RFCScope: Detecting Logical Ambiguities in Internet Protocol Specifications

Mrigank Pawagi[1] Lize Shao[2] Hyeonmin Lee[2] Yixin Sun[2] Wenxi Wang[2]

[1]Indian Institute of Science, Bengaluru, India [2]University of Virginia, Charlottesville, USA

Motivation

- Internet Protocol specifications are prone to ambiguities, undermining their interoperability and implementation correctness.
- LLMs can automatically analyze specifications by reading heterogeneous formats, combining information from different sources, and applying common networking knowledge.

Challenges of Using LLMs

1. Lengthy specification documents
2. Multi-document reasoning
3. Limited domain knowledge
4. Prone to hallucination

Key Contributions

- First systematic study of logical ambiguities in Internet Protocol specifications (RFCs)
- First framework for automatically detecting ambiguities in RFCs

- Natural Language (in RFC 9460)

```
Within a SVCB RRset, all RRs
SHOULD have the same mode. If
an RRset contains a record in
AliasMode, the recipient MUST
ignore any ServiceMode records
in the set.
...
```

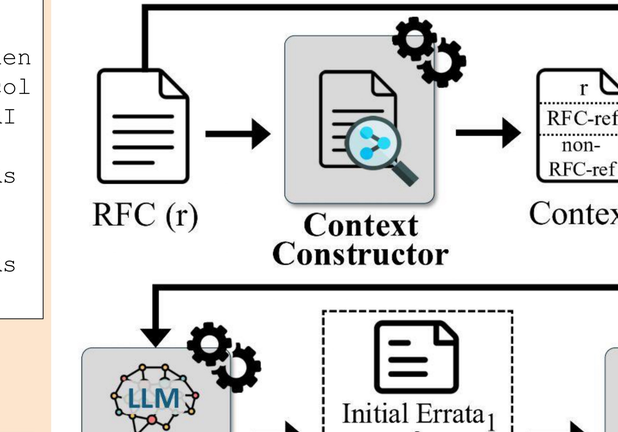
- Pseudocode (in RFC 9000)

```
ReadVarint(data):
 // The length of variable-length
    integers is encoded in the
 // first two bits of the first byte.
  v = data.next_byte()
  prefix = v >> 6
  length = 1 << prefix
...
```

- Formal Notation (in RFC 8461)

```
The formal definition of the "_mta-sts" TXT record, defined using
ABNF[RFC7405], is as follows:
 sts-text-record  =  sts-version 1*(sts-field-delim sts-field)
                     [sts-field-delim]
 sts-field        =  sts-id /           ; Note that sts-id record
                     sts-extension      ; is required.
                                ...
```

- Diagram (in RFC 9085)

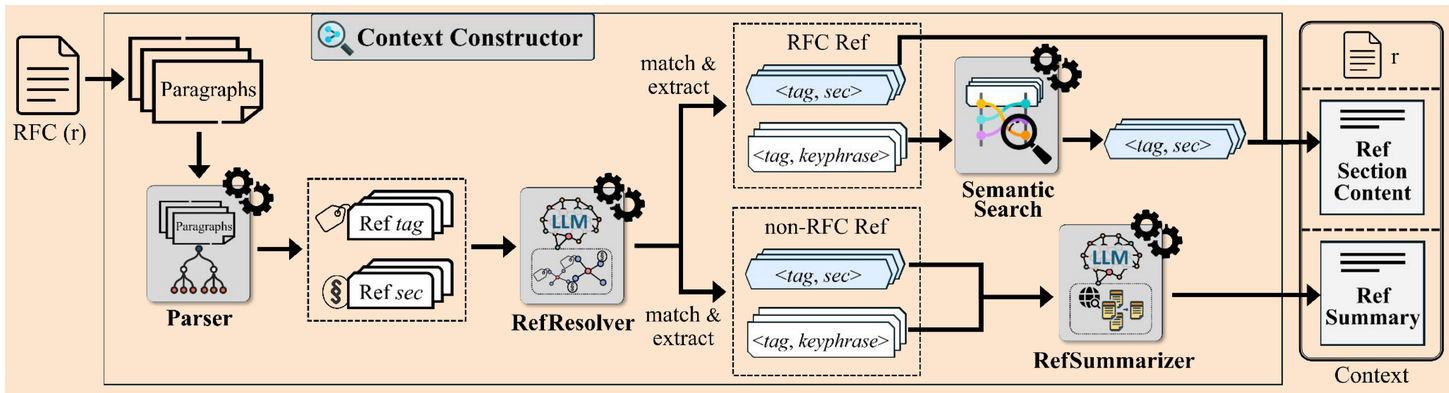


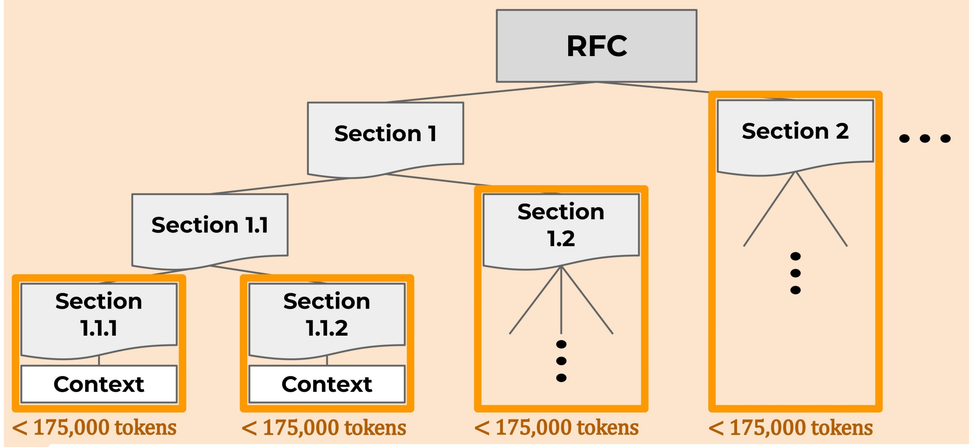
Study

Manual classification of 273 verified technical errata reports from Standards Track RFCs published between January 2014 and January 2025

Main Category (Total Count)	Sub-Category		Count
Inconsistency (202)	I-1	Direct inconsistency within or across specifications	119
	I-2	Indirect inconsistency within or across specifications	70
	I-3	Inconsistency with commonly accepted knowledge	13
Under-specification (37)	U-1	Direct under-specification due to undefined terms	7
	U-2	Direct under-specification due to incomplete constraints	15
	U-3	Indirect under-specification within or across specifications	10
	U-4	Under-specification due to incorrect or missing references	5
Others (34)		Editorial errors	15
		IANA considerations	13
		Suggestions or proposals	6

Approach





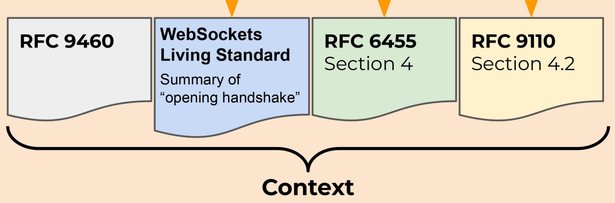
Example: RFC 9460

9.6. Use of HTTPS RRs in Other Protocols

```
All HTTP connections to named origins
are eligible to use HTTPS RRs, even when
HTTP is used as part of another protocol
or without an explicit HTTP-related URI
scheme (Section 4.2 of [HTTP]). For
example, clients that support HTTPS RRs
and implement [WebSocket] using the
altered opening handshake from
[FETCH-WEBSOCKETS] SHOULD use HTTPS RRs
for the request URL.
```

Look up and summarize `websockets.spec.whatwg.org`

Semantic search for "opening handshake"



KEY SOURCES

- Draft versions from **IETF Datatracker**
- Discussions between authors from **IETF Mail Archive**
- Commit history, pull requests, issues from **GitHub**

The LLM is asked to detect ambiguities in the given section with the **context** and **insights from our study**.

1. Ambiguity types to find
2. Strategies to detect different ambiguity types
3. Examples of each ambiguity type

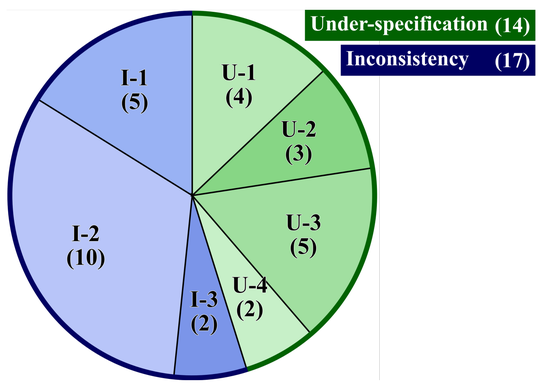
LLM-as-a-Judge

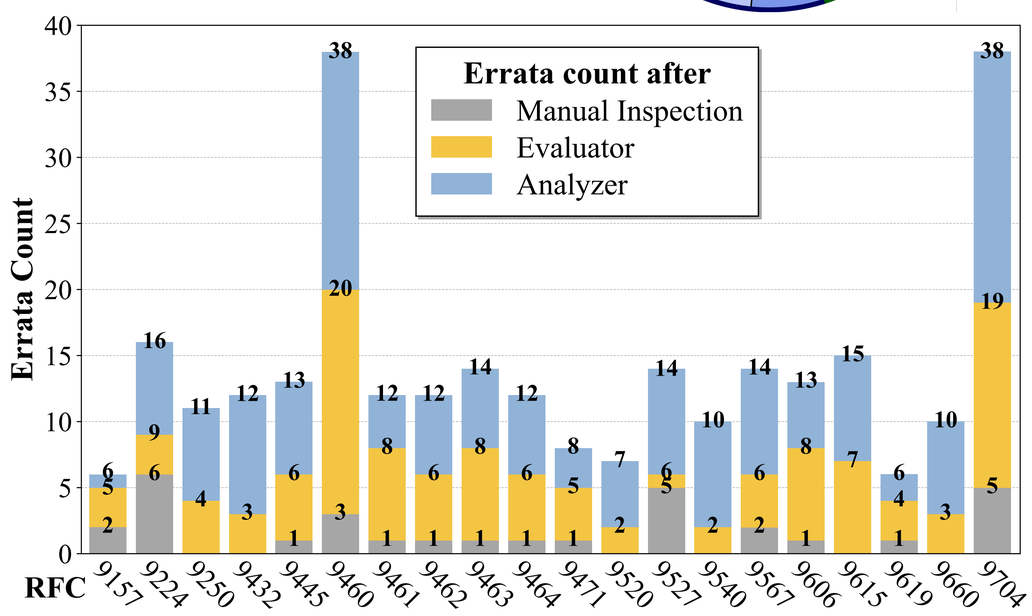
The LLM is prompted to evaluate the results of the Analyzer by

1. validating the reasoning provided by the Analyzer.
2. checking each reported errata against a **checklist-criteria** for plausible errata.
3. identifying which category of ambiguities each reported errata belongs to.

Evaluation

- RFCScope discovered 31 previously unreported ambiguities across 14 of the 20 most recent RFCs related to Domain Name System (DNS).
- 8 of these have been confirmed by RFC authors, of which 3 are officially verified as technical errata.





Verified Technical Errata 8426 (`rfc-editor.org/errata/eid8426`) **I-2**

```
RFC 9619

Section 1. Introduction                                              (Spec)
...update the DNS base specification to clarify the allowable values of the QDCODE parameter in the
specific case of DNS messages with OPCODE = 0.
```

Verified Technical Errata 8431 (`rfc-editor.org/errata/eid8431`) **I-2**

```
RFC 9445

Section 5. An Example: Applicability to Encrypted DNS Provisioning                (Spec 1)
...replies with an Access-Accept message (possibly after having sent a RADIUS Access-Challenge message...

Section 7. Table of Attributes                                                    (Spec 2, table)
+=============+=======+=========+===========+=====+================+
| Access-     |Access-| Access- | Challenge |#    | Attribute      |
| Request     |Accept | Reject  |           |     |                |
+=============+=======+=========+===========+=====+================+
```

Verified Technical Errata 8590 (`rfc-editor.org/errata/eid8590`) **U-4**

```
RFC 9704

Section 6.2. Using DNSSEC                                                         (Spec 1)
The client ... performs full DNSSEC validation locally [RFC6698].

RFC 6698                                                                          (Spec 2, ref)

Section 1.2. Securing the Association of a Domain Name with a Server's Certificate
This document only relates to securely associating certificates for TLS and DTLS with host names;
retrieving certificates from DNS for other protocols is handled in other documents.

Section 1.3. Method for Securing Certificate Associations
This document does not specify how DNSSEC validation occurs...
```