDs for BPv7; however, the ipn URI scheme was originally defined for use with BPv6. This document does not update any of the behaviors, wire-formats, or mechanisms of BPv6. Therefore, ipn EIDs with non-default Allocator Identifiers **MUST NOT** be used with BPv6, and the Allocator Identifier prefix **MUST** be omitted from any textual representation. It should be noted that BPv6 has no concept of LocalNode EIDs and will therefore treat such EIDs as routable.

### 7.5. Late Binding

[RFC9171] mandates the concept of the "late binding" of an EID, whereby the address of the destination of a bundle is resolved from its hop-by-hop identifier as it transits a BPv7 network. This per-hop binding of identifiers to addresses underlines the fact that EIDs are purely names and should not carry any implicit or explicit information concerning the current location or reachability of an identified node and service. This removes the need to rename a node as its location changes.

The concept of late binding is preserved in this ipn URI scheme. Elements of an ipn URI **MUST NOT** be regarded as carrying information relating to location, reachability, or other addressing/routing concerns.

An example of incorrect behavior would be to assume that a given Allocator assigns Node Numbers derived from link-layer addresses and to interpret the Node Number component of an ipn URI directly as a link-layer address. No matter the mechanism an Allocator uses for the assignment of Node Numbers, they remain just numbers, without additional meaning.

## 8. Security Considerations

This section updates the security considerations from [Section 4.2.5.1.2](#) of [RFC9171] to account for the inclusion of Allocator Identifiers in the ipn URI scheme when used with BPv7.

### 8.1. Reliability and Consistency

None of the BPv7 endpoints identified by ipn EIDs are guaranteed to be reachable at any time, and the identity of the processing entities operating on those endpoints is never guaranteed by the Bundle Protocol itself. Verification of the signature provided by the Block Integrity Block (BIB) targeting the bundle's primary block, as defined by "[Bundle Protocol Security \(BPsec\)](#)" [RFC9172], is required for this purpose.

### 8.2. Malicious Construction

Malicious construction of a conformant ipn URI is limited to the malicious selection of Allocator Identifiers, Node Numbers, and Service Numbers. That is, a maliciously constructed ipn EID could be used to direct a bundle to an endpoint that might be damaged by the arrival of that bundle or, alternatively, to declare a false source for a bundle and thereby cause incorrect processing at a node that receives the bundle. In both cases (and indeed in all bundle processing), the node that receives a bundle should verify its authenticity and validity before operating on it in any way, such as the use of BPsec [RFC9172] and TCP Convergence Layer version 4 (TCPCLv4) with TLS [RFC9174].

### 8.3. Back-End Transcoding

The limited expressiveness of URIs of the ipn scheme effectively eliminates the possibility of threats due to errors in back-end transcoding.

### 8.4. Local and Private Use ipn EIDs

Both [LocalNode](#) ([Section 3.4.2](#)) and [Private Use](#) ([Section 3.4.3](#)) ipn URIs present a risk to the stability of deployed BPv7 networks. If either type of ipn URI is allowed to propagate beyond the domain in which they are valid, then the required uniqueness of ipn URIs no longer holds, and this fact can be abused by a malicious node to prevent the correct functioning of the network as a whole.

See "[LocalNode ipn EIDs](#)" ([Section 5.4](#)) and "[Private Use ipn EIDs](#)" ([Section 5.5](#)) for required behaviors to mitigate against this form of abuse.

## 8.5. Sensitive Information

Because ipn URIs are used only to represent the numeric identities of resources, the risk of disclosure of sensitive information due to interception of these URIs is minimal. Examination of ipn URIs could be used to support traffic analysis; where traffic analysis is a plausible danger, bundles should be conveyed by secure convergence-layer protocols that do not expose endpoint IDs, such as TCPCLv4 [[RFC9174](#)].

## 8.6. Semantic Attacks

The simplicity of the ipn URI scheme syntax minimizes the possibility of misinterpretation of a URI by a human user.

# 9. IANA Considerations

The following sections detail the creation of two new IANA registries and the renaming of an existing IANA registry under the "Uniform Resource Identifier (URI) Schemes" registry group.

IANA has also updated the reference for the 'ipn' scheme to this document in the "Uniform Resource Identifier (URI) Schemes" registry.

## 9.1. 'ipn' Scheme URI Allocator Identifiers Registry

IANA has created a new registry titled "'ipn' Scheme URI Allocator Identifiers". Using terms defined in [[RFC8126](#)], the registration policies for this registry are:

Range	Registration Procedures	Note
0..0xFFFF	Expert Review	Single Allocator Identifiers only
0x10000..0x3FFFFFFF	Expert Review	
0x40000000..0x7FFFFFFF	Experimental Use	
0x80000000..0xFFFFFFFF	Reserved	Future Expansion
$\geq 2^{32}$	Reserved	

*Table 2: Registration Policies for the 'ipn' Scheme URI Allocator Identifiers Registry*

Each entry in this registry associates one or more Allocator Identifiers with a single organization. Within the registry, the organization is identified using the "Name" and "Change Controller" fields. It is expected that each identified organization will publish some listing of allocated Node Numbers, the pointer to which is listed in the "Reference" field of the registry.

Note that the "Single Allocator Identifiers only" language in the registration policy for this registry indicates that, within the indicated range, the allocation of a sequence of consecutive Allocator Identifiers to a single organization is prohibited.

The initial values in the registry are:

Name	Range (dec)	Range (hex)	Range Length (Bits)	Reference	Change Controller
Default Allocator	0	0x0	0	RFC 9758, <a href="#">Section 3.2.2</a>	IETF
Example Range	974848-978943	0xEE000-0xEEFFF	12 bits	RFC 9758	IETF

Table 3: Initial Values in the 'ipn' Scheme URI Allocator Identifiers Registry

The "Example Range" is assigned for use in examples in documentation and sample code.

### 9.1.1. Guidance for Designated Experts

Due to the nature of the CBOR encoding of unsigned integers used for Allocator Identifiers with BPv7, Allocator Identifiers with a low value number are encoded more efficiently than larger numbers. This makes low value Allocator Identifiers more desirable than larger Allocator Identifiers; therefore, care must be taken when assigning Allocator Identifier ranges to ensure that a single applicant is not granted a large swathe of highly desirable numbers at the expense of other applicants. To this end, designated experts are strongly recommended to familiarize themselves with the CBOR encoding of unsigned integers in [\[RFC8949\]](#).

## 9.2. 'ipn' Scheme URI Default Allocator Node Numbers Registry

IANA has renamed the "CBHE Node Numbers" registry (defined in [Section 3.2.1](#) of [\[RFC7116\]](#)) to the "'ipn' Scheme URI Default Allocator Node Numbers" registry and moved it to the "Uniform Resource Identifier (URI) Schemes" registry group. IANA has added the following note to the "CBHE Node Numbers" registry:

Note: Renamed "CBHE Node Numbers" as "'ipn' Scheme URI Default Allocator Node Numbers" and moved it to <https://www.iana.org/assignments/uri-schemes> per RFC 9758.

Using terms defined in [\[RFC8126\]](#), the registration policies for this registry are:

Range	Registration Procedures
1..0x3FFF	Private Use

Range	Registration Procedures
0x4000..0xFFFFFFFF	Expert Review
$\geq 2^{32}$	Invalid

*Table 4: Registration Policies for the 'ipn' Scheme URI Default Allocator Node Numbers Registry*

IANA has registered the following values in the "'ipn' Scheme URI Default Allocator Node Numbers" registry:

Value	Description	Reference
0	Reserved for the Null ipn URI	[RFC7116] and RFC 9758, <a href="#">Section 3.1</a>
4294967295	Reserved for LocalNode ipn URIs	RFC 9758, <a href="#">Section 3.4.2</a>
$\geq 4294967296$	Invalid	RFC 9758

*Table 5: New Values in the 'ipn' Scheme URI Default Allocator Node Numbers Registry*

As IANA has only renamed the registry, all existing registrations will remain.

### 9.3. 'ipn' Scheme URI Well-Known Service Numbers for BPv7 Registry

IANA has created a new registry titled "'ipn' Scheme URI Well-Known Service Numbers for BPv7". Using terms defined in [RFC8126], the registration policies for this registry are:

Range	Registration Procedures
1..127	Private Use
128..255	Standards Action
0x0100..0x7FFF	Private Use
0x8000..0xFFFF	Specification Required
0x10000..0xFFFFFFFF	Private Use
$\geq 2^{32}$	Reserved for future expansion

*Table 6: Registration Policies for the 'ipn' Scheme URI Well-Known Service Numbers for BPv7 Registry*

The initial values for the registry are:

Value	Description	Reference
0	The Administrative Endpoint	[RFC9171] and RFC 9758, <a href="#">Section 5.7</a>
1..27	Reserved for Private Use	RFC 9758
0x0100..0x7FFF	Reserved for Private Use	RFC 9758
0xEEEE0..0xEEEF	Example Range	RFC 9758
0x10000..0xFFFFFFFF	Reserved for Private Use	RFC 9758
$\geq 2^{32}$	Reserved for future expansion	RFC 9758

Table 7: Initial Values in the 'ipn' Scheme URI Well-Known Service Numbers for BPv7 Registry

The "Example Range" is assigned for use in examples in documentation and sample code.

### 9.3.1. Guidance for Designated Experts

This registry is intended to record the default Service Numbers for well-known, interoperable services that are available and of use to the entire BPv7 community; hence, all ranges not marked for Private Use **MUST** have a corresponding publicly available specification describing how one interfaces with the service.

Services that are specific to a particular deployment or co-operation may require a registry to reduce administrative burden, but do not require an entry in this registry.

## 10. References

### 10.1. 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/info/rfc2119>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC6260] Burleigh, S., "Compressed Bundle Header Encoding (CBHE)", RFC 6260, DOI 10.17487/RFC6260, May 2011, <<https://www.rfc-editor.org/info/rfc6260>>.
- [RFC7116] Scott, K. and M. Blanchet, "Licklider Transmission Protocol (LTP), Compressed Bundle Header Encoding (CBHE), and Bundle Protocol IANA Registries", RFC 7116, DOI 10.17487/RFC7116, February 2014, <<https://www.rfc-editor.org/info/rfc7116>>.

- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/info/rfc8949>>.
- [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/info/rfc9171>>.

## 10.2. Informative References

- [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/info/rfc1918>>.
- [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/info/rfc3986>>.
- [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <<https://www.rfc-editor.org/info/rfc4632>>.
- [RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, DOI 10.17487/RFC5050, November 2007, <<https://www.rfc-editor.org/info/rfc5050>>.
- [RFC9172] Birrane, III, E. and K. McKeever, "Bundle Protocol Security (BPsec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <<https://www.rfc-editor.org/info/rfc9172>>.
- [RFC9174] Sipos, B., Demmer, M., Ott, J., and S. Perreault, "Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4", RFC 9174, DOI 10.17487/RFC9174, January 2022, <<https://www.rfc-editor.org/info/rfc9174>>.

## Appendix A. ipn URI Scheme Text Representation Examples

This section provides some example ipn URIs in their textual representation.

### A.1. Using the Default Allocator

Consider the ipn URI identifying Service Number 2 on Node Number 1 allocated by the [Default Allocator \(0\)](#) ([Section 3.2.2](#)).

The recommended seven-character representation of this URI would be as follows:

```
ipn:1.2
```

The nine-character representation of this URI, with the explicit Allocator Identifier, would be as follows:

```
ipn:0.1.2
```

### A.2. Using a Non-Default Allocator

Consider the ipn URI identifying Service Number 3 on Node Number 1 allocated by Allocator 977000.

The 14-character representation of this URI would be as follows:

```
ipn:977000.1.3
```

### A.3. The Null ipn URI

The [Null ipn URI](#) ([Section 3.1](#)) is represented as:

```
ipn:0.0
```

### A.4. The LocalNode ipn URI

Consider the ipn URI identifying Service Number 7 on the local node.

The recommended seven-character representation of this URI would be as follows:

```
ipn:!.7
```

The numeric 16-character representation of this URI would be as follows:

```
ipn:4294967295.7
```

## Appendix B. ipn URI Scheme CBOR Encoding Examples

This section provides some example CBOR encodings of ipn EIDs.

### B.1. Using the Default Allocator

Consider the ipn EID `ipn:1.1`. This textual representation of an ipn EID identifies Service Number 1 on Node Number 1 allocated by the [Default Allocator \(0\)](#) ([Section 3.2.2](#)).

The recommended five-octet encoding of this EID using the two-element scheme-specific encoding would be as follows:

```
82      # 2-Element Endpoint Encoding
  02    # uri-code: 2 (IPN URI scheme)
  82    # 2 Element ipn EID scheme-specific encoding
    01  # Node Number
    01  # Service Number
```

The six-octet encoding of this EID using the three-element scheme-specific encoding would be as follows:

```
82      # 2-Element Endpoint Encoding
  02    # uri-code: 2 (IPN URI scheme)
  83    # 3 Element ipn EID scheme-specific encoding
    00  # Default Allocator
    01  # Node Number
    01  # Service Number
```

### B.2. Using a Non-Default Allocator

Consider the ipn EID `ipn:977000.1.1`. This textual representation of an ipn EID identifies Service Number 1 on Node Number 1 allocated by Allocator 977000.

The recommended 10-octet encoding of this EID using the three-element scheme-specific encoding would be as follows:

```
82      # 2-Element Endpoint Encoding
  02    # uri-code: 2 (IPN URI scheme)
  83    # 3 Element ipn EID scheme-specific encoding
    1A 000EE868 # Allocator Identifier
    01    # Node Number
    01    # Service Number
```

The 13-octet encoding of this EID using the two-element scheme-specific encoding would be as follows:

```

82          # 2-Element Endpoint Encoding
02          # uri-code: 2 (IPN URI scheme)
82          # 2 Element ipn EID scheme-specific encoding
1B 000EE86800000001 # Fully-Qualified Node Number
01          # Service Number

```

### B.3. The 'null' Endpoint

The 'null' EID of ipn:0.0 can be encoded in the following ways:

The recommended five-octet encoding of the 'null' ipn EID using the two-element scheme-specific encoding would be as follows:

```

82          # 2-Element Endpoint Encoding
02          # uri-code: 2 (IPN URI scheme)
82          # 2 Element ipn EID scheme-specific encoding
00 # Node Number
00 # Service Number

```

The six-octet encoding of the 'null' ipn EID using the three-element scheme-specific encoding would be as follows:

```

82          # 2-Element Endpoint Encoding
02          # uri-code: 2 (IPN URI scheme)
83          # 3 Element ipn EID scheme-specific encoding
00 # Default Allocator
00 # Node Number
00 # Service Number

```

## Acknowledgments

The following DTN Working Group participants contributed technical material, use cases, and critical technical reviews for this URI scheme update: Scott Burleigh of the IPNGROUP, Keith Scott, Brian Sipos of the Johns Hopkins University Applied Physics Laboratory, Jorge Amodio of LJC Electronics, and Ran Atkinson.

Additionally, the authors wish to thank members of the CCSDS SIS-DTN Working Group at large who provided useful reviews and commentary on this document and its implications for the future of networked space exploration.

## Authors' Addresses

### Rick Taylor

Aalyria Technologies

Email: [rtaylor@aalyria.com](mailto:rtaylor@aalyria.com)

**Ed Birrane**

JHU/APL

Email: [Edward.Birrane@jhuapl.edu](mailto:Edward.Birrane@jhuapl.edu)