

IP Performance Measurement
Internet-Draft
Intended status: Standards Track
Expires: 1 November 2026

T. J. Pinkert
Siemens Mobility GmbH
30 April 2026

Inter-layer Protocol
draft-pinkert-ippm-inter-layer-protocol-01

Abstract

This document describes an inter-layer protocol, that can be used to insert an arbitrary number of headers between the internet protocol (IPv4, IPv6) header and a layer four header like the UDP or TCP header. By doing so, it consumes part of the space available for IP payload data. It is, in particular, useful to extend the space reserved for IP options as defined in the IPv4 protocol [RFC791].

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://DutchyatWork.github.io/rfc-draft-inter-layer-protocol/draft-pinkert-ippm-inter-layer-protocol.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-pinkert-ippm-inter-layer-protocol/>.

Discussion of this document takes place on the IP Performance Measurement Working Group mailing list (<mailto:ippm@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/ippm/>. Subscribe at <https://www.ietf.org/mailman/listinfo/ippm/>.

Source for this draft and an issue tracker can be found at <https://github.com/DutchyatWork/rfc-draft-inter-layer-protocol>.

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 1 November 2026.

Copyright Notice

Copyright (c) 2026 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	2
2. Conventions and Definitions	3
3. Generic properties of an inter-layer protocol in the stack	3
4. Inter-layer protocol for IPv4	4
5. IPv4 inter-layer protocol usage.	6
6. Encoding existing IP options in the inter-layer block.	8
7. Security Considerations	9
8. IANA Considerations	9
9. Normative References	9
Acknowledgments	10
Author's Address	10

1. Introduction

The inter-layer protocol is a somewhat dirty protocol. A handler for it is written as part of the lower protocol layer. The inter-layer protocol is identified by a protocol identifier at that layer.

In the original scope the protocol was designed to be placed in between the IP layer and a transport layer protocol like UDP or TCP. In that scope an IP protocol number is to be assigned to this protocol by IANA.

The protocol handler takes over from the IP protocol handler when the IP Protocol indicates the inter-layer protocol. Since inter-layer protocol blocks, may be concatenated, the protocol handler keeps parsing inter-layer protocol blocks, until the next protocol field (IP Protocol) identifies a different next layer protocol (typically UDP or TCP). It hands back control to the IP handler, that invokes the next protocol handler.

Since the inter-layer protocol is highly related to the lower layer handler, another implementation option, is an extension of the lower layer handler, in the scope of this document, this would be the IPv4 protocol handler.

One of the goals of the inter-layer protocol, is to open up freely useable space / header extension, of the lower layer protocol. This can be flexibly used if the lower layer protocol has no, or not enough, flexibility in the protocol header. In case of the IPv4 header, the use of IPv4 options is possible, but the space in the IPv4 header for IP options is limited to 10 words, or 40 octets, of data. For some applications this may simply be too limited.

E.g., routing options requesting routers to place a tag in an IPv4 option, may allow for only 9 router IP addresses. In an inter-layer implementation, this space can be much larger, e.g., allowing 100 routers to place their addresses in the table. Another use case, can be the use of IP options that have a native length that extends beyond 10 words. Securing IP option contents by use of encryption, or digital signatures, may be such a case. A ciphertext of 40 octets may already be limited for use of state-of-the-art encryption methods.

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 [BCP14] (RFC2119) (RFC8174) when, and only when, they appear in all capitals, as shown here.

3. Generic properties of an inter-layer protocol in the stack

The inter-layer protocol is designated by use of the next protocol number to implement a recursive / repetitive block structure between the lower layer and upper layer protocol. The entire data structure can, however, not grow beyond the boundary set by the upper data unit size of the lower layer protocol. In case of the IPv4 protocol that upper limit is 64 kilobytes.

An inter-layer protocol header MUST at least contain the following to function:

1. A field repeating the lower protocol number,
 - * In case of IPv4, the Protocol field
2. A field indicating the length of the inter-layer protocol block
 - * The length includes the header of this block.
 - * The length can be indicated in bits, octets, or words as is seen fit by the protocol designer.
3. A field indicating the type of contents of the inter-layer data,
 - * In some cases, the type is the data.
 - * In other cases, the type specifies which parser is to be used for the inter-layer data.

The following fields COULD be contained:

1. An (optional) inter-layer user data block,
2. An (optional) Checksum block,

In many cases the checksum block can be skipped, since the lower layer protocol already includes a checksum over the protocol data unit transported. This checksum then includes the data of the inter-layer blocks, which are not seen separately from the upper layer protocol data unit.

The length of the lower layer protocol data unit is the length of the upper layer protocol data unit plus the length of all inter-layer protocol blocks.

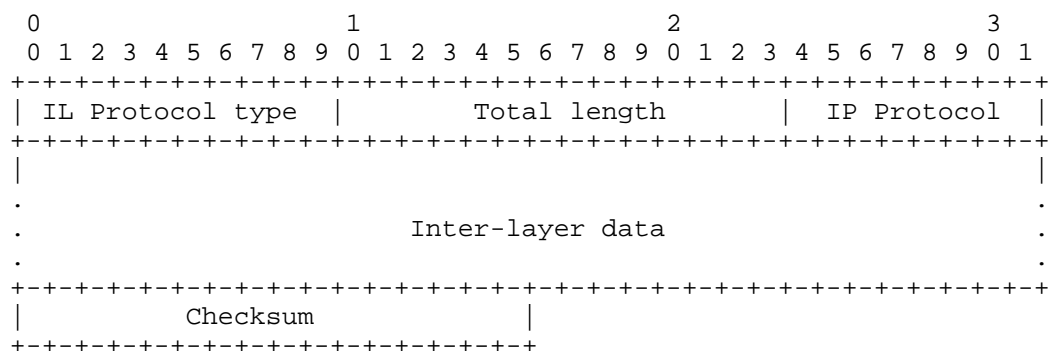
4. Inter-layer protocol for IPv4

It is proposed to shape the inter-layer header for the IPv4 protocol as follows. A 10-bit inter-layer protocol type allows for 1024 inter-layer protocol types. Since it is proposed to reuse the IP option types, 256 of these protocol numbers are already taken, still leaving sufficient space for new protocol definitions.

The length is defined in octets, a 16-bit length identifier would allow to use the entire IP user data space as inter-layer space. This is not deemed useful. A length of 16 kbyte, is deemed

sufficient. Then 8 bits are left for the IP Protocol number, needed to jump back from parsing inter-layer protocol blocks to the normal protocol stack parsing. A 16-bit checksum, such as used in the UDP protocol is the last part of the inter-layer protocol data unit, and is included in the length of the protocol data unit. The inter-layer data is not necessarily word aligned. The total length indication is typically used by the protocol handlers to indicate the start position of the next protocol data unit.

The inter protocol data unit now looks as follows. The minimum length is 4, when only the header is present.



IL Protocol type (10 bits):

inter-layer protocol type, designates how to interpret the inter-layer data defined by a separate inter-layer protocol specification.

Total length (14 bits):

Length of the inter-layer protocol unit in words.

IP Protocol (8 bits):

IP protocol type of the next IP header part.

Inter-layer data:

inter-layer protocol specific data part defined by a separate inter-layer protocol specification.

Checksum (16 bits):

16-bit checksum over the inter-layer protocol data unit.

5. IPv4 inter-layer protocol usage.

In the figure below an IP header with a UDP packet is depicted.

0																1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																																
Version				IHL				Type of Service				Total Length																IPv4																																			
Identification																Flags				Fragment Offset																IPv4																											
Time to Live								Protocol = 0x11								Header Checksum																				IPv4																											
Source Address																																				IPv4																											
Destination Address																																				IPv4																											
Options																				Padding																IPv4																											
Source Port																Destination Port																				UDP																											
Length																Checksum																				UDP																											
UDP data octets																																				UDP																											
																																				UDP																											

We see that the Protocol number in the IP protocol header indicates the use of the UDP protocol as next protocol in the stack. The IP protocol handler will hand the IP user data to the UDP handler based on that number.

For the inter-layer protocol, it would look as follows. We assume the IL Protocol number for the IPv4 Protocol field is 0xFF. The IL protocol types are taken as 0xFFE and 0xFFFF. Each of these protocols has an own handler. Protocol 0x11 is the UDP protocol again.

0										1										2										3										
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1									
+-----+																																								
Version					IHL					Type of Service					Total Length																				IPv4					
+-----+																																								IPv4
										Identification					Flags					Fragment Offset															IPv4					
+-----+																																								IPv4
										Time to Live					Protocol = 0xFF					Header Checksum															IPv4					
+-----+																																								IPv4
										Source Address																				IPv4										
+-----+																																								IPv4
										Destination Address																				IPv4										
+-----+																																								IPv4
										Options																				Padding										IPv4
+-----+																																								IPv4
IL Protocol = 0xFFFF										Total length										Protocol = 0xFF															ILP					
+-----+																																								ILP
																																			ILP					
.										Inter-layer data for protocol number 0xFFFF																									ILP					
																																			ILP					
																				+-----+															ILP					
																														Checksum										ILP
+-----+																																								ILP
IL Protocol = 0xFFE										Total length										Protocol = 0x11															ILP					
+-----+																																								ILP
																																			ILP					
.										Inter-layer data for protocol number 0xFFE																									ILP					
																																			ILP					
																				+-----+															ILP					
																														Checksum										ILP
+-----+																																								UDP
										Source Port										Destination Port															UDP					
+-----+																																								UDP
										Length										Checksum															UDP					
+-----+																																								UDP
																																			UDP					
.										UDP data octets																									UDP					
																																			UDP					
+-----+																																								UDP

In the IP header, the protocol type 0xFF hands control for the next layer protocol to the inter-layer protocol handler. This handler interprets the first block of inter-layer protocol data. It then reads the IP Protocol number from the inter-layer protocol header, the next protocol being 0xFF. Thus the inter-layer protocol handler also interprets the next block of inter-layer protocol data. After that it reads the IP Protocol number from the header of the second block of inter-layer protocol data, it being 0x11. The inter-layer

protocol handler knows that no more inter-layer blocks of data are to be expected, and either hands control back to the IP layer handler, or to the next layer handler. These make the decision based on the last Protocol type handed to them by the inter-layer protocol layer handler, that the UDP protocol handler must be invoked.

Note that for convenience the inter-layer header blocks are word aligned in the drawing. This must not be the case in reality.

6. Encoding existing IP options in the inter-layer block.

One of the applications of the inter-layer protocol is a redefinition of IP options to reach more flexibility. Since the IP option type is encoded in one byte, it is proposed to use the 0x0xx block of codes for the already specified IP options. For the one octet IP options (option type only) the inter-layer length becomes 6 octets. E.g., the No Operation IP option [RFC791] would be encoded as follows in the inter-layer protocol.

0	1	2	3																								
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																											
+-----+																											
0 0 0 0 0 0 0 0 0 0 1										0 0 0 0 0 0 0 0 0 0 0 1 0 1										IP Protocol							
+-----+																											
										Checksum																	
+-----+																											

The next layer protocol being left open, and the checksum correctly calculated over the first 4 octets.

The record route option with a storage space of 64 octets of IP addresses would look as follows. We encode the length octet in the Total length field of the inter-layer protocol data unit.


```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|0 0 0 0 0 0 0 1 1 1|   Total length = 0x47   |   IP Protocol   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   pointer   |   IP address 1   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               |   IP address 2   |
+-----+-----+-----+-----+-----+-----+-----+-----+
.               .               .
.               .               .
.               .               .
+-----+-----+-----+-----+-----+-----+-----+-----+
|               |               Checksum               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Also, here it must be noted that the option data are not aligned. However, this should not be a problem in modern computer systems that have sufficient processing power.

7. Security Considerations

No special considerations are needed regarding this protocol. It has no fundamental differences to known protocols without security such as IPv4, IPv6, UDP, or TCP.

8. IANA Considerations

This document requests an IP Protocol number to be assigned by IANA with the following properties:

- * keyword: I-L-Proto
- * Protocol: Inter-Layer Protocol

It also requests the creation of a registry for Inter-Layer Protocol numbers. Table 00 of this registry is to be reserved for use with existing IP option numbers as listed in the IP Option Numbers registry.

9. Normative References

- [BCP14] Best Current Practice 14,
<https://www.rfc-editor.org/info/bcp14>.
 At the time of writing, this BCP comprises the following:

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>>.

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>>.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/rfc/rfc791>>.

Acknowledgments

Tjeerd Pinkert wants to thank Gert Bolz, and Benjamin Schilling for their support of the hybrid network measurements innovation project, and Sascha Liebscher, Achim Willers, Tobias Grosch, and Jaime Lazaro Calderon for their support to make this work possible within Siemens Mobility.

Author's Address

Tjeerd J. Pinkert
Siemens Mobility GmbH
Ackerstrasse 22
38126 Braunschweig
Germany
Email: tjeerd.pinkert@siemens.com
URI: <https://www.mobility.siemens.com>