Network Working Group
Internet-Draft
Intended status: Informational
Expires: 3 October 2026

C. Kerrison, Ed.
1 April 2026

# Representation of Intricate Communications
# draft-kerrison-representation-intricate-comms-00

## Abstract

Complex inter-party communication or relationship dynamics can be implied within the use of more structured protocols. This document proposes a compact binary representation for describing these dynamics in a non-protocol-binding manner that can be readily converted back to a readable format and provide additional context for implementers.

## Status of This Memo

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

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

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

This Internet-Draft will expire on 3 October 2026.

Table of Contents



## 1. Introduction

The role of this document is to address complex inter-party communication or relationship dynamics in systems, where there may be compliance with a technical specification or protocol but where there are additional conventions or semantics which fall outside of the scope of application layer protocols.

This document proposes a common lexicon and compact binary representation for describing these dynamics in a non-protocol-binding manner that can be readily converted back to a readable format and provide additional context for implementers.

## Distribution

This memo is submitted for informational purposes within the IETF community. Distribution of this memo is unlimited.

## 2. Relationship Representation and Bitstream Encoding

A series of high-level statements about the relationship maintained between the two parties will first be created using a limited lexicon. While this may reduce the apparent flexibility of the model, an example will be provided later to demonstrate the flexibility of this approach. Through a reversible mapping of this lexicon to known length bit patterns, a continuous high-density bitstream can be used to represent these complex relationships in a manner that reduces overhead. In all following examples, bit patterns should be read left to right.

## 3. Lexicon

These terms allow a flexible representation of the relationship between parties, and were selected to allow encoding of complex relationships. The lexicon is divided into four categories: Separators, Nouns, Verbs, and Adjectives.

### 3.1. Separator

The separator token is encoded using the bit pattern 00. Separators may be used to terminate statements, provide structural boundaries, or introduce intentional pauses. When multiple separators appear consecutively, the first is interpreted as a statement boundary, while subsequent occurrences are treated as placeholders or extended pauses for emphasis.

* 00 — Separator

### 3.2. Nouns

Nouns are encoded using the bit prefix 01 followed by four additional bits that identify the specific semantic token.

* 01 0000 — Transmitting Party
* 01 0001 — Receiving Party
* 01 0010 — Both Parties
* 01 0011 — Internal State
* 01 0100 — Connection/Communication
* 01 0101 — Protocol Specification
* 01 0110 — Malintent
* 01 0111 — Sequence/History
* 01 1000 — Application
* 01 1001 — Operation
* 01 1010 — Heuristic
* 01 1011 — Reserved
* 01 11xx — Illegal or malformed noun

### 3.3. Verbs

Verbs are encoded using the bit prefix 10. These tokens describe actions, intentions, or operational semantics between the parties.

* 10 0000 — Connect/Transmit
* 10 0001 — Disconnect
* 10 0010 — Propose/Consider
* 10 0011 — Know/Comprehend/Agree
* 10 0100 — Option/Choice
* 10 0101 — Must
* 10 0110 — Will Not
* 10 0111 — Drop
* 10 1000 — Iterate
* 10 1001 — Ignore
* 10 1010 — Intend
* 10 1011 — Fabricate
* 10 1100 — Damage/Impact
* 10 1101 — Execute
* 10 1110 — Also (Adverb)
* 10 1111 — Receive

### 3.4. Adjectives

Adjectives are encoded using the bit prefix 11. These tokens modify or qualify nouns or verbs within the relationship description.

* 11 0000 — Established
* 11 0001 — Complete
* 11 0010 — Not Available

* 11 0011 — Other
* 11 0100 — No Time
* 11 0101 — Short Time
* 11 0110 — Moderate Time
* 11 0111 — Long Time
* 11 1000 — Reluctant
* 11 1001 — Unable
* 11 1010 — Obscured
* 11 1011 — Previous Object
* 11 11xx — Illegal or malformed adjective

## 4. Encoding and Decoding

Encoding is performed as an append operation, adding the bit pattern for the new token to the end of the stream.

Decoding is performed by taking the preamble of each new token, a 2-bit pattern, to classify which portion of the lexicon is being used. The token is either handled as a separator or the next 4 bits are read to index into the lexicon subset. Overall only a pointer into the bytestream and a count of recently decoded separators are required state to cleanly decode a well-formed bitstream.

### 4.1. Example Statement

The following shows a statement about the relationship between both parties encoded into a bitstream.

Whitespace in the encoded form is for readability only and does not appear in the actual bitstream.

Example (human-readable form):

[Both Parties] [Must] [Know/Comprehend/Agree] [Protocol Specification] [Separator]

Encoded bitstream:

01 0010 10 0101 10 0011 01 0101 00

## 5. Extended Example

Much longer descriptions of the dynamic between two nodes expected to communicate can be represented in hexadecimal form. The following example shows a complete encoded relationship description, with four additional separators appended to ensure byte alignment.

```
 4a35d5118d5b90318d088466bfbcd010aa04c96d4630d11461344519cd11a2944d1146db0d1182134460a
ec444a3cf7114ec466804d28c4a3612b7b244609104c9918119a3d1181184d118119cd11a2944d1146db0d118
2134460aec44d11461344519cd11a2944d1146db0d1182134460aec4408118408118434460d1180811840d118
3446020461128f3dc453b119a0134a3128d84adec42a81325b518c3445184d1146734468a5134451b6c344608
4d1182bb11
```

## 6. Conclusion

The sample generated above demonstrates a description of communication dynamics first translated from plain English into lexicon tokens then into bitstream representation. The original representation consisted of 2106 bytes of Unicode text. In this compact representation, only 182 bytes were needed to represent the communication. Further, this improved even on the deflated size for the English/Unicode representation of 397 bytes. These results highlight the efficiency of the encoding model, particularly for verbose descriptions of inter-party dynamics.

While the lexicon is intentionally limited, the examples illustrate that it is expressive enough to capture a wide range of relationship semantics. The reversible mapping to fixed-length bit patterns ensures that encoded streams remain compact, unambiguous, and straightforward to decode. This approach may be applicable to systems where communication patterns, expectations, or behavioral conventions must be conveyed alongside or outside of traditional protocol structures.

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

This document should not affect the security of the Internet.

## Indirect Contributors

The author acknowledges the indirect influence of works created by M. Stock, M. Aitken, and P. Waterman, which informed aspects of the extended example. These individuals did not participate in the preparation or submission of this document.

## Author’s Address

Clinton Kerrison (editor)

Email: aussieklutz@gmail.com