

团体标准

T/ITS 00XX—20XX

智能驾驶电子地图路侧分发机制

Distribution mechanism of smart driving electronic map from RSU to OBU

(报批稿)

(本草案完成时间：2022年5月)

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

XXXX - XX - XX 发布

XXXX - XX - XX 实施

中国智能交通产业联盟

目 次

前言.....	11
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义、缩略语.....	1
3.1 术语和定义.....	1
3.2 缩略语.....	2
4 应用场景.....	2
4.1 云端——>边端.....	2
4.2 边端——>车端.....	3
5 系统架构.....	3
5.1 总体架构.....	3
5.2 功能要求.....	4
5.3 智能驾驶电子地图数据格式.....	4
5.4 数据模型.....	5
6 地图分发策略.....	6
6.1 分发策略.....	6
6.2 整合策略.....	7
6.3 分发路径.....	9
6.4 分发数据.....	9
6.5 分发能力.....	9
6.6 OBU 测试.....	12
7 地图解析验证.....	13
7.1 地图启动.....	13
7.2 配置及读取设置.....	14
7.3 解析测试.....	14
7.4 解析结果.....	15
7.5 结论.....	16
附录 A （规范性） 压缩前后数据量要求的详细说明.....	17
A.1 RSU 电子围栏.....	17
A.2 地图分发时间.....	17
A.3 数据更新频率.....	17
A.4 压缩前数据量阈值.....	17
A.5 压缩后数据量阈值.....	17
附录 B （规范性） 智能驾驶电子地图要素的数据模型.....	19
附录 C （规范性） 电子地图数据编译信息标识.....	20
参考文献.....	21

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国智能交通产业联盟（C-ITS）提出并归口。

本文件起草单位：亿咖通（上海）科技有限公司，同济大学，上汽友道智途，清华大学，百度，易图通，武汉理工大学，上海机动车检测认证技术研究中心有限公司

本文件主要起草人：姚灿，胡笳，王平，李俊琦，张杰，郭熹，张显宏，袁泉，肖骁，胡星，汤咏林，陈志军，王观，栗琦，徐佳磊

智能驾驶电子地图路侧分发机制

1 范围

本文件规定了整体智能驾驶电子地图路侧分发的工作内容及流程,本文件的目的是针对不同智能驾驶场景,提供地图分发策略,明确路车交互方式,向车端提供稳定、可靠、灵活的智能驾驶电子地图数据。

本文件适用于基于车路协同的路侧智能驾驶电子地图分发系统。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

- GB/T 31024.1—2014 合作式智能运输系统 专用短程通信 第1部分:总体技术要求
 GB/T 31024.2—2014 合作式智能运输系统 专用短程通信 第2部分:媒体访问控制层和物理层规范
 GB/T 31024.4—2014 合作式智能运输系统 专用短程通信 第4部分:设备应用规范
 T/ITS 0058-2017 合作式智能运输系统 车用通信系统 应用层及应用数据交互标准

3 术语和定义、缩略语

GB/T 31024.1—2014、GB/T 31024.2—2014界定的以及下列术语和定义适用于本文件。

3.1 术语和定义

3.1.1

车用无线通信技术 V2X

(3.1.2) 与其他设备相连接的新一代信息通讯技术,包括但不限于(3.1.2)之间通讯(V2V), (3.1.2)与(3.1.3)通讯(V2I), (3.1.2)与行人设备通讯(V2P), (3.1.2)与网络之间通讯(V2N)。

3.1.2

车载单元 on-board unit

安装在车辆上的具备信息采集、处理、存储、输入和输出接口,具有专用短程无线通信模块的硬件单元。

[来源:GB/T 31024.2—2014, 2.4, 有修改]

3.1.3

路侧单元 road side unit

安装在道路两侧或门架上,通过专用短程无线通信接收来自(3.1.2)的信息和向(3.1.2)发送信息的硬件单元。

[来源:GB/T 31024.2—2014, 2.5, 有修改]

3.1.4

专用短程通信 dedicated short range communication

专门用于道路环境的车辆与车辆、车辆与基础设施之间通信距离受限的无线通信方式,是合作式智能运输系统应用领域的基础通信方式之一。

[来源:GB/T 31024.1—2014, 2.2]

3.1.5

车路协同 vehicle-infrastructure cooperative system

基于无线通信、传感探测等技术进行车路信息获取，通过车车、车路信息交互，实现车辆与基础设施之间协同与配合，以及车辆与其它交通要素之间的通信，从而形成的安全、高效的道路交通系统。

3.1.6

直通接口 PC5

基于车用无线通信技术的一种实现车、人、路之间的短距离直接通信接口。

3.2 缩略语

以下缩略语适用于本文件：

ACK：确认（Acknowledgement）

AD：自动驾驶（Automated Driving）

AID：应用标识（Application ID）

CRC：循环冗余校验（Cyclic Redundancy Check）

MSG：消息（Message）

OBU：车载单元（On-Board Unit）

OD：起讫点（Origin-Destination）

ODD：设计运行区域（Operational Design Domain）

REQ：请求（Request）

RSU：路侧单元（Road-Side Unit）

V2X：车用无线通信技术（Vehicle-To-Everything）

4 应用场景

图1为本标准所规定的智能驾驶电子地图分发的应用场景，包括云端——>边端和边端——>车端两个场景。根据不同的应用场景，采用相应的地图分发策略，从而向车端提供稳定、可靠的智能驾驶电子地图。

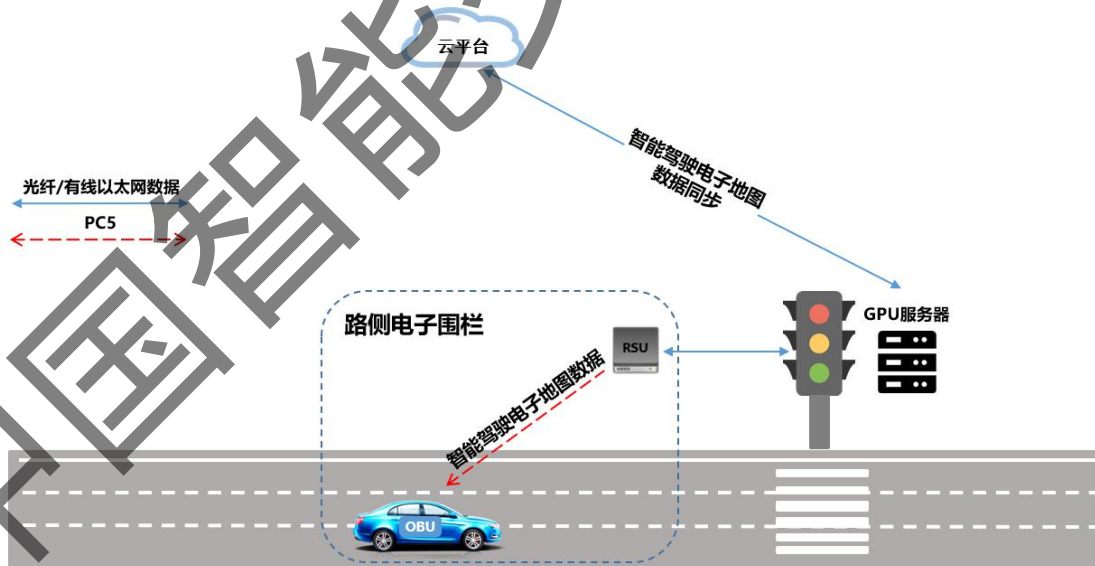


图1 智能驾驶电子地图应用场景示意图

4.1 云端分发至边端

云平台通过光纤或有线以太网将对应区域内智能驾驶电子地图分发到边缘GPU服务器，从而实现云端——>边端的地图分发。

4.2 边端分发至车端

下发到GPU服务器中的地图以以太网的形式传送到RSU。当装载OBU的自动驾驶车辆进入路侧电子围栏并发出下载地图请求后，RSU通过PC5方式将智能驾驶电子地图下发至车辆，实现边端——>车端的地图分发。

5 系统架构

5.1 总体架构

本节定义构成地图分发系统的云平台、边缘云、车路终端，并说明它们之间的数据传输关系。

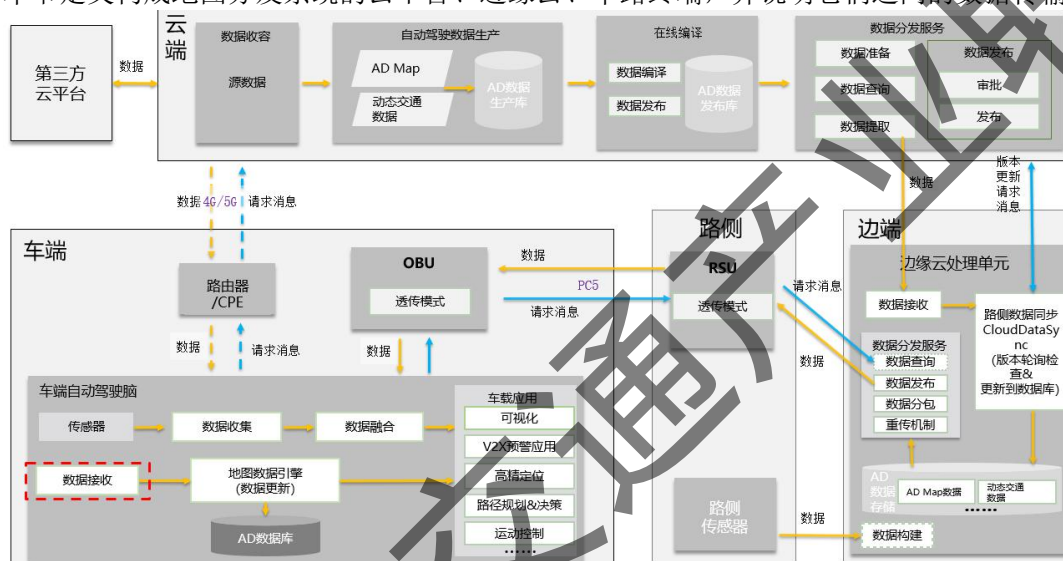


图2 地图分发系统总体架构

如图2所示，智能驾驶电子地图通过云平台——>云端——>边端——>路侧——>车端的路径实时分发到车端。

5.1.1 云端

当车辆发送地图请求消息后，第三方云平台首先传递数据到云端。云端依次进行数据收容、自动驾驶数据生产、在线编译、数据分发等处理步骤，再将数据下发至边端。

- 数据收容：云端对接收到的源数据进行收容处理，得到智能驾驶电子地图数据；
- 自动驾驶数据生产：将智能驾驶电子地图数据导入数据生产库；
- 在线编译：对数据生产库中的自动驾驶数据进行数据编译和数据分布后导入自动驾驶发布库；
- 数据分发服务：包括数据准备、数据查询、数据提取等过程。

云端的数据经过以上几个处理过程后，将智能驾驶电子地图数据由云端下发到边端。

5.1.2 边端

边端对从云端传递的数据在其边缘云处理单元中进行再处理，主要包括以下几个过程：

- 数据接收；
- 路侧数据同步：包括版本检查以及数据库的更新；
- 数据存储：完成数据同步后的智能驾驶电子地图存储到自动驾驶数据库；
- 数据分发服务：包括数据查询、数据发布、数据分包和重传。

边端完成上述过程后，将数据下发到路侧RSU，再由RSU将数据分发到车端OBU。

5.1.3 车端

自动驾驶车辆装载的OBU将接收到的RSU数据传递给车端自动驾驶脑，对地图数据进行更新，再结合车载传感器对周围环境进行感知和定位的数据，车辆开始自动驾驶模式，实现V2X预警、路径规划与决策、运动控制等车载应用。

智能驾驶电子地图利用边缘GPU设备，使得云端的计算负荷整合到边缘层，在边缘计算节点完成绝大多数的计算，并通过路侧单元实时将结果发送给装置车载单元（OBU）的车辆，满足车路协同的需要。通过云—边—车端的路径分发到车辆，整个地图分发过程利用了边缘计算高实时性的优势，有利于实现云车之间高效的互联互通。

5.2 功能要求

表1列出了云端-边端-车端场景下，自动驾驶车辆的行驶范围和行车路径、数据更新类型、时间、频率、策略、数据量等要求。

数据量要求的具体计算可参考附录A。

表 1 自动驾驶车辆的分发需求

字段内容	说明
运营场景	途经车路协同路线的自动驾驶车辆
行驶范围	固定行驶区域(ODD)，部分线路路侧存在RSU单元
行车路径	ODD区域内固定线路/非固定线路
数据更新类型	智能驾驶电子地图数据
数据更新频率	1) 途径RSU动态完成 2) 非运营时段，出车前/充电时完成 3) 行驶时完成
数据更新策略	云端—>边端：地图变化即更新 边端—>车端：10HZ—80HZ
对数据量要求	压缩前：不超过2.35MB 压缩后：不超过1.90MB

5.3 智能驾驶电子地图数据格式

智能驾驶电子地图数据格式采用（XML）文件格式的数据组织方式，其基于国际通用的OpenDrive规范，并根据本标准所针对的自动驾驶业务需求拓展修改而成。主要改动和扩展了以下几个方面：

- 地图元素形状的表述方式。以车道边界为例，标准OpenDrive采用基于Reference Line的曲线方程和偏移的方式来表达边界形状，而本标准对应的OpenDrive采用绝对坐标序列的方式描述边界形状；
- 元素类型的扩展。例如新增了对于禁停区、人行横道、减速带等元素的独立描述；
- 元素之间相互关系的描述扩展。比如新增了junction与junction内元素的关联关系等；
- 配合无人驾驶算法的扩展。比如增加了车道中心线到真实道路边界的距离、停止线与红绿灯的关联关系等。

OpenDrive节点主要包括以下几个部分：

- Header节点：包括Geo Reference节点；
- Road节点：主要包括RouteView节点、Road Link节点、Road Lanes节点、Road Objects节点；
- Junction节点：主要指Junction outline、Junction Connection、Junction Object Overlap Group等部分。

改动和扩展后的规格在实现上更加的简单，同时也兼顾了无人驾驶的应用需求。

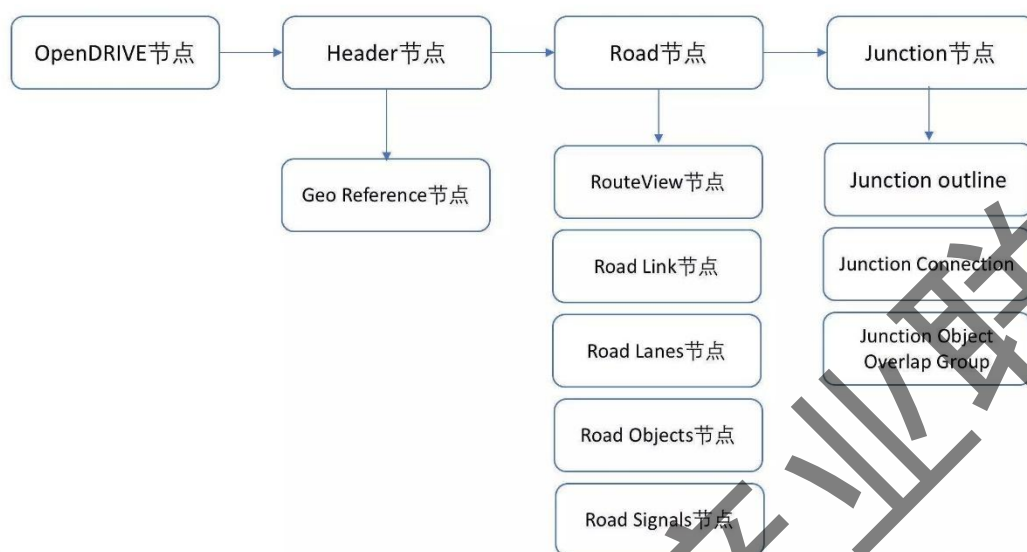


图3 智能驾驶电子地图数据组织形式

5.4 数据模型

智能驾驶电子地图的数据模型主要分为四大类，分别是道路模型（Road Model）、车道模型（Lane Model）、道路标记模型（Road Mark）、基本对象模型（Object）。这四大类模型覆盖了从地面道路信息、行驶车道信息、沿路标志信息等整个自动驾驶地图的基础先验数据库。

智能驾驶电子地图要素的数据模型具体可见附录表A.1所示。

5.4.1 道路模型

道路几何形状是指数据制作时，形状点连接成的道路的几何形状、通过形状点描述道路的几何形状时，形状点以坐标形式进行描述，如道路中心线与道路拓扑等。

曲率表示道路的弯曲程度，弯曲程度越大曲率值越大。计算时，采取曲线拟合的方法，得到各个形状点所在的曲率半径的倒数，根据道路的几何形状，进行曲线拟合后计算出离散点的曲率值。

坡度是指道路纵向的起伏程度，道路起伏程度越大说明坡度越大。计算时，对形状点高程差与水平距离取反正切计算得到坡度。

5.4.2 车道模型

车道类型指地面道路上该车道的类型，主要包括普通车道、入口车道、出口车道、进入匝道、退出匝道、应急车道、连接匝道等。

普通车道指无特殊属性的车道，一般为主行车道。普通车道普遍指高速中的主路，按照实际的车道形态进行制作表达，并在右侧车道线上赋值主路属性。车道类型会赋值在该车道上。

5.4.3 道路标记模型

道路标记模型主要指车道线的样式，包括无属性、单实线、长虚线、双实线、左实右虚线、右实左虚线、双虚线、路沿线、护栏线。道路标线会赋值在对应的车道线上。

5.4.4 基本对象模型

智能驾驶电子地图中对象的类型包括杆、牌、龙门架、地面标线等。其中杆类型中包括灯杆、基站杆、摄像头杆、交通标志牌依附的杆等。而地面标线可细分为多个子类型，如地面箭头、地面文字、导流区、地面限速等。

对象表达为一个能够容纳整个对象的包围盒，该包围盒按照对象的外切线将对象完全包围，一个对象对应一个包围盒，且包围盒属性与对象的属性对应。制作范围为道路两侧和上方，两侧制作范围一般为道路横向外20cm，仅制作隶属于道路的对象。

6 地图分发策略

如第5章所述，本标准规定的智能驾驶电子地图应用场景包括云端——>边端和边端——>车端两个地图分发场景。具体流程为：自动驾驶车辆在进入RSU电子围栏后发出地图请求，云平台将切分完成的智能驾驶电子地图分发到边端服务器，再由边端服务器通过RSU、OBU分发到车端，从而获取最新的地图，开始自动驾驶模式。

本章主要针对地图分发策略进行详细说明。

6.1 分发策略

6.1.1 全量分发

本文件的实际测试采用全量分发策略。

全量分发是通过4G/5G网络对车辆进行全量地图更新，即导航电子地图厂商根据地图要素的变化所实现的对原有地图的整体更新。全量分发可分为整体分发和基于Tile切分的分发方式。由于全量分发需替换全幅地图数据或分发ODD范围内所有Tile的数据，故相对于增量分发方式而言，全量分发处理的数据量大，成本较高。但全量数据的后续更新只需覆盖式整体替换即可实现，因此其数据结构、数据存储以及数据质量控制比较简单。基于全量分发的优势，本标准对应的实际测试选用全量分发。

6.1.2 沿途分发

本文件规定的智能驾驶电子地图分发策略采用路侧沿途下发的方案，即沿着自动驾驶车辆的起讫点所确定的行驶路线进行多次分发。这种分发方式使得每次分发地图的数据量小，可实时更新，实现边走边发。另外车内地图拼接过程简单、迅速，且车辆地图更新可通过PC5接口下载，节省流量费用，降低成本。综上，本标准提出的智能驾驶电子地图路侧分发方案具有明显的优势。

下面是沿途下发方案的具体说明。

图4是沿途路侧分发方案的示意图，汽车行驶路线为P1-P2-P3-P4-P5，图中共有12个小方块，每个方块称为1个Tile，单个Tile大约4km²，而一个RSU电子围栏是半径300m的圆形区域，面积为0.28 km²，故一个Tile内会有多个RSU，二者是一对多的关系。后续沿途RSU作地图版本检查，异常情况发生时负责下发地图，保证了地图分发的效率和准确率。

沿途路侧地图分发的具体流程为：

- a) 获取地图图幅ID：根据OD点确定自动驾驶车辆的行驶路径，从而明确需要获取的智能驾驶电子地图的图幅ID；
- b) 地图分发至RSU：路侧设备RSU应能存储一定范围内智能驾驶电子地图。如第5章所述，地图沿云平台——>边端——>RSU的路径下发并储存在RSU中；
- c) 车辆数据接收：当自动驾驶车辆进入RSU电子围栏范围后，向云平台发出数据请求，接收需要的地图图幅ID的数据，走到下一个范围时，删除前一刻的图幅数据，直至进入下一个RSU范围接收下一个图幅数据。因此，车上永远只保持1-2个图幅数据，故本地图切分方案具有数据存储量小、更新速度快的优势。

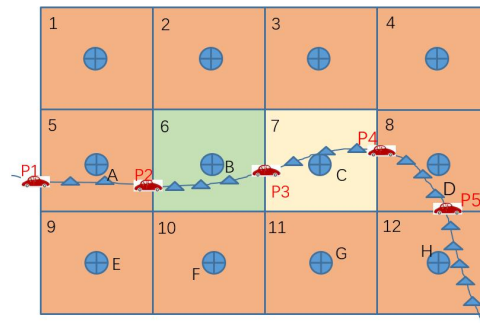


图4 沿途路侧分发方案

6.2 整合策略

本节定义说明整合策略的通信协议、数据编译、分包及重传机制等。

6.2.1 通信协议

实现RSU和OBU的通信，需要使用两个AID：AID1和AID2。RSU在AID1上按照一定的频率广播自己拥有的电子地图的TileID和版本号。OBU在AID1上接收该广播的消息，并且对比自己设备上电子地图的TileID和版本号，当存在更新的版本时，在AID2上下载最新的电子地图。AID2上下载地图时，以OBU（client端）发送请求特定TileID的地图为起始时刻。

6.2.2 数据编译

进行数据编译时，各标识对应的含义如附录表A.2所示。

6.2.3 分包及重传机制

智能驾驶电子地图分发及重传机制的具体步骤如下：

a) Client端/OBU端：请求下载TileID的地图；

Command: REQ	TileID
--------------	--------

b) Server端/RSU端：接收到1的REQ后，发送对应申请的TileID的电子地图的基本信息。超时k秒未收到ACK，重发，最多2次，后退出；

Command: FILEMSG	FileSize	Packetnum	fileCRC
------------------	----------	-----------	---------

c) Client端/OBU端：接收到FILEMSG后，准备维护一个丢包序列，创建一个接收的文件，返回ACK_FILEMSG，并开始仅接收DATA数据包，当收到FILEEND数据包时停止接收DATA；

Command: ACK_FILEMSG	FileSize	Packetnum	fileCRC
----------------------	----------	-----------	---------

Command:FILEEND

d) Server端/RSU端：接收到3的ACK_FILEMSG后，发送分片文件数据DATA包。直到发送完毕，发送END包；

Command:DATA	PacketID	FilePos	PacketLen	CRC	Data
--------------	----------	---------	-----------	-----	------

Command:FILEEND

- e) Client端/OBU端：每收到一个验证通过的DATA包，在维护的丢包序列里面去除该PacketID，并将该包写入file对应的FilePos位置里。当收到FILEEND包，检查丢包队列，若无丢包，则发送ACK_FILEEND。结束通信或发送新的REQ；

Command:ACK_FILEEND

若有丢包，则发送ACK_RESEND包。超时未收到RESEND包，则至多重传2次。

Command: ACK_RESEND	PacketID	FilePos	PacketLen	CRC
---------------------	----------	---------	-----------	-----

- f) Server端/RSU端：超时未收到ACK_FILEEND或ACK_RESEND消息，则重传FILEEND，至多两次。当收到ACK_FILEEND的时候，等待REQ或REQ_DOWNLOAD。当收到ACK_RESEND时，发送对应的包：

Command: RESEND	PacketID	FilePos	PacketLen	CRC	Data
-----------------	----------	---------	-----------	-----	------

- g) Client端/OBU端：每收到一个RESEND包，在维护的丢包序列里面去除该FrameID，并将该包写入file对应的FilePos位置里。检查丢包队列，若无丢包，则发送ACK_FILEEND，结束通信；否则发送ACK_RESEND包；
- h) Server端/RSU端：接收到ACK_FILEEND，回到等待1. REQ的过程。

以上步骤可用流程图表示，如图5所示：

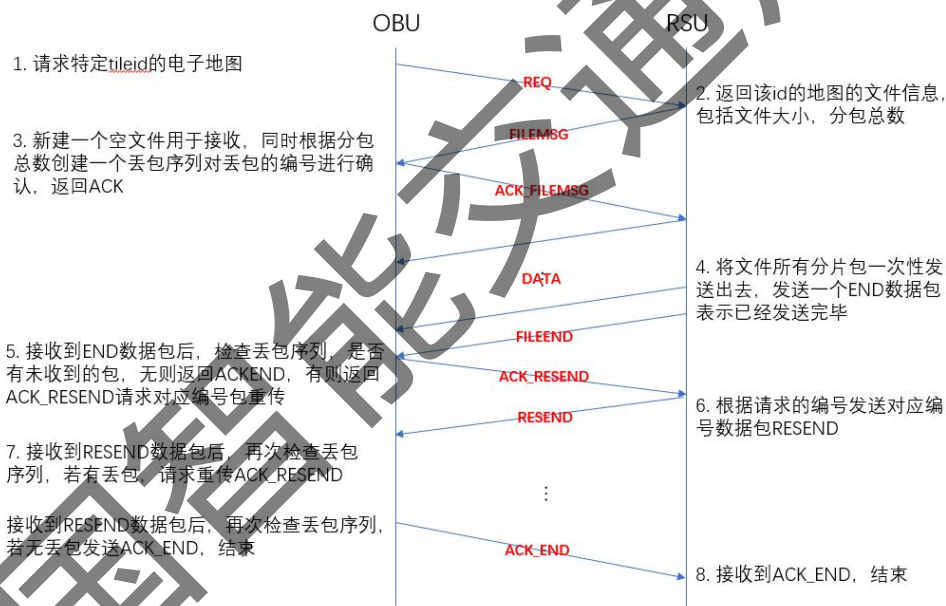


图 5 PC5 数据下发分包及重传流程图

6.2.4 优化策略

为有效地提高分包及重传效率，实际测试中采取了如下优化措施：

- a) 拆包时，单包数据传输前增加checksum。传输完成后先校验，当校验成功再解压缩；
- b) 设置OBU端接收超时的消息timeout；
- c) 针对OBU接收回复增加超时或者异常的情况，增加设置对应的异常回复消息类型；
- d) 增加最大重发次数设置；
- e) 每包数据大小、发送间隔时间等参数都从同一个配置文件里读取；

- f) 增加压力测试脚本，脚本中动态修改配置文件中的数据包大小和间隔时间等参数，通过不同参数组合的下发数据成功率，找到最合理的参数配置；
- g) 增加地图版本查询字段。

6.3 分发路径

本节定义说明数据稳定支持分发的路径，包括分发对象、通讯方式、分发场景、数据配置要求等。智能驾驶电子地图分发路径包括云端-车端和边端-车端，整个下发流程为云平台——>边端服务器——>RSU——>车端，具体可参考第5章和6.1节的内容。分发路径的数据端和数据传输方式如表2。

表2 应用场景数据流

数据流向		数据流传输方式
数据流出端	数据流入端	
云端：云平台	边端：GPU 服务器	光纤/有线以太网
边端：GPU 服务器	路端：RSU	以太网
路端：RSU	车端：OBU	PC5

6.4 分发数据

本标准支持分发的数据类型为智能驾驶电子地图数据，采用XML文件格式的数据组织方式。可扩展标记语言XML是一种标记语言，用于以人类可读和机器可读的格式对文档进行编码。这种文本数据格式的好处是通过 Unicode 为不同的人类语言提供强大的支持，并且广泛用于表示任意数据结构。

智能驾驶电子地图数据类型基于国际通用的OpenDrive规范，并根据本标准所针对的自动驾驶业务需求拓展修改而成。OpenDrive 是一种描述道路网络逻辑的开放格式规范，其目标是规范逻辑描述整个道路网络，以方便不同驾驶模拟器之间的数据交换。

关于改动和扩展的内容，具体可参考5.3节。OpenDrive源文件裁剪优化后大概749KB。

6.5 分发能力

为研究数据应用端之间地图分发能力、服务稳定性、分发速度、单次分发数据量等，本标准针对不同分发距离、分包大小、发包频率等参数，进行了多次静态电子地图传输测试，并考虑文件压缩对地图分发能力的影响，形成最优的分包、频率、重传次数配比，得出可靠性的结论。

6.5.1 基础能力测试

在正式的压力测试之前，首先进行路测到车端的地图分发基础能力测试，目的是对地图分发能力有初步的了解和判断。故基础能力测试组数少，但分发大小和发送频率范围较大，有利于确定两者是否存在上下限值。

6.5.1.1 测试步骤

- a) 两台设备静止不动，相距50m, 100m, 300m（目测，不精准）共做三部分的测试；
- b) 一台设备作为server，控制其每包大小和发送频率；另一台设备作为client，发送请求下载server上的文件（tileid 1: send.xml.gz- 5.6Mb）；
- c) 记录不同距离，不同包大小，不同发送频率下，文件下载完成的时间。

6.5.1.2 测试结果

测试结果如表3所示：

表3 基础能力测试结果

每包大小/bytes	发送频率/Hz	距离/m	下载完成的时间	备注
8000	10	50	1min17s	无重传
		100	1min18s	无重传

		300	1min20s	有个位数的包重传
8000	100	50	1min17s	约有 70%的包重传
		100	1min17s	约有 70%的包重传
		300	1min30s	约有 90%的包重传
4000	20	50	1min17s	无重传
		100	1min17s	无重传
		300	1min21s	有少量重传
4000	10	50	1min17s	无重传
		100	1min17s	无重传
		300	1min18s	无重传
4000	100	50	1min35s	约有 80%的包重传
		100	1min32s	约有 80%的包重传
		300	1min40s	约有 90%的包重传

6.5.1.3 结论

- 存在分发数据速度上限约80Kb/s:结合此次文件传输测试以及上次发包频率和发包大小的两次测试结果来看,对于8000bytes * 10Hz,和4000bytes * 20Hz来说,结果是相近的。因此存在一个上限值约为80kb/s,当发包的设置超过该上限值时,会有丢包且需要重传,反之当小于这个上限值时,收包的丢包率几乎为0。
- 丢包率受多种因素影响:设备丢包率实际上还受到距离,障碍物遮挡,信道忙率等多种因素影响。在距离较远,障碍物较多,上百个设备通信的大规模场景中,丢包会较严重。

6.5.2 压力测试

在完成基础能力测试之后,对发包频率和发包大小的水平进行细分,在不同距离下设置多个组合,进行压力测试。

6.5.2.1 测试步骤

- 选择两台设备静止不动,分别相距10m, 150m, 300m进行静态地图传输压力测试;
- 其中一台设备作为server,另一台设备作为client,发送请求下载server上的文件(tileid1: send.xml.gz-5.6Mb);
- 记录不同的距离、发包大小、发包频率下,文件下载完成的时间和丢包重传次数。

6.5.2.2 测试组合

进行压力测试的分包大小、频率组合、总数据包等参数的组合如表4所示,并记录每一组的下载完成时间和重传次数:

表 4 压力测试组合

每包大小/bytes	发送频率/h	总数据包	距离/m
8000	10,20, ……100 共 10 个频率	705	10, 150, 300
7000	10,20, ……100 共 10 个频率	806	10, 150, 300
6000	10,20, ……100 共 10 个频率	940	10, 150, 300
5000	10,20, ……100 共 10 个频率	1128	10, 150, 300
4000	10,20, ……120 共 12 个频率	1410	10, 150, 300

3000	10,20, ……130 共 13 个频率	1880	10, 150, 300
2000	10,20, ……130 共 13 个频率	2820	10, 150, 300

6.5.2.3 测试结果

由于实际测试数据较多，在此不详细列出每一组测试结果，仅列出由结果得出的推荐分包大小和频率组合，如表5所示。

表5 发包大小和频率组合推荐

远 (200m 以上或遮挡较多)	近 (150m 内遮挡较少)	5.6MB 文件传递完成大致时间
8000*30Hz	8000*40Hz	26~39s
7000*30Hz	7000*40Hz	29-35s
6000*30Hz	6000*40Hz	26-40s
5000*40Hz	5000*50Hz	25-45s
4000*40Hz	4000*50Hz	31-38s
3000*70Hz	3000*80Hz	26-35s
2000*110Hz	2000*110Hz	28-30s

6.5.2.4 结论

根据测试结果，有以下结论：

- 对于不同大小的包，有最适合的发包频率，使下载时间最短，丢包重传次数较少。此外，下载时的距离会对丢包重传次数有影响，距离越大，传输时丢包重传次数越多，此时下载时间也会变长。综合考虑下载时长和距离的影响，得出推荐的发包大小和频率组合，记录在表7。
- 由于此次测试环境基本是在无过多遮挡的情形下完成的，在设备较多，遮挡较多，环境较为复杂的情况下，远距离下载时丢包率可能会更高，下载时间会更长。考虑到这种情况，在表7的推荐组合的基础上，可以略微下调各个发包频率，从而保证更加有效的传输。
- 对于一些表现较好的情形，以下特别列出：

表6 部分表现优秀组合测试结果

距离	发包大小和频率组合	5.6MB 文件传递完成大致时间
10m	5000*80Hz	16s
10m	5000*90Hz	17s
150m	8000*30Hz	26s
300m	3000*80Hz	28s

6.5.3 压缩前后对比测试

为比较地图文件压缩分包和未压缩分包的分发完成时间的差异，设置对比测试。

6.5.3.1 测试步骤

- 两台设备静止不动，相距10m, 150m, 300m共做三个距离的压力测试测试；
- 一台设备作为server；另一台设备作为client，发送请求下载server上的文件。此次测试server发送两个类型的文件，一个是未压缩的原地图文件（约749kB），一个是使用rar压缩后的文件（约80KB）。client端对发来的压缩文件，发起请求至解压完毕的时间作为下载完成的时间；对未压缩的原地图文件，直接统计下载完成的时间；
- 记录每个距离，每个发包大小，发包频率下，文件下载完成的时间和丢包重传次数。

6.5.3.2 测试组合

压缩前后对比测试的组合如下表所示：

表 7 压缩前后对比测试组合

每包大小/bytes	发送频率/Hz	距离/m	不压缩总数据包	压缩后总数据包
8000	10, 20, …… , 100 共 10 个频率	10,150,300	96	9
7000	10, 20, …… , 100 共 10 个频率	10,150,300	109	11
6000	10, 20, …… , 100 共 10 个频率	10,150,300	128	13
5000	10, 20, …… , 100 共 10 个频率	10,150,300	153	15
4000	10, 20, …… , 100 共 10 个频率	10,150,300	192	19
3000	10, 20, …… , 100 共 10 个频率	10,150,300	255	25
2000	10, 20, …… , 100 共 10 个频率	10,150,300	383	38

有关测试记录的说明如下：

- 分包大小的分组为：2000bytes，3000bytes，4000bytes……，8000bytes，共七个水平；
- 发送频率的分组为：10Hz，20Hz，30Hz，……，100Hz，共10个水平；
- 还需统计每次测试的不压缩重传次数、压缩下载丢包重传次数等指标。

6.5.3.3 测试结果

通过比较地图压缩前后的完成时间，来分析两者分发能力的差异。

由于实际测试数据较多，在此不详细列出。主要有以下结果：

- 对比压缩文件以及不压缩文件发送下载的时间，压缩发送并在Client端解压的效率更高，平均压缩时间为45ms，平均解压缩时间16ms，基本下载完成的时间仅为26秒；
- 300m无测试记录，原因是此次测试场所(T字路口)无300m直道，而300m弯道距离由于遮挡过多无法收到消息，RSU在地上或许有一定的影响，此次测试极限距离大约就为150m；
- 存在特殊情况：即便收到所有的包，但由于文件CRC校验时出错，Client会重新发送这个文件下载的Request。

6.5.3.4 结论

从测试结果中，可以得出结论：server端发送压缩文件至client端接收并解压所用的时间更短，地图分发能力更强，效率更高。

6.6 OBU 测试

为验证市面上的OBU设备是否满足地图分发的要求，选择三种不同OBU设备进行测试。结果如下所示：

6.6.1 OBU1

直道：

表 8 OBU1 直道测试结果

距离/m	收包时间/min	收包数	丢包率
10	1	306	0
50	1	306	0
100	1	109	0.64
	1	281	0.082
200	1	283	0.075

300	1	293	0.042
370	1	189	0.382

弯道：

表 9 OBU1 弯道测试结果

有无遮挡	距离/m	收包时间/min	收包数	丢包率
有遮挡-树木	50-35	1	0	0
	35-35	1	86	0.719
有遮挡-楼房	50-50	1	0	0
	50-20	1	29	0.905

6.6.2 OBU2

直道：视距情况下无遮挡

表 10 OBU2 直道测试结果

距离/m	发包数量/个	平均时延/ms	丢包率
100	1000	21.758	0
200	1000	21.613	0
300	1000	21.780	0.1%

弯道：非视距情况下，中间有一片树林作为遮挡物

表 11 OBU2 弯道测试结果

距离/m	发包数量/个	平均时延/ms	丢包率
100	1000	21.699	0
200	1000	21.721	0
300	1000	21.798	0.5%

6.6.3 OBU3

RSU 和 OBU 距离在 250 米到 300 米，基本没有丢包，时延根据距离不同在 5~8ms 之间。在多用户背景压力测试的条件下，V2V 同向，间隔 30 米，统计丢包率 0.68%，时延 20ms~30ms。

6.6.4 结论

综合以上结果，可以得出以下结论：

- 直道的透传能力大于弯道；
- 距离增加以及遮挡物的存在，将使得丢包率增加；
- 尽管测试的集中 OBU 设备各有优劣，但均满足实际使用需求。

因此，常用的 OBU 设备均能满足本标准针对的智能驾驶电子地图的分发需求。

7 地图解析验证

7.1 地图启动

为了验证实际场景下智能驾驶电子地图的可用性，需要对地图文件进行解析验证。首先启动本标准针对的智能驾驶电子地图。

```

albert@in_dev_docker:/apollo$ bash apollo.sh build
System check passed. Build continue ...
[WARNING] ESD CAN library supplied by ESD Electronics does not exist. If you need ESD CAN, please refer to third_party/can_card_library/esd_can/README.md.
[INFO] Start building, please wait ...
INFO: Reading 'startup' options from /apollo/tools/bazel.rc: --batch_cpu_scheduling
[INFO] Building on x86_64...
INFO: Reading 'startup' options from /apollo/tools/bazel.rc: --batch_cpu_scheduling
INFO: Analysed 1660 targets (8 packages loaded).
INFO: Found 1660 targets...
INFO: Elapsed time: 22.498s, Critical Path: 12.66s
INFO: Build completed successfully, 35 total actions
fatal: Not a git repository (or any parent up to mount point /apollo)
Stopping at filesystem boundary (GIT_DISCOVERY_ACROSS_FILESYSTEM not set).
=====
[ OK ] Build passed!
[INFO] Took 27 seconds
=====

```

图 6 地图启动

7.2 配置及读取设置

新建new_map，地图文件命名为apollo_map.xml，添加配置挂载地图命名为new_map，读取文件为apollo_map.xml。

```

73 # configure file: modules/common/data/global_flagfile.txt
74 # Add new lines |
75 --base_map_filename=/apollo_map.xml
76
77 --map_dir=/apollo/modules/map/data/demo
78
79 --map_dir=/apollo/modules/map/data/new_map

```

图 7 配置及读取设置

7.3 解析测试

为了使用智能驾驶电子地图框架下Map模块中的方法opendrive_adapter.cc解析并读取文件，方法调用Adapter中的xml_parser针对道路的不同元素部分进行解析。

```

1 modules/map/hd_map/opendrive_adapter.cc
2 OpendriveAdapter::LoadData(const std::string& filename,apollo::hdmap::Map* pb_map)

```

图 8 方法调用

```

├─ adapter
│  ├─ BUILD
│  ├─ coordinate_convert_tool.cc // 坐标转换工具
│  ├─ coordinate_convert_tool.h
│  ├─ opendrive_adapter.cc // 加载opendrive格式地图
│  ├─ opendrive_adapter.h
│  ├─ proto_organizer.cc //
│  └─ proto_organizer.h
├─ xml_parser // xml_parser针对道路的不同元素做相应解析
│  ├─ common_define.h
│  ├─ header_xml_parser.cc
│  ├─ header_xml_parser.h
│  ├─ junctions_xml_parser.cc
│  ├─ junctions_xml_parser.h
│  ├─ lanes_xml_parser.cc
│  ├─ lanes_xml_parser.h
│  ├─ objects_xml_parser.cc
│  ├─ objects_xml_parser.h
│  ├─ roads_xml_parser.cc
│  ├─ roads_xml_parser.h
│  ├─ signals_xml_parser.cc
│  ├─ signals_xml_parser.h
│  ├─ status.h
│  └─ util_xml_parser.cc

```

图9 解析测试

7.3.1 地图版本 1

对第一个版本地图进行读取解析测试，操作及结果如图 10 所示：

```
albert@in_dev_docker:/apollo$ ./bazel-bin/modules/map/tools/proto_map_generator
I0112 16:14:39.906782 14667 proto_organizer.cc:173] cannot find signal object_id:610000000000130115
I0112 16:14:39.908744 14667 proto_organizer.cc:390] hmap statistics: roads-44,lanes-44,crosswalks-6,parking spaces-0,clear areas-0,speed bumps-3,signals-3,stop signs-0,yield signs-0,junctions-2,overlaps-37
I0112 16:14:39.951349 14667 proto_map_generator.cc:57] load map success
```

图 10 方法调用

报错误缺失 id 为 610000000000130115 的信号灯元素 (cannot find signal object_id: 610000000000130115)，其它元素正常解析并成功加载 (load map success)。元素包括道路 (roads)、车道 (lanes)、人行横道 (crosswalks)、限速标志 (speed bumps)、信号灯 (signals)、交叉路口 (junctions) 以及重叠元素 (overlaps)。

7.3.2 地图版本 2

对修改后的第二个版本地图进行读取解析测试，操作及结果如图11所示：

```
albert@in_dev_docker:/apollo$ ./bazel-bin/modules/map/tools/proto_map_generator
I0113 16:08:13.717027 6395 proto_organizer.cc:390] hmap statistics: roads-44,lanes-44,crosswalks-6,parking spaces-0,clear areas-0,speed bumps-3,signals-4,stop signs-0,yield signs-0,junctions-2,overlaps-38
I0113 16:08:13.758198 6395 proto_map_generator.cc:57] load map success
```

图 11 地图版本 2 解析测试

程序结果显示所有元素均被成功读取和解析，地图能被正常加载。

7.4 解析结果

道路拓扑解析结果如图12所示：

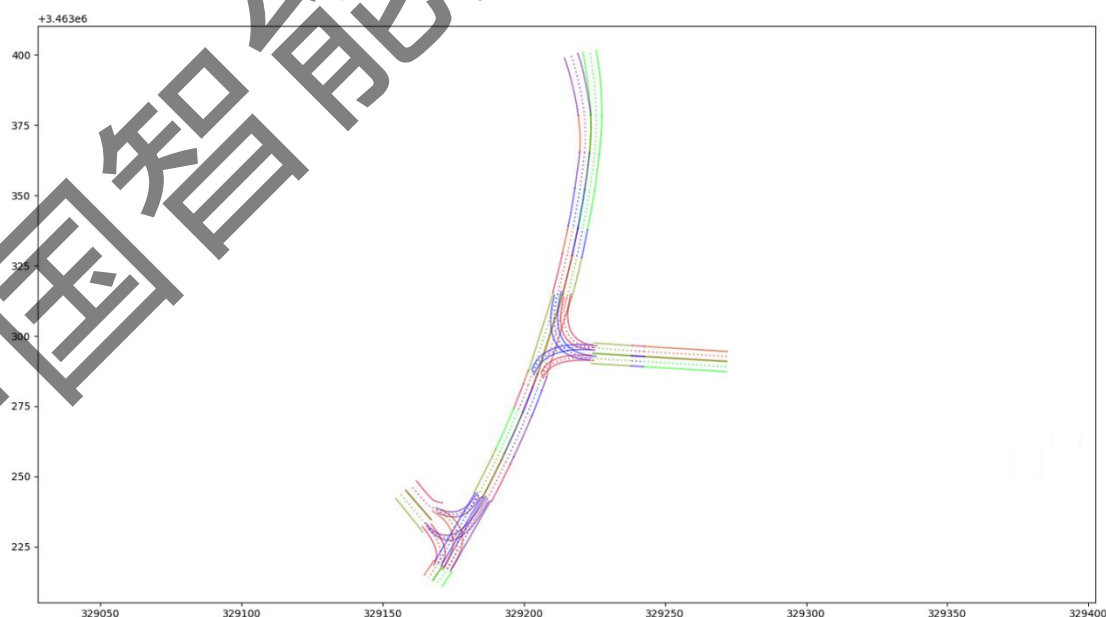


图 12 道路拓扑解析

对地图提取车道ID元素，结果如图13和图14所示：

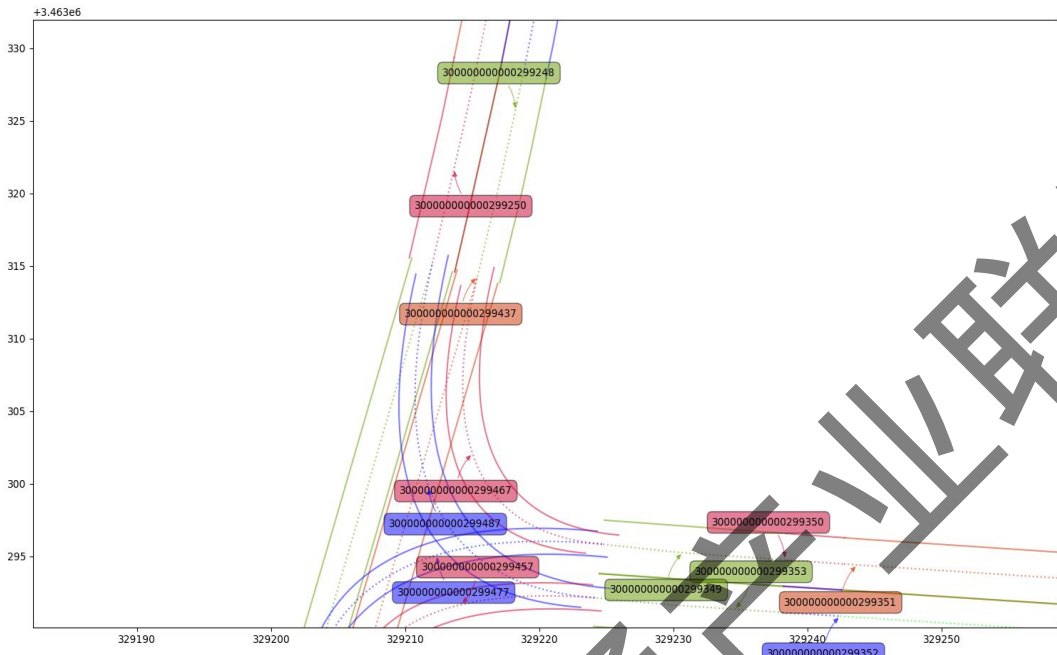


图 13 车道 ID (1)

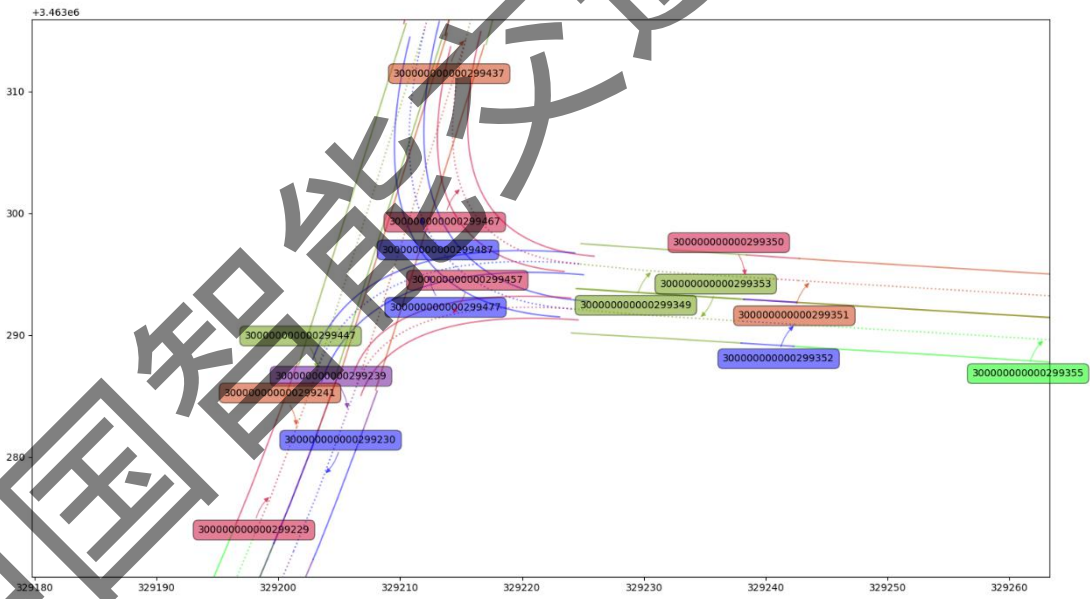


图 14 车道 ID (2)

7.5 结论

本次解析测试验证了智能驾驶电子地图可被正常读取与解析。

附录 A
(规范性)
压缩前后数据量要求的详细说明

A.1 RSU 电子围栏

如图，电子围栏半径实测为300m，将整个圆形区域分成两部分（ r 为到RSU的距离）：

① $200m \leq r \leq 300m$ ：车辆行驶在该区域内，RSU需完成地图的分发同时车端完成地图的下载。如图5.3中的P1~P2阶段。

② $0 \leq r \leq 200m$ ：车辆在该区域内行驶时，RSU需完成国标消息集的发送。

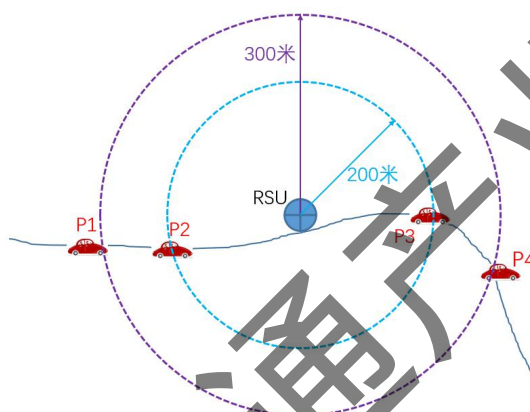


图 C.1 车道 ID (2)

A.2 地图分发时间

实测车辆行驶速度在 $0 \sim 60km/h$ 之间，即 $0 \sim 16.7m/s$ ，那么车辆在P1~P2范围内行驶的最短时间 t_{min} 为：

$$t_{min} = 100 / 16.7 = 6s$$

因此地图分发所需时间应不少于 $6s$ 。

A.3 数据更新频率

本标准实测时选择的数据更新频率范围是 $10Hz \sim 80Hz$ ，故可合理选择中间频率 $50Hz$ 作为计算值。

A.4 压缩前数据量阈值

数据下载速度为 $400Kb/s$ （即 $8Kb \times 50Hz$ ），地图未压缩所对应的下载时间为 $6s$ ，故最大下载数据量为

$$\frac{400Kb/s \times 6s}{1024} = 2.35Mb$$

因此，压缩前的地图数据量阈值为 $2.35Mb$ 。因此 $2.35Mb$ 可用于判定是否需要进行地图分包。

A.5 压缩后数据量阈值

实测压缩时间 $0.9s$ ，解压缩时间 $0.24s$ ，则最大下载时间为

$$6s - 0.9s - 0.24s = 4.86s$$

此时，最大下载数据量为

$$\frac{400Kb/s \times 4.86s}{1024} = 1.90Mb$$

因此，压缩后的地图数据量阈值为1.90Mb。

中国智能交通产业联盟

附 录 B
(规范性)
智能驾驶电子地图要素的数据模型

表B.1给出了智能驾驶电子地图要素的数据模型。

表 B.1 智能驾驶电子地图要素的数据模型

主 题	要 素	英文说明
道路模型	道路中心线	Road Centerline Geometry
	道路拓扑	Road Topology
	隧道	Tunnel
	收费站	Toll Station
	曲率	Curvature
	坡度	Slope
车道模型	车道边界线	Lane Marking
	车道数	Lane Number
	车道宽度	Lane Width
	车道类型	Lane Type
	车道线颜色	Lane Marking Color
	车道线线型	Lane Marking Style
	车道线材质	Lane Marking Material
	车道线宽度	Lane Marking Width
	车道在建信息	Lane Construction State
	道路连接	Junction
	车道拓扑	Lane Topology
	护栏	Barrier
	路沿	Curb
	道路标记模型	地面限速
箭头		Arrow
文字		Words
导流区		Diversion Zone
基本对象模型	限速	Speed Limit
	禁止超车	Overtaking Prohibited
	线型诱导	Delineator
	其他标牌	Other Signs
	杆	Pole
	龙门架	Gantry
	跨线桥	Overpass

附录 C
(规范性)
电子地图数据编译信息标识

表C.1给出了电子地图进行数据编译时的信息标识。

表 C.1 信息标识

名称	含义	取值	备注
Command	消息类型	int 枚举	必选。包含 DATA、ACK 等
TileID	地图块标识	int	
FileSize	传输文件大小	int	
Packetnum	总数据包个数	Int	
fileCRC	文件的 CRC 校验	Unsigned int	
PacketID	当前文件分片标识	int	
FilePos	当前数据块的文件位置	Int	
PacketLen	数据部分长度	Int	
CRC	校验部分	Int	
Data	数据部分	字节串	

参 考 文 献

- [1] GB/T 2312 信息交换用汉字编码字符集 基本集
- [2] GB/T 31024.1—2014 合作式智能运输系统 专用短程通信 第1部分：总体技术要求
- [3] T/CSAE 53-2017 合作式智能运输系统 车用通信系统 应用层及应用数据交互标准
- [4] CJJ 37—2012 城市道路工程设计规范（2016年版）
- [5] IMT-2020(5G) C-V2X白皮书
- [6] 道路交通车路协同信息服务通用技术要求
- [7] 国家车联网产业标准体系建设指南

中国智能交通产业联盟



中国智能交通产业联盟
标准
智能驾驶电子地图路侧分发机制
T/ITS XXXX-20XX

北京市海淀区西土城路 8 号（100088）
中国智能交通产业联盟印刷
网址：<http://www.c-its.org.cn>

20XX 年 XX 月第一版 20XX 年 XX 月第一次印刷