# 背景

交通系统对国家的政治经济、文化教育、科学技术等方面的发展影响极大，也是城市建设的一个非常重要的方面。交通的便捷程度已经成为衡量一个城市乃至一个国家发展程度的重要标志。

近年来，随着世界经济的飞速发展，车辆在世界范围内得到广泛的普及，车辆的普及为人们的出行带来了极大的便利，但与此同时，各种问题也应运而生。各大城市严重的交通拥堵、频发的交通事故、低下的运输效率、燃料资源的严重浪费以及由此造成的环境污染已成为重要的全球性问题。发达国家和发展中国家都受到交通问题的困扰。频发的交通事故将造成巨大的生命和财产损失。据世界卫生组织（WHO）的统计，全球超过1亿人死于交通事故，每年由此造成的经济损失高达5000亿美元。而且，城市交通拥堵使运输系统的效率下降，从而阻碍了经济的增长。因此，目前交通系统的安全和效率都有较大的提升空间。

车联网的概念引申自物联网（Internet of Things, IOT），物联网的概念自从1999年首次被提出之后，立即在世界范围内得到迅速的发展。因此物联网也被誉为继通信、计算机、互联网之后的第三次信息发展浪潮。而车联网（IOV, Internet of Vehicle），作为物联网的重要应用之一，在2005年被提出之后，迅速成为学术领域的研究热点。车联网是车辆工程与通信工程、网络技术的高度融合，它是实现智能交通系统（Intelligent Transportation System, ITS）的重要手段与理论基础。

车联网旨在通过车对车（Vehicle-to-vehicle, V2V）以及车对基础设施（Vehicle-to-infrastructure, V2I）的通信来提供信息交换。据报道，超过50％的受访消费者对汽车通信有非常浓厚的兴趣，其中，将近22％的受访者愿意每月支付30至65美元用于车联网的增值服务。1999年，FCC（Federal Communications Commission）分配了75 MHz（从5.850 GHz到5.925 GHz）专用短距离通信（Dedicated short-range communications, DSRC）带宽。美国交通部（U.S. Department of Transportation, DOT）估计，基于DSRC的V2V通信可以减少美国所有道路交通事故的82％，这将会拯救数千人的生命，同时挽回数十亿美元的经济损失。另外，移动蜂窝网络对车载业务的适用性也给予了极大的支持，为车载用户提供广覆盖和高数据速率的业务。

车载自组织网络（Vehicular ad-hoc network, VANET）是无线网络的一个新兴领域，通过车辆与车辆（V2V）之间的通信，车辆与城市或道路基础设施（V2I）之间的通信，以构建起一个实时、高效、可靠的交通信息网络。这个新兴技术领域旨在提高乘客的安全性，减缓交通流量，减少污染，并为乘客提供丰富多样的车载娱乐应用。通过向驾驶员提供交通路况信息，例如碰撞警告，路面状况或交通状况汇报，可以协助驾驶员提前做出相应的预防措施，从而减少事故的发生。此外，乘客可以利用城市可用的基础设施连接互联网进行娱乐应用。

通常，在车辆通信的研究中存在两种情景，即非视距（NLOS）和视距（LOS）情景。在城市或曼哈顿街区模型中，存在着大量诸如建筑物和树木等障碍物，因此街区场景中的通信被认为是NLOS，而高速场景中的通信被认为是LOS的。目前用于V2V通信的传统解决方案依赖于IEEE 802.11p标准中的自组织通信。由于车辆通信环境的动态特性以及较为严格的服务质量（QoS）要求（如可靠性和延迟），这些传统解决方案并不能直接应用于新兴的V2V场景。因此，提出一个高效可靠的无线网络解决方案是我们目前面临的严峻挑战。

为了解决上述难题，学术界先后提出了几个关于V2V资源分配的解决方案。基于二分组匹配的V2V通信的低复杂度中断优化分布式信道分配。基于干扰图的资源分配方案，以提高网络吞吐量。基于情境感知聚类机制的方案，以优化车辆之间的合作。基于启发式位置依赖资源分配方法。

# 研究内容

该部分主要介绍LTE-V2X仿真平台的设计以及实现，仿真平台从功能上可以划分为场景单元、无线资源管理单元、路由单元、无线传输单元、业务模型与控制单元。

### 整体架构与功能概述

LTE-V2X系统级车联网仿真平台模拟了车辆与车辆、车辆与RSU之间的通信场景，并输出时延信息，丢包率等多项统计数据。利用Visual Studio开发工具以及C++编程语言进行搭建。该仿真平台从功能上可以划分为以下几个功能单元。

* 地理拓扑与传播单元
* 无线资源管理单元
* 无线传输单元
* 路由单元
* 业务模型与控制单元

每个功能单元面向接口编程，提供了多个不同的具体实现。例如，地理拓扑与传播单元包括曼哈顿街区模型以及高速公路模型两种；无线资源管理单元包括时分、频分、基于地理位置、基于载波侦听等多种不同的具体实现；路由单元可以选择可靠转发模式与非可靠转发模式两种。

LTE-V2X仿真平台采用时间驱动的模型，最小时间粒度为1个TTI，在每个TTI时刻，调用相应的功能单元进行状态更新。包括车辆的位置更新，信道响应计算，事件的触发，消息的传播与转发等等。具体的流程详见图 2-1。



图 2-1 LTE-V2X仿真平台总体流程

### 反射模块以及IoC容器

反射是指程序可以访问、检测和修改它本身状态或行为的一种能力。反射一般用于获取类型的相关信息、动态调用方法、动态构造对象等等。

控制反转（IoC，Inverse of Control）并不是一种技术，而是一种设计模式，一种重要的面向对象编程的法则，它能指导我们设计出松耦合、更优秀的程序。它的核心思想是：将对象创建以及依赖注入的控制权从相关类中剥夺，并交由第三方容器来完成。

LTE-V2X系统级仿真平台用C++编程语言搭建，其中包含了数个模块，模块之间相互依赖，调用关系错综复杂。一个模块对象往往依赖其他多个模块，从而导致模块对象难以创建，类与类之间高度耦合，更难以测试。反射模块结合IoC容器的实现能够很好地解决这一问题。

### 地理拓扑与传播单元

地理拓扑与传播单元是LTE-V2X系统级仿真平台中最基础的功能单元，其功能包括车辆撒点，位置刷新，信道响应计算，衰落计算等等。根据36.885协议规定，仿真场景有曼哈顿街区模型和高速公路模型两种，每种类型依据相关协议选用相应的信道模型。

依据36.885协议中的定义，在地理拓扑与传播单元中，我设计了相应的子过程，包括场景建模、车辆撒点、车辆运动、传播模型与信道刷新4个子过程。各个子过程的调用关系详见图 2-2



图 2-2 地理拓扑与传播单元各子过程调用关系

### 无线资源管理单元

在车联网系统中，时频资源是非常稀有的共享资源。从某种程度上，资源管理算法的好坏将决定了整个车联网系统的性能，包括整体的吞吐量、丢包率、冲突率、时延等性能指标。因此无线资源管理单元在LTE-V2X车联网仿真平台中起到至关重要的作用。

无线资源管理算法要解决的问题是，尽可能地提高时频资源利用率，降低消息传输时延，减少冲突次数。在资源分配的过程当中，依据不同的情景，冲突又可以细分为以下两类：

1. **选择冲突**：即某车辆节点进行时频资源选择时，选取了邻近车辆已经占用的时频资源块。选择冲突必然会导致传输冲突。
2. **传输冲突**：即由于不同车辆选择了相同的时频资源块并进行消息的传输，此时产生了非常严重的干扰，极大地提高了消息的丢包率。

V2X消息可以细分为V2I消息、V2V消息以及V2P消息，在不同的应用场景中，采用的资源管理策略也不尽相同。但总体上，包含如下三种策略：

1. Collision avoidance based on sensing (P1)：基于载波侦听的选择冲突避让机制，车辆在自由选择资源块进行数据包传输前，首先会检测当前选择的资源块是否被占用，如果检测到该资源块已被占用，那么会执行选择冲突避让策略，即当前车辆不会选择该资源块，进而选择其他未被占用的资源块，从而达到避免选择冲突的效果。
2. Enhanced random resource selection (P2)：最基本的随机选择算法是状态无关的随机选择算法，其性能较差，如果我们能够充分利用车辆的状态（车辆自身的信息，以及车辆搜集到的周边信息）来加强随机选择算法，那么同样能够得到非常显著的增益。
3. Location-based resource allocation (P3)：虽然我们研究的是无中心自组织网络的资源分配算法，但同样我们也可以借助基站、RSU等基础设施提供的信息，来进行资源分配。最常见的方式就是根据地理位置，将车辆进行。

### 无线传输单元

无线传输单元是LTE-V2X系统级车联网仿真平台中工作于链路层的功能单元。由车辆产生的周期性或者非周期性事件在车辆与车辆之间或者车辆与RSU之间进行传输，无线传输单元的主要功能就是利用地理拓扑与传播单元实时更新的信道响应信息来计算通信链路的载干比，并以此判定传输的数据包是否丢失。注意到，车联网仿真平台并非真正意义上进行比特或者符号传输，而是利用信道响应信息计算而来的载干比，并将其映射为错误概率，从而进行丢包的判定。

### 路由单元

路由单元是LTE-V2X车联网仿真平台中位于网络层的一个功能单元。在Ad hoc这样多跳、无中心的自组织车联网络中。每个车辆节点同时扮演者不同的角色：既可以是信源，即消息的发送端；也可以是信宿，即消息的接收端；还可以充当一个中间节点，即路由节点，起到转发消息的作用，将来自上一节点的消息转发给下一个节点。

路由单元在LTE-V2X车联网仿真平台中起到至关重要的作用。首先，它屏蔽了底层场景的具体建模信息；其次，提供完备、可靠的消息转发协议；最后，为各种不同的路由算法提供统一的接口。

在车联网系统中的每个车辆都可以作为路由节点，接收来自上一个车辆节点的消息，并将其转发给下一个车辆节点。不同的消息转发协议将对消息传递的效率，消息传递的可靠性产生非常大的影响。本篇论文中将给出两种不同的消息转发协议，一种面向消息传递的可靠性，另一种面向消息传递的时效性。

### 业务模型与控制单元

车联网应用可以支持的消息业务大体可以分为两类，安全相关的和非安全相关的。安全相关的业务可以包括交通事故预防、交通事故处理和交通拥堵原因调查。非安全相关的业务可以包括所有与道路交通安全无关的个人数据业务，包括驾驶员和乘车人在车内可能需要的各类业务，例如导航、网页浏览、网络游戏和微博等娱乐业务。

其中安全相关的业务还可以进一步分为周期性的安全消息和事件触发的紧急安全消息。

1. 周期性安全消息：例如信标消息，发送模式主要是广播，通过周期性的广播车辆可以将自己当前的状态通知给周围的其他车辆，因为车辆的状态信息对于周围所有的邻居车辆都十分重要。这类消息主要包含车辆的速度、方向和位置等信息。产生这种数据包的频率为5-10HZ。数据包长度在字节200-400bytes。
2. 事件触发型的紧急安全消息：一般用于道路上突发的安全事故或者安全隐患的广播。一旦在道路上发现安全隐患，具有广播能力的车联网终端须要马上发送紧急安全消息通知附近车辆。事件触发的安全消息可能是通知车内异常情况，包括发送机、制动和动力系统故障，也可以通知路上的异常情况，例如碰撞警告、急刹车和低能见度等情况。这类数据包的大小略小于周期性安全消息，长度为100-300bytes。

## 资源分配算法的研究

V2X消息大致上可以分为V2I消息与V2V消息。针对V2I消息，无限资源分配算法包括：轮询算法、基于载波侦听的算法、基于地理位置的分簇算法等。对于V2V消息，无限资源管理算法包括：随机选择算法、冲突避免算法等等。

LTE-V2X车联网系统中的资源分配算法与传统的LTE系统中的调度算法有显著区别。LTE系统中的调度算法是集中式的调度算法，所有的移动设备首先要接入基站，然后由基站统一进行调度。而车联网系统是一个典型的无中心自组织网络，每个车辆节点若要占用时频资源，必须通过车辆节点自身收集到的信息或者由RSU等车联网辅助设备提供的辅助信息来进行无线时频资源的抢占与竞争。

### 基于V2I的资源分配算法

我们总共提出了3种方案：

* 轮询调度算法：该算法属于集中式资源分配算法，仅作为对比方案。
* 基于分簇的时分算法：该算法首先根据RSU覆盖范围进行分簇，每个RSU下的不同分簇之间进行时分复用。
* 基于分簇的非时分算法：该算法首先根据RSU覆盖范围进行分簇，不同分簇不进行时分复用，也就是频域复用的最小单元就是簇。

1. **基于分簇的轮询调度算法**

轮询调度算法（Clustering Based Round Robin, C-RR），轮询调度算法本质上属于集中式资源管理方案，此时，RSU不仅仅是一个静态基础设施节点，同时它还具有类似于基站的接入、调度等功能，时频资源的分配完全由RSU实现。

根据36.885协议规定，车联网占用的频段带宽大约在50个RB，我们将整个频域资源划分成若干个，每个包含一个或者多个RB。该轮询调度算法在不同的并行进行轮询调度，最大程度提高系统的吞吐率。

1. **基于分簇的时分算法**

基于分簇的时分算法（Clustering Based Time Division Multiplexing, C-TDM）是一种基于时分、频分的分布式资源管理方案。我们将RSU的一个时域调度单元，划分为多个时段，每个簇内的车辆只能在该簇所分配的时段内才能进行数据包的发送，以位于道路中央的RSU为例，该RSU的时域调度单元被分为了4个时段，不同分簇时段的长度非固定，而是根据该簇内车辆的数目进行动态调整，车辆数目越多，分配的时段就越长。另外，为了保证紧急事件的高时效性，与高优先级，我们单独划分出一个频段用于传输紧急事件，而该频段在时域上不进行划分，即只要紧急事件触发，在任何时段都可以进行接入，保证了紧急性事件具有较低的时延特性。



图 2-3 C-TDM算法的流程

1. **基于分簇的非时分算法**

C-TDM算法可以很好地保证消息传输的可靠性，很大程度上避免了冲突。但是由于簇间时分的引入，会导致消息的传输时延变得非常大，同时由于资源预留机制的引入，也会导致频域资源利用率下降，还需要耗费额外的车载功率。

基于分簇的非时分算法（Clustering Based Non Time Division Multiplexing, C-NTDM）不再进行时域资源的分割，同一RSU覆盖范围内的车辆可以在任意时刻进行接入。

此外，在频域资源的划分上，C-NTDM算法与C-TDM算法也有所不同，C-NTDM算法不再根据消息类型划分出不同的，而采用了单一大小的。如此一来，任意优先级的消息都可以选择任意进行传输。由于C-NTDM算法也采用类似于C-TDM算法的冲突避让机制，这里不再赘述，因此我们可以通过给不同优先级的消息设置不同的退避窗口来体现优先级，以达到尽可能地让优先级较高的消息快速抢占资源的效果。



图 2-4 C-NTDM算法的流程

### 基于V2V的资源分配算法

1. **随机选择算法**

首先，我们将整个频域资源划分成多个，一个包含一个或多个RB。随机选择算法非常简单，就是在可选的集合中，挑选一个用于广播本次消息。在车辆密度较小的场景中，随机选择算法可以取得非常好的效果，并且算法实现复杂度极低，效率非常高，但是在车辆密度较高的场景中，该算法的效果就会急剧下降，因为在同一时间段内，可能会有许多地理位置相近的车辆同时占用同一个进行传输，会产生严重的干扰。

1. **基于载波侦听的随机选择算法**

在随机选择算法中，一个车辆节点独立地随机选择一个可用，而完全没有考虑到该是否已被其他邻近车辆占用。因此，我们可以引入载波侦听机制。

在载波侦听机制中，车辆节点在进行选取的时候，首选会侦听所有上的载波功率强度，功率强度低于某个临界值时，即可认为该未被邻近车辆占用，因此可以生成一个空闲链表。然后我们在这个空闲链表中挑选一个载波功率最小的，用于本次消息的传输。这样一来，可以大大缓解V2V消息相互干扰的情况。

1. **基于地理位置的时分算法**

车辆之间的同频干扰随着距离增大而降低，示意图详见图 2-5(a)。在此示意图中，A、B、C、D四辆车占用同一个，以车辆A为中心进行分析，将干扰区域划分为三段，每段的边界分别是、、。表示与车辆A的距离，表示**冲突区域**，表示**强干扰区域**，表示**弱干扰区域**，表示信号可达的最远距离。车辆B位于冲突区域，在进行载波侦听功率检测时，车辆A在该上的功率大于临界值，于是根据基于载波侦听的随机选择算法，车辆B将不会选用该来传输消息，从而有效地避免冲突；车辆C位于强干扰区域，在进行载波侦听功率检测时，车辆A在该上的功率小于临界值，于是根据基于载波侦听的随机选择算法，车辆C有可能选用该来传输消息，详见示意图图 2-5(b)，此时，车辆A、C复用同一个，这将会导致位于车辆A、C之间的区域（即示意图中标注的区域）将产生较大的干扰，该区域内的车辆可能无法同时收到A、C发送的消息。车辆D位于弱干扰区域，车辆A、D复用同一个所造成的干扰微乎其微，因此可以忽略不计。

因此，如果我们将车辆A与车辆C进行时域划分，那么对于区域的车辆来说，其受到的干扰就会大大降低。



图 2-5 同频干扰示意图

# 关键技术

## 反射的实现

要实现反射，首先要解决的第一个问题是：如何在运行时获取类型信息。我给出了一种实现方式，提供一个基类object，要实现反射的所有类必须继承这个基类object。然后重写object的operator new和operator delete方法，这两个方法是从堆中分配和释放内存的方法。在分配内存时，额外分配4个字节（int类型）用于存放class id。

此时，基类object的内存构造与子类derive的内存构造详见图 3-1：



图 3-1 内存构造示意图1

其中，代表基类object的起始地址，代表class id的起始地址，代表子类derive的起始地址，其中derive1、derive2继承自object，且derive1带有虚方法。可以看出，由于子类derive1含有虚函数，与之间产生了额外的偏移量，这对我们提取class id造成了一定的困扰，为此，我们给基类额外增加一个虚方法，此时，基类object的内存构造与子类derive的内存构造详见图 3-2：



图 3-2 内存构造示意图2

此时，可以看出，无论子类derive是否含有虚方法（只要object是derive的第一基类），与之间不会引入额外的偏移量，于是我们拿到object的指针，往前移动4个字节就可以提取出事先存入的class id。于是我们给每个子类生成一个唯一的class id，然后重写子类的operator new与operator delete方法，这两个方法会调用基类object的相应方法来分配和释放这额外4个字节的内存，用于存取类型信息。这两部分形式都是统一的，可以用宏来进行实现。

实现反射要解决的第二个问题是：如何通过指定字符串创建对象，以及如何通过基类object的指针，并根据指定字符串调用相应的方法。该问题的实现，完全依赖于C++原生语法，因此这部分的实现具有一定的侵入性。首先，定义一个工厂方法以及数个反射调用入口方法，形如：

* object\* new\_instance(const string& class\_name);
* void invoke(object\* obj, const string& method\_name);
* void invoke(object\* obj, const string& method\_name, int param1);
* void invoke(object\* obj, const string& method\_name, object\* param1);

注意到，反射方法的调用形式是固定的，而非任意的，并且在每个反射调用入口方法中以硬编码的形式写下两部分内容：

1. 强制类型转换，将基类的指针转型成子类型的指针。
2. 根据字符串调用相应的方法。

* 这两部分形式都是统一的，因此可以用宏来进行实现。

## 分簇接入机制

为了实现分簇设计思路，车辆节点必须要能够接入到某一个RSU，并根据所在的区域范围决定当前车辆归属于哪一个分簇。具体的接入方案描述如下：

* 每个车辆节点将自己的车辆描述信息、车辆位置信息（通过GPS定位装置获取）、反馈频段通过某个特定频段进行广播。
* 每个RSU覆盖的区域明确规划，并根据RSU的种类（道路中央或十字路口）进行簇的划分。
* RSU收集车辆广播的位置信息，并确定是否位于当前RSU的范围，若不属于当前RSU覆盖范围，则不做任何处理；若属于当前RSU覆盖范围，则计算该车辆所属簇号，将RSU信息以及分簇信息通过接收信息中指定的反馈频段发送给该车辆。
* 如果车辆在指定时延内未收到RSU的反馈信息，则认为当前位置信息的广播发生了冲突，根据随机算法调整广播时刻。

## 冲突避让机制

此外，我们还采用了冲突避让机制，最大程度地减少车辆节点由于竞争资源而产生的冲突。控制过程包含四个处理内容：侦听、发送、检测、冲突处理：

* 侦听：UE在发送数据前先侦听一下是否有数据正在传送（线路是否忙）。若“忙”则进入后述的“退避”处理程序，进而进一步反复进行侦听工作。若“闲”，则进行数据发送；
* 发送：当确定要发送数据后，进行数据的发送；
* 数据发送后，也可能发生数据传输冲突。因此，要对数据边发送，边接收，以判断是否冲突了；
* 当确认发生冲突后，进入冲突处理程序。有两种冲突情况：
* 若在侦听中发现线路忙，则等待一个延时后再次侦听，若仍然忙，则继续延迟等待，一直到可以发送为止。每次延时的时间不一致，由退避算法确定延时值；
* 若发送过程中发现数据冲突，先发送阻塞信息，强化冲突，再进行侦听工作，以待下次重新发送；

# 论文计划

2017年1月：确定论文主题方向，进行论文题目的筛选。

2017年1月：以论文题目为核心，对相关资料进行收集和翻阅。

2017年1月：对已搜集的资料加以整理，论证分析论文的可行性、实际性，将论文题目和大致范围确定下来，进行开题报告。

2017年2月-6月：整合已有资料、构筑论文的大纲。

2017年7月-10月：根据查找的数据和相关资料，进行深入详实的论文编写工作，对论文编写过程中所发现的问题，研究其解决方案，推敲整合，并进行修改完善，准备论文中期检查。

2017年11月：完成论文的初稿部分，向指导老师寻求意见，优化论文的结构，润色语句，修改不当之处，补充不足之处。

2017年12月，论文资料整合，最终定稿，为最终的答辩做好各方面准备，熟悉论文内容，增强自己对论文内容的把握，进行一定的思维发散，设计论文答辩。

2018年3月：论文答辩。

# 论文进度及目标

目前正在进行仿真结果分析，争取在2017年11月15日之前完成初稿。