Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 12, 2011

K. Uehara
Keio Univ.
M. Imaike
FEAC
K. Shima, Ed.
IIJ II

November 8, 2010

The Transport Protocol for Decentralized Probe Applications for Vehicles

draft-uehara-dtnrg-decentralized-probe-transport-00

Abstract

(This document is a description of the decentralized viechle to viechle probe mechanism currently being prototyped. The main purpose of disclosing this -00 document is to introduce the application idea and start discussion within the DTNRG. Because of this, the current mechanism described in this docment does not conform to the protocols defined in the DTNRG at this moment. The mechanism should be updated based on the discussion in this group.)

This document describes the transport protocol specification used for the decentralized probe system for vehicles. The probe system exchanges probe messages between vehicles using a wireless communication methods with low bandwidth and lossy properties. The communication environment dynamically changes as vehicle moves. The protocol try to utilize the limited bandwidth as much as possible and increase reachability rate by using sender-side error correction mechanism.

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 http://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 May 12, 2011.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Protocol Overview 3
4. Message Multiplex/De-multiplex 4
5. Message Fragment/Reassemble 4
5.1. Reed-Solomon Processing 5
6. Packet Sending Rate Control 6
7. ICM Packet Format 7
7.1. ICM Common Header 7
7.2. ICM Multiplex Header 8
7.3. ICM Fragment Header 9
8. Protocol Constants 10
9. IANA Considerations 10
10. Security Considerations 10
11. Normative References 10
Authors' Addresses 10

1. Introduction

Thanks to the the advance of the Internet and the wireless communication technology, it becomes possible to equip communication mechanisms for vehicles for various purposes, such as for safety applications, entertainment applications, and so on. Recently, another approach utilizing the DTN (Delay Tolerant Network) technology for inter-vehicle communication is attracting people from both academy and industry. One of the application of the approaches is decentralized probe system for traffic information. It is considered that by combining the existing centralized approach for traffic monitoring and the proposed decentralized approach, more detailed information exchange is possible and wider area can be covered. This document proposes a standard way to exchange packets between vehicles.

2. Terminology

ICM: Inter-vehicle communication module.

3. Protocol Overview

Each vehicle is equipped with at least one ICM (Inter-vehicle communication module) device. ICM interacts with other ICM devices equipped in other vehicles using the UDP mechanism. ICM receives application messages which are generated by the application software (e.g. traffic information generator) in the same vehicle and broadcasts to other ICM devices nearby as ICM packets. It also receives broadcasted packets from other ICM devices, extract application messages and pass them to the application software.

To utilize the lower layer link bandwidth as much as possible, the ICM device uses the following two techniques.

1. Message multiplexing/de-multiplexing

2. Message fragmentation/reassemble with FEC

Since the lower layer media may only have limited bandwidth, the protocol tries to multiplex application messages into one UDP packet whenever possible. The number of multiplexed messages depends on the MTU size of the lower layer media and the size of each application messages. When the ICM device receives a multiplexed packet, it de-multiplexes the received packet and passes each message to upper layer applications.

On the other hand, some application messages may be bigger than the maximum allowed transmission size of the link layer. In this case, the ICM device fragments the message into several blocks to fit to the MTU size, and add additional error recovery information using the Reed-Solomon Coding. Since the packets are broadcasted and no acknowledgement messages is defined, the FEC mechanism is adopted to reduce the loss of the fragmented packets which causes the entire message loss.

The underlying layer 2 mechanism is out of the scope of this document. The requirements to the lower layer media is as follows.

o UDP and IP (either IPv4 or IPv6) capability

4. Message Multiplex/De-multiplex

If the ICM packet that contains an application message is smaller than the MTU of the outgoing interface, the ICM device may multiplex following ICM packets up to the MTU size. Every ICM packets is concatenated and packed into a UDP payload as long as the total size of the UDP packet does not exceeds the MTU size.

When receiving an ICM packet, the node first checks the ICM payload length of the first ICM packet in the received UDP payload. If the size of the UDP payload is bigger than the size of the first ICM packet, then extract following ICM packets until no extra bytes are found in the UDP payload. Each extracted ICM packets is passed to the upper layer application after removing the ICM header described in Section 7.

5. Message Fragment/Reassemble

If the application message passed from the upper layer to the ICM layer is bigger than the link MTU size, the message will be fragmented into several ICM packets. Since this Inter-vehicle communication protocol does not provide message reachability confirmation, the Reed-Solomon error correction mechanism is used when making each fragment. Each ICM packet header contains the partial length of the fragmented data and offset from the original message.

Since the Reed-Solomon mechanism is used, even if some of the fragmented packets are lost, the original message can be reassembled. Once the application message is reassembled, the message is passed to the upper layer application.

5.1. Reed-Solomon Processing

A large message will be fragmented to fit to the link MTU. Since the link is typically provided as a wireless media, the link quality is not always stable. To avoid losing application message as much as possible, the large message is encoded using the Reed-Solomon procedure.

The parameter of coding is described in the ICM Fragment Header as described later in Section 7.3. In this section, we use 4 as the bit length of symbols, and 11 as the number of data symbols as example parameters. With these parameters, we can encode 5 bytes data into 7 bytes data including 2 bytes error correction code, and we can recover 1 byte error at maximum.

When the message is fragmented and encoded, the original message is first divided into several messages using the unit of the fragmented message size. The Reed-Solomon coding is done using the n-th bytes of each fragmented message. With the parameters assumed in this section, 5 packets are sent first and 2 packets which include error correction data follow the data packets. If the last part of the encoded block is shorter than the 5 packets, then it is assumed that all-0 data are virtually appended to the end of the message.

Figure 1 shows an example of a large message fragmented into 8 parts. Since we can only encode 5 bytes at one time with the currently assumed parameters (4 bits symbol data, and 11 data symbols), the first 5 fragment packets are sent first. The following 2 packets include the error correction data calculated based on the Reed-Solomon coding. As described in the previous paragraph, the calculation is done based on the position of each data in the each fragmented packets. The first bytes of the two error correction packets is calculated with the first bytes of the previously sent 5 data packets. The following error correction data are calculated in the same way.

In this case, the second data block does not have full data, because of the original message is too short to fill the last block. In this case, the Reed-Solomon coding is performed as if there were data in the missing positions. The value of the virtual data is 0.

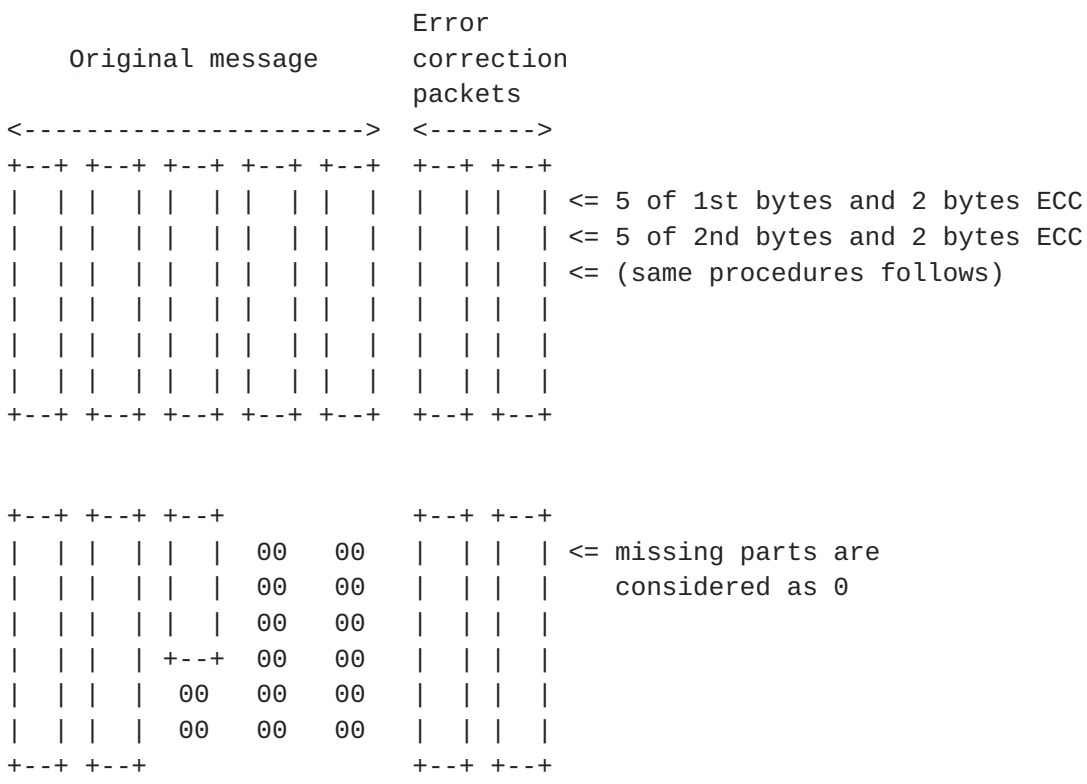


Figure 1: Reed-Solomon Processing Example

6. Packet Sending Rate Control

Since the communication model defined in this specification is not centralized, and not based on the one to one communication model, it is impossible to achieve precise information of the packet sending rate. To avoid link congestion, all the sender node must control its packet sending rate autonomously.

One assumption in this specification is that the available bandwidth of the link is notified by the lower layer protocol stack. The packet sending/receiving layer should behave not to overuse the available bandwidth, and not to overload the link so that other nodes nearby can send their packets.

A reference congestion control scheme defined in this specification is as follows.

The node periodically counts the number of nodes nearby by monitoring the broadcast (in IPv4) or multicast (in IPv6) packets on the link. The available bandwidth for this node is calculated as (Available link bandwidth notified by the lower layer) / (number of neighboring

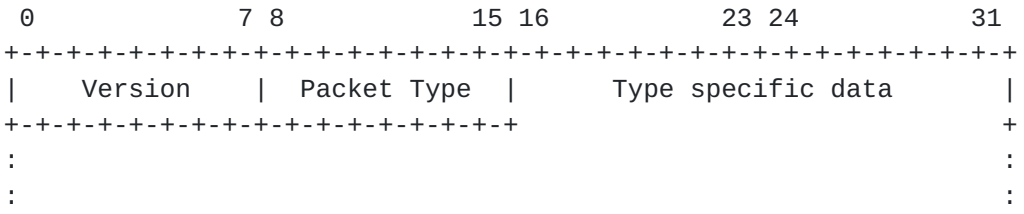
nodes). The packet sending function must respect the value and try to send packets within the available bandwidth to the node.

7. ICM Packet Format

An application message, or one of the fragmented application messages are encapsulated by the ICM packet header described below. All the ICM packets are sent as UDP packets.

7.1. ICM Common Header

The following header is the ICM common header. Based on the value of the Type field, every ICM header has its unique header format.



IP source: The link-local address of the interface of the source node of the packet

The IPv6 link-local address [RFC4291] assigned to the outgoing interface is used if the communication is done in IPv6. The dynamically configured IPv4 link-local address [RFC3927] assigned to the outgoing interface is used if the communication is done in IPv4.

IP destination: The link-local multicast or broadcast address of the outgoing medium.

The IPv6 all-node multicast address [RFC4291] of the outgoing medium is used if the packet is sent as an IPv6 packet, otherwise the IPv4 link-local broadcast address [RFC0919] is used if the packet is sent as an IPv4 packet.

Next header: 17 (UDP)

Source port: (TO BE ASSIGNED BY IANA)

Destination port: Same as the value of the source port

Version: The ICM header format version

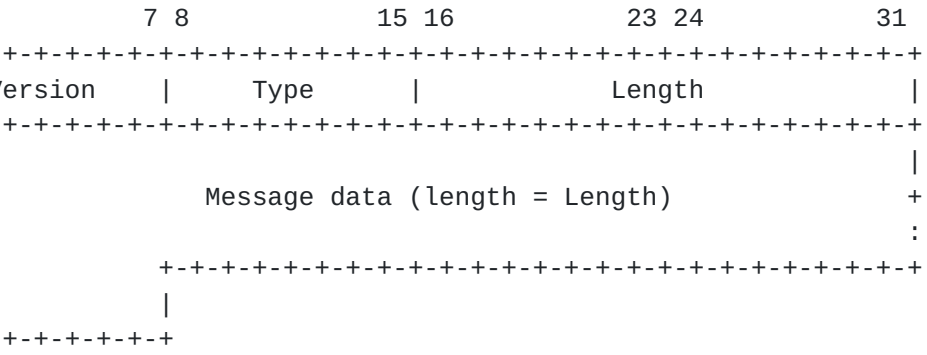
The version number 1 is the only version currently defined.

Packet Type: The packet type of the following ICM packet.

Type specific data: The type specific data

The ICM packet is broadcasted (in IPv4) or multicasted (in IPv6) to all the neighboring nodes. The version number of the ICM packets is 1 at this moment. Based on the processing type, different ICM packet headers are defined.

7.2. ICM Multiplex Header



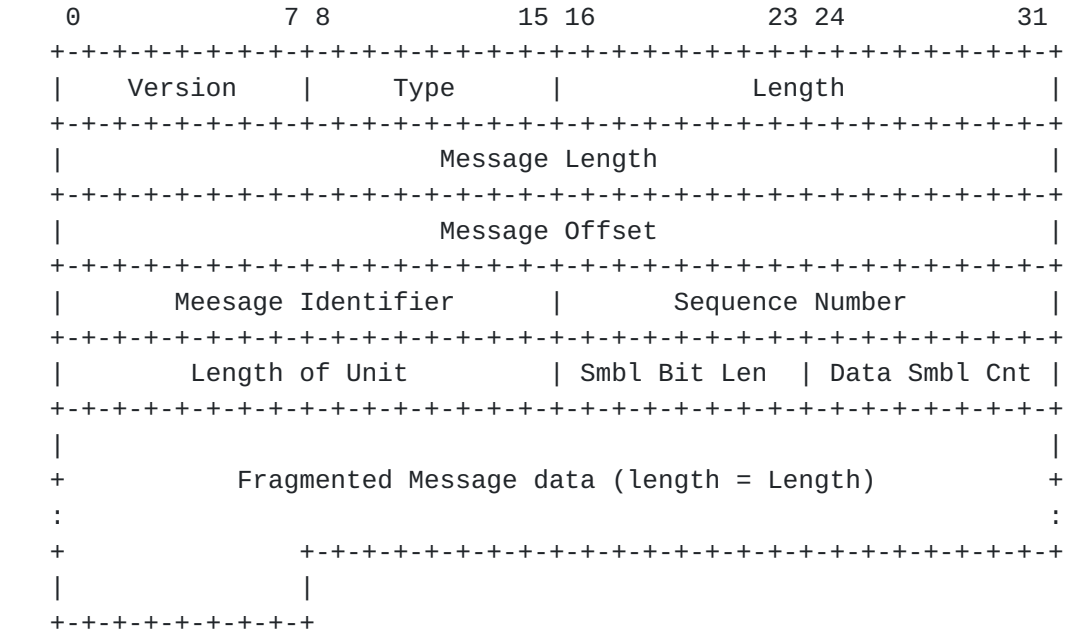
Version: 1

Packet Type: 0

Length: The length of the message data

When multiple messages are multiplexed in one UDP packet, the ICM multiplex messages are concatenated. The total length of the multiplexed UDP packet must not exceed the link MTU.

7.3. ICM Fragment Header



Version: 1

Type: 1 or 2

The packet type 1 is used for the data packets and the packet type 2 is used for the error correction packets.

Length: The length of the fragmented message data

Message Length: The length of the original message

Message Offset: The offset of this fragmented message data from the beginning of the original message

This field includes the offset value of the fragmented part of the message, if the Packet Type value is 1. If the Packet Type value is 2, the field is set to 0.

Message Identifier: A unique identifier assigned to the series of fragmented packets

Sequence Number: The sequence number of fragmented packets and error correction packets

Length of Unit: The unit size of the fragmented message data

Smbl Bit Len: The bit length of the Reed-Solomon coding

Data Smbl Cnt: The number of symbols used as data symbols in the Rees-Solomon processing

8. Protocol Constants

ICM_VERSION: 1

ICM_TYPE_MULTIPLEX: 0

ICM_TYPE_FRAGMENT: 1

ICM_TYPE_ECC: 2

9. IANA Considerations

The UDP port number should be assigned by IANA.

10. Security Considerations

In this layer, message integrity is not assured. The validation of the message contents is performed in the application layer.

11. Normative References

[RFC0919] Mogul, J., "Broadcasting Internet Datagrams", STD 5, RFC 919, October 1984.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

Authors' Addresses

Keisuke Uehara
Keio University
5233 Endo
Fujisawa-shi, Kanagawa 252-8520
Japan

Email: kei@wide.ad.jp

Masayoshi Imaike
FEAC International
3-4-11 Mita
Minato-ku, Tokyo 108-0073
Japan

Email: isle@feac.jp

Keiichi Shima (editor)
IIJ Innovation Institute Inc.
1-105 Kanda-Jinbocho
Chiyoda-ku, Tokyo 101-0051
Japan

Email: keiichi@iijlab.net