

调研情况

TDMA-VANET 要解决的问题是两个：join collision 和 merge collision。在 VANET 中，由于车辆运动快，merge collision 的影响会比较大。根据最新的文献，主流的做法是考虑车辆的运行方向，将一帧分为左右两部分；不同方向的车辆会到不同的部分去申请时隙以避免 merge collision。但这个做法一个不好的地方就是不能很好的适应车流不均衡的情况（不同的方向车辆密度差距大），为了解决这个问题，一些工作试图通过引入更多的判断信息来动态规划不同方向车辆可以申请的时隙数量。在这些工作中，时隙分配的基础依然是 RR-ALOHA。在其中，每个节点在一帧开头时广播当前帧的时隙占用情况（FI 报文）；每个节点收到 FI 后就能够直接知晓两跳范围内所有节点的时隙占用情况。每个节点需要保证的是自己所选择的时隙在两跳范围内的节点中是独占的。

刘瑞霖的方案简述

刘瑞霖的时隙分配方案的思路和上述不同，在他的设计中每个节点在广播 FI 的基础上广播在每个时隙上，两跳邻居的占用的数量（这个信息是由原 FI 报文直接累加出来的）。根据增强版的 FI 报文，节点可以将收到的“两跳节点占用情况 $\text{Count}_{2\text{hop}}$ ”（每个时隙占用 8 位）进行累加，计算出每个时隙的三跳邻居占用数量 $\text{Count}_{3\text{hop}}$ 。算法根据 $\text{Count}_{3\text{hop}}$ 的值来对每一个时隙进行评价， $\text{Count}_{3\text{hop}}$ 的值越低代表该时隙被隐藏节点所占用的可能性更小。由于存在重复计算， $\text{Count}_{3\text{hop}}$ 的值会比实际情况要偏大，但由于算法只进行相对值的对比，所以偏大的值实际上不会对时隙评价产生影响。有了对每个时隙的评价后，节点会选择评价最优（根据 $\text{Count}_{3\text{hop}}$ ）的两跳邻居未占用（根据原始 FI）时隙作为自己的基础时隙（BCH）。在选择好基础时隙后，节点可以根据时隙评价的变化情况实时对自己的 BCH 做出调整，以此来最大程度避免时隙冲突。

加入的一些新设计

行驶方向的引入

刘瑞霖方案中没有使用到车辆的方向信息，我认为这是不合理的。存在如下的情况：假设有时隙 A 和时隙 B，他们的 $\text{Count}_{3\text{hop}}$ 值一致；有两个方向的车辆。假设实际占用时隙 A 的节点刚好都是由东向西，而占用时隙 B 的节点都是由西向东。则对于一辆由东向西的车辆来说，申请时隙 A 明显要比申请时隙 B 要好（因为方向一致，所以时隙的占用情况更加稳定，时隙碰撞的可能性也会降低）。这个问题是刘瑞霖方案中没有考虑的。

我的基本思路是将车辆方向信息加入两跳节点占用情况 $\text{Count}_{2\text{hop}}$ 中，节点可以分别计算一个时隙在两个方向的车辆的 $\text{Count}_{3\text{hop}}$ ；并给一个时隙两个评价分别针对两个方向。具体的做法是，将原 $\text{Count}_{2\text{hop}}$ 的 8 位拆成 4+4 位分别代表两个方向。要注意的是 4 位最大表示的数是 15，由于这个值是每个时隙都有的，我认为 15 应该是足够的。

算法开销的降低

由于在 FI 的基础上额外引入了 8 位的 $\text{Count}_{2\text{hop}}$ ，这意味着算法开销的进一步扩大。当一帧为 100ms，包含 100 个时隙时，每个 FI 包的开销在 RR-ALOHA 的基础上增加 $8 \times 100 / 8$

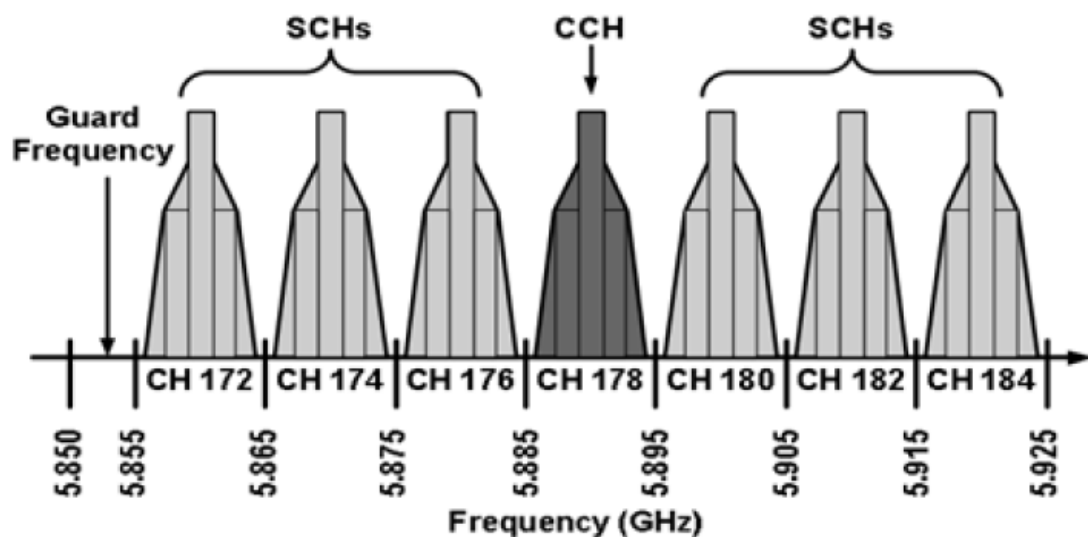
= 100 字节。则对于 1ms 一个的时隙来说，总开销约为 $23 * 100 / 8 = 287.5$ 字节，开销额外增加 53 %。在之前的测试中，以 12M 的速度发 287 字节时间需要约 300us，则总开销占带宽的 30%以上。这是一个很可观的数字。

我的改进思路是将一个帧进一步分为两个小帧，每个节点只能选择其中一个小帧内的时隙进行申请；并且只需要广播当前小帧内所有时隙的占用情况。这样的设计会使算法开销减半。目前还没有想到会不会产生其他的负面效果。

多信道以及服务信道的接入

基本设计

整个车载网信道分为 7 个频点，包括 1 个 CCH(控制信道)和 6 个 SCH(服务信道)。



CCH 中时隙与帧长的设计：1 秒分为 1024 个时隙；每个时隙时长为 976.5625 us。1 秒分为 16 个帧，每个帧分为 64 个时隙（62.5 ms）。每个节点必须且只能在 CCH 上申请一个时隙。CCH 负责发送 BCH 控制所需的 FI 包、SCH 接入的通知及确认报文、高优先级的安全报文。

SCH：采用帧长可变的设计，专门发送应用数据。应用数据分为“大量数据”和“实时数据（双工）”：大量数据的时隙申请策略为长帧、多时隙（连续），特点是延迟稍高、带宽大；实时数据的申请策略为短帧、单时隙，特点是低延迟，带宽小。

采用双接口设计：一个接口专门调至 CCH 频段；另一个接口专门用于 SCH 的收发（可以在多个 SCHs 上进行切换）。初期的实验可以用 433M+5.9G 组成双接口，用于演示少量节点时的情况；后期可以重新设计板子，做成两个 5.9G 模块的接口（原来的单 MiniPCle 板子去掉几乎没用的 RJ45 接口和 USB 接口后应该可以很方便的放下另一个 MiniPCIE 插槽）。

CCH 的时隙接入由上面 BCH 分配策略进行控制，下面专门讨论 SCH 的接入。

SCH 的接入基本策略如下：

1. 由软件给出一个报文的属性（应用 ID，实时还是大量）；
2. 由硬件判断当前服务信道的占用情况，分析出最优的时隙和帧的分配情况；
3. FPGA 在 CCH 上进行通知（时隙和帧分配情况、服务信道号、下一跳节点编号），并直接占用时隙（无需再走申请-通过的流程）。

4. 每次的分配通知有效期为 CCH 的一帧长度。即每个节点在 CCH 帧中的时隙上通知接下来一个 CCH 帧时长中，其服务信道时隙占用情况和下一跳节点。

SCH 中根据占用情况和需求分配时隙

CCH 有 64 个时隙一个帧，最多可以容纳 64 个节点；每个节点最多只能在同一时间段申请一个 SCH 上的时隙（只有一个 SCH 接口）。即每个 SCH 信道最多会容纳 $64/6$ 约 10 个点。

首先，服务信道的任务是提供点到点的帧传递（有或没有 Ack）。可以区别应对高数据量、延迟高和低数据量、延迟低的应用需求。

基本的 SCH 接入方式是直接占用、贪婪方式、根据新接入的节点实时调整时隙的占用。每个节点每次都直接占用“一个 SCH 帧”的长度，并且 SCH 帧长可以随着节点增加进行二分调整。一个 SCH 帧长最短为 2 个时隙，即每个节点分配到的时隙数量是 2 的倍数，我称成对的连续时隙为时隙组（由 A, B 两个时隙组成）；由此可以区分出两种数据方式的时隙占用：对于低延迟的应用，每个节点只占用时隙组的 A 时隙，留空 B 时隙给对向节点；对于大数据量的应用则每次都占满时隙组的 A, B 时隙，只在每一帧最后留一个时隙给对向节点用于 ACK 的传输（可选）。

以下是时隙分配示例（图片）：

其中存在的问题：

1. 怎样保证在有帧要发送时尽快发出去，并且对方还能够尽快收到（因为对方的服务接口不一定在这个频点上）。解决方法是可以从 CCH 中实时监听到节点 next-hop 目前所在的频点。有发送至 n 的帧倾向于直接使用 n 当前所在的频点；或是在 CCH 中通知 n 切换频点。这里又涉及到 n 节点是否在向其他节点传输数据的问题。
2. 潜在的碰撞问题：由于 CCH 的申请应该是避免了碰撞了，而 SCH 时隙的申请又是每一个 CCH 帧单独进行控制，所以 SCH 上应该不会出现碰撞的问题。

