

面向车载自组织网络路由的轨迹预测算法

黎 阳 王 哲 张楚文 戴惠辰 徐文全 姬雪枫 万 颖 刘 斌

(清华大学计算机科学与技术系 北京 100084)

(liyang-12@mails.tsinghua.edu.cn)

Trajectory Prediction Algorithm in VANET Routing

Li Yang, Wang Zhe, Zhang Chuwen, Dai Huichen, Xu Wenquan, Ji Xuefeng, Wan Ying, and Liu Bin

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract In vehicular ad hoc network (VANET), geographic routing protocols can preferably adapt to frequent topology changes and unstable link quality. Beacon messages are needed to share the positions of neighboring nodes, so forwarding decisions in the interval of successive beacon messages may be inaccurate due to the movement of the vehicle nodes. In this situation, trajectory prediction is needed to amend the positions of the vehicle nodes. Existing prediction algorithms are either lack of universality or suffered from large prediction errors. To solve the problems above, this paper proposes a new trajectory prediction algorithm, which is based on the measurement result that the vehicle accelerations obey normal distribution. The new algorithm uses linear regression to do the prediction and applies a feedback mechanism to amend error. The new trajectory prediction algorithm can greatly improve the prediction accuracy in several real trajectory trace tests. Then this paper proposes a new position based instant routing protocol. In instant routing protocol, a forwarder uses the predicted position of neighboring nodes and destination node to calculate the next hop. We apply our new trajectory prediction algorithm in instant routing to predict and update vehicle positions in real time. We use SUMO to generate real maps and vehicle trajectory traces, and use NS3 to do the simulation. Experimental results show that instant routing with the new trajectory prediction algorithm outperforms the traditional GPSR protocol and instant routing without trajectory prediction in terms of packet delivery ratio and network latency, while reducing protocol processing overhead remarkably.

Key words vehicular ad hoc network (VANET); instant routing (IR); trajectory prediction; geographic routing protocol; named data networking (NDN)

摘 要 在车载自组织网络(vehicular ad hoc network, VANET)(也称车联网)中,基于地理位置的路由协议能够较好地适应网络拓扑的动态性变化和链路质量的不稳定性.由于位置信息需要在邻居节点间采用信标分组进行交互,信标分组间隔内的转发决策可能因车辆节点位置的移动而不准确,需要进行位置预测来修正车辆节点的位置.已有的位置预测算法存在普适性差或预测误差大的问题.针对上述问题,提出了一种新的预测算法,首次通过测量得到车辆加速度服从正态分布的结论,利用线性回归进行预测,并采用反馈机制进行结果修正.利用真实车辆轨迹进行测试,新的预测算法的预测精度大为提高.

收稿日期:2017-05-31;修回日期:2017-08-08

基金项目:华为公司创新研究计划项目;国家自然科学基金项目(61602271,61373143,61432009);中国博士后科学基金项目(2016M591182)

This work was supported by the Huawei Innovation Research Program (HIRP), the National Natural Science Foundation of China (61602271, 61373143, 61432009), and the China Postdoctoral Science Foundation (2016M591182).

通信作者:刘斌(liub@mail.tsinghua.edu.cn)

然后,提出了一种新的基于位置的即时路由协议.在该协议中,发送节点利用邻居节点位置和目的节点位置计算出转发下一跳.将新的位置预测算法加入到即时路由协议中,实时预测和更新车辆的位置.利用 SUMO 软件生成了基于真实地图道路轨迹的车辆运动模型,结合 NS3 网络仿真平台进行了仿真实验.实验结果表明:采用新的预测算法后,相比传统的 GPSR 协议和不带预测的即时路由协议,新方法的收包率提高、延迟下降,并且协议开销显著降低.

关键词 车载自组织网络;即时路由;轨迹预测;地理位置路由协议;命名数据网络

中图法分类号 TP393.03

随着智能汽车系统和自动驾驶技术的兴起,车载自组织网络(vehicular ad hoc network, VANET)(也称车联网)其 V2X(vehicle to everything)技术迎来了巨大的发展机遇.而在基础设施(LTE-V-Cell 或 802.11p 路边单元)无法接入的情况下,VANET 是一种重要的通信方式^[1-3].但是,由于网络拓扑的动态性变化和链接质量的不稳定性,车载自组织网络在维持端到端通信过程中面临极大的挑战.由于全球定位系统正成为车辆的一个必备装置,实现基于位置的车载自组织网络路由协议成为可能^[4-6].基于位置的路由协议需要通过邻居和目的节点的位置信息来进行路由决策.而获取邻居位置需要邻居节点间定时交互信标分组(beacon packet).为了降低开销和提高路由协议的准确性,如何选取信标分组的发送间隔成为一个重要问题.

在基于位置的 VANET 路由算法中,如经典的路由算法 GPSR^[4],在源节点发出一个分组之后,每经过一跳都要根据当前节点掌握的邻居节点位置和目的节点位置选取下一跳转发节点.由于信标分组有一定的发送间隔(如 1s),当前路由决策节点掌握的其他节点位置信息并不是最新的.但是在某个数据分组的传输时间内,在一般的城市道路模型中,车辆的拓扑变化相对较为缓慢.基于这个基本观察,之前的研究提出了基于预测进行路由的思想,即在信标分组间隔之内,使用预测的位置信息^[7-10]作为节点路由选择的依据.

但是,之前基于位置预测的研究文章^[7-10]对于车辆行驶状况的抽象较为理想化,并且很多算法需要借助电子地图进行辅助判断,普适性较差.本文提出了一种新的车辆位置预测算法,对车辆行驶轨迹进行了重新建模,发现车辆加速度服从正态分布的特性,降低了预测开销,并利用真实车辆运行数据进行测量实验,得到了新型的路由协议中的位置预测算法.为了验证新的位置预测算法的有效性,设计了一种新型的即时路由协议(instant routing, IR)进行验证.

本文的主要贡献有 3 个方面:

- 1) 实验发现,车辆行驶过程中加速度更加服从正态分布的特征,提出了一种新的车辆位置预测算法,从理论上分析了预测算法的正确性;
- 2) 利用真实车辆行驶轨迹数据验证了新的车辆位置预测算法的有效性;
- 3) 设计了一种新型的基于位置信息的即时路由协议 IR,并且利用 NS3 和 SUMO 工具验证了新的车辆位置预测算法对路由协议性能提升的效果.

1 相关工作

目前,研究者提出了两大类基于位置预测的算法:1)对 2 个节点的可通信时长进行预测,经典算法有文献^[7-9];2)直接对节点的行驶轨迹进行预测^[10].

据我们所知,Menouar 等人^[7]最早提出基于位置预测的算法.作者将 2 个车辆节点的可通信时间记为 $Lifetime_{link}$,利用节点速度 v_i, v_j 和节点间的距离 $d_{i,j}$ 进行预测.另外,作者考虑到车道之间的距离可能相当大,因此也需要考虑车道之间的距离 ω .若设无线通信半径为 R ,则有预测模型:

$$Lifetime_{link} = \frac{\sqrt{R^2 - \omega^2} + s\sqrt{d_{i,j}^2 - \omega^2}}{v_i + v_j}, \quad (1)$$

其中, s 的取值与 2 个车辆节点的相对行驶方向有关.若两辆车相向行驶,则 $s=1$,否则 $s=-1$.上述模型是一个确定性的预测模型,其优点在于节点间交互信息较少,只需要知道节点间的速度信息,即可进行预测.但是,由于这个模型没有引入任何的先验假设,因此在面对不同的路况和不同的车节点拓扑时,其适应性不够好. Wang Xiufeng 等人^[8]提出可以假设道路上行驶车辆的速度为随机变量,且服从一般的正态分布 $N(\mu, \sigma^2)$.作者通过推导,得出了两辆车通信时长的预测模型:

$$E(T) = \int_{L/(\mu-4\sigma)}^{L/(\mu+4\sigma)} \frac{1}{t^2 \sqrt{2\sigma\pi}} \exp\left(-\frac{(L/t-\mu)^2}{2\sigma^2}\right) dt. \quad (2)$$

上述预测模型只利用了车辆的速度信息,需要交互的信息较少,计算方法也较为简洁.但是,这个预测模型是一个一维的预测模型,即只考虑两辆车的速度方向在一条直线上,不具备普适性.与文献[8]类似,Eiza 等人^[9]同样基于车辆速度服从正态分布的假设,但是利用了基于网络拓扑的演化图(evolution graph)给出了详细的算法.文章估计了2个车节点的通信时长及其置信度,由此构建一个图,节点为车辆,边权为车辆节点的可通信时长及置信度,利用EG-Dijkstra 算法选择路由.

Xue Guangtao 等人^[10]利用变阶 Markov 模型对车辆的行驶模式进行识别,通过借鉴自然语言处理中的机器学习方法,对大量车辆运行数据进行建模,为较大时间跨度的预测提供了可能性.但是,上述方法并不适用于 VANET 路由选择.原因是:1) 车辆运行的轨迹模式十分复杂,是相当于定制化的数据模式,普适性较差;2) 如果使用上述轨迹预测方法进行路由,需要结合地图信息,然而在许多车载自组织网络应用场景中,并没有精确的电子道路地图.

综上,这些算法各有侧重,但是都无法较好地应用于基于位置的路由协议计算.为此,我们将在第2节中提出一种新的预测算法.

2 基于加速度的预测算法

在本节中,我们将提出一种新的基于加速度的预测算法.本节中的实验数据均来自真实的道路车辆运行数据.首先我们将在2.1节提出一种基于加速度正态分布假设的朴素预测方法;针对朴素预测算法对于拐弯预测较差的情况,我们将在2.2节中提出一种改进的预测算法.

2.1 朴素预测算法

我们对基于速度正态假设的预测算法^[8-9]进行了分析.从本质上讲,Wang Xiufeng 等人^[8]提出的预测算法是在匀速运动的假设上进行的,即假设在同一时刻道路上各种车辆的速度分布为正态分布,但在一段时间内各个车辆保持匀速.上述假设导致了预测模型的实时性较差.我们从新的角度思考,不再考虑所有路上车辆的速度分布,转而考虑单个车辆的速度作为随机过程的统计规律.我们通过实车实验得到了一辆汽车在城市道路运行的轨迹,并统

计了一段时间内的运行速度,如图1所示.从图1我们可以发现速度的正态分布规律不够明显,因此我们进一步分析了加速度的分布.我们利用车辆轨迹,取时间间隔为1s,计算加速度,得到了加速度分布直方图,如图2所示.从图2可以看出,加速度拥有较好的正态性.如果设每一时刻的加速度为一随机变量 a ,则应该有 a 服从 $N(\mu, \sigma^2)$.

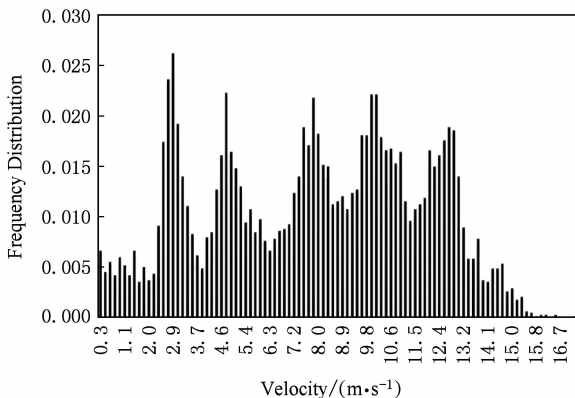


Fig. 1 Velocity-frequency statistics

图1 速度-频率统计图

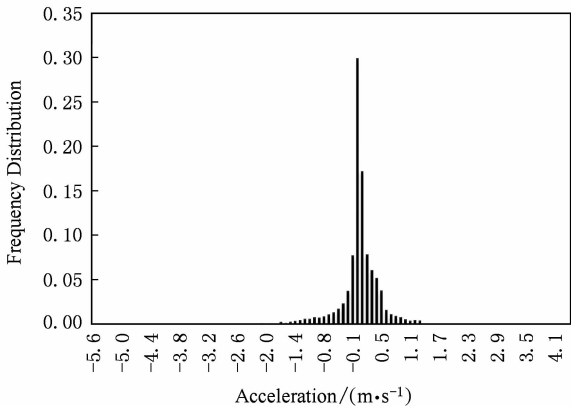


Fig. 2 Acceleration-frequency statistics

图2 加速度-频率统计图

基于上述假设,我们可以给出一种基础预测算法.随机变量 a 服从正态分布 $N(\mu, \sigma^2)$,其密度函数满足:

$$f_a(x) = \frac{1}{\sqrt{2\sigma\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (3)$$

有了上述概率密度函数,一个重要问题是对分布参数进行估计.由于我们已经拥有了较大量真实车辆运行轨迹数据,因此基于统计的参数估计是首要选择.假设已经有 n 个加速度数据 $a_i, i=1, 2, \dots, n$,则:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n a_i, \quad (4)$$

$$s^2 = \frac{n}{n-1} \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (a_i - \hat{\mu})^2. \quad (5)$$

注意,此处使用 s^2 对其方差进行估计,而并非使用 $\hat{\sigma}^2$. 由此,假设在一小段时间 Δt 内车辆做匀加速运动,初速度为 v_0 ,则运动距离为

$$\Delta x = v_0 \Delta t + \frac{1}{2} a \Delta t^2,$$
$$a \sim N(\mu, \sigma^2).$$

(6)

为了使距离预测尽可能准确,我们设总的预测时间为 T ,并给出区间 $[0, T]$ 上的一个分割: $0 = t_0 < t_1 < t_2 < \dots < t_n = T$,记预测结果为

$$\Delta x_i = v_{0i}(t_i - t_{i-1}) + \frac{1}{2} a(t_i - t_{i-1})^2,$$

(7)

其中, $i = 1, 2, \dots, n$. 则最终的预测结果为

$$x = \sum_{i=1}^n \Delta x_i.$$

(8)

在实际情况下,方便的方法是取所有分割长度相等,记为 Δt . 所以,可得到一维车辆节点位置预测算法 1.

算法 1. 一维车辆节点位置预测算法.

输入:当前车辆速度 v 、加速度参数 μ 和 σ 、积分步长 Δt 、预测时间 T ;

输出: T 时间后的车辆位移 x .

- ① $time \leftarrow 0$;
- ② $x \leftarrow 0$;
- ③ $temp \leftarrow v$;
- ④ For $time \leq T$ do
- ⑤ $a \leftarrow normal_variable_producer(\mu, \sigma)$;
- ⑥ $x \leftarrow x + temp \times \Delta t + a \Delta t^2 / 2$;
- ⑦ $temp \leftarrow temp + a \Delta t$;
- ⑧ $time \leftarrow time + \Delta t$;
- ⑨ End For

针对上述算法,利用一段直路上的真实行车轨迹对这个算法进行了检验,其中取 $\Delta t = 0.1$ s,得到结果如图 3 所示. 图 3 中,横轴表示取不同的预测时长 T ,纵轴表示在该条件下进行多次预测的误差的平均值. 从图 3 可以看出,即使在预测时长较长时,

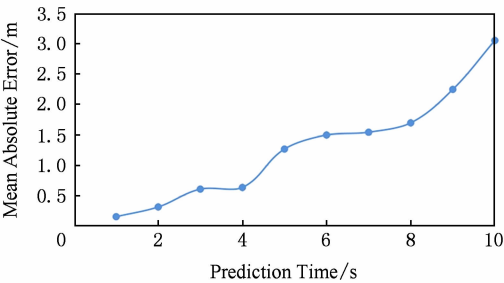


Fig. 3 Absolute error of simple prediction
图 3 朴素预测的绝对误差

其效果仍然较好,说明该模型是正确可用的. 但值得指出的是,当预测时长 T 较大时,预测结果将失去意义,因为真实车辆并不能在一条直线道路上长时间行驶而不拐弯或掉头.

另外,原则上讲 Δt 的取值应该越小越好,但是,在实际操作中, Δt 只要不过大,对预测结果的影响就很小. 为了保证积分的正确性, $T/\Delta t$ 必须是整数. 需要进一步说明的是,在实际应用场景下对加速度的正态分布的参数估计可以采用多种方法. 其中一种方法是:在预测开始时,用一个按 2.1 节所述方法估计的先验参数,之后再随着时间的推移,不断修正参数. 本文主要讨论最基本的预测算法原理,所以将不对上述工程细节展开探讨.

在此基础上,加上对速度方向的预测,即可对车辆二维坐标位置进行预测. 基于对车辆位置的预测,可以很容易预测接下来需要的参数,如通信时长等. 在预测速度方向时,一个朴素的想法是将开始预测时的速度方向作为接下来预测的速度方向. 通过上述思想,利用一段真实车辆行驶轨迹,得到了朴素算法的实验结果如图 4 所示. 其中,圆圈表示实际的运行轨迹,十字表示预测的运行轨迹,预测时长 $T = 10$ s. 预测的绝对误差分布如图 5 所示,表示的是每次预测误差的分布图. 横轴表示预测误差,纵轴表示出现这种误差的频率. 通过图 5 可以得到在这种朴素的预测方法下,误差大多数都集中在 10 m 以下.

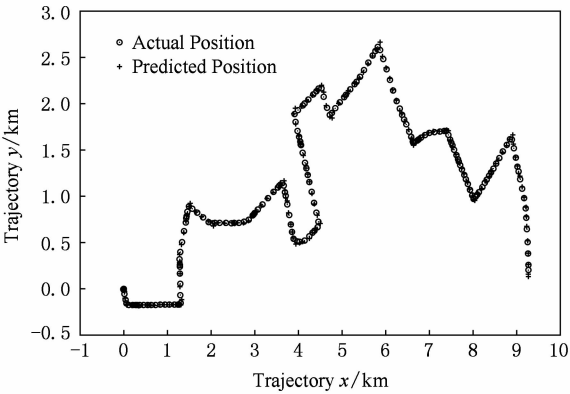


Fig. 4 Trajectory prediction result of simple prediction
图 4 朴素预测算法的轨迹预测结果

在图 4 中,大多数的大误差点都集中在拐弯处,如图 6 所示. 事实上,由于我们对速度方向的预测是基于预测时的速度方向,导致预测实质上是在一维方向上的预测. 因此,如果在预测时,车辆恰好在拐弯的过程中,则拐弯的偏差就会较大,极端情况下可能有 50 m 的误差.

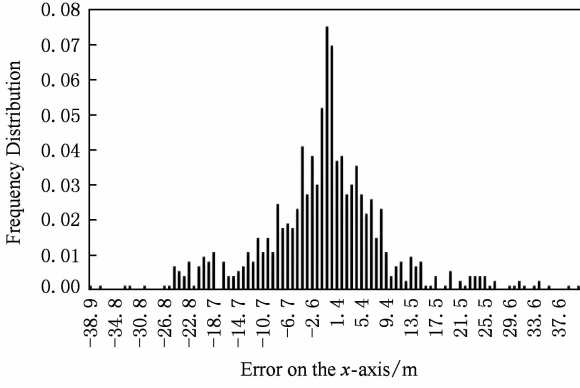


Fig. 5 Error distribution of simple prediction

图 5 朴素预测算法的误差分布

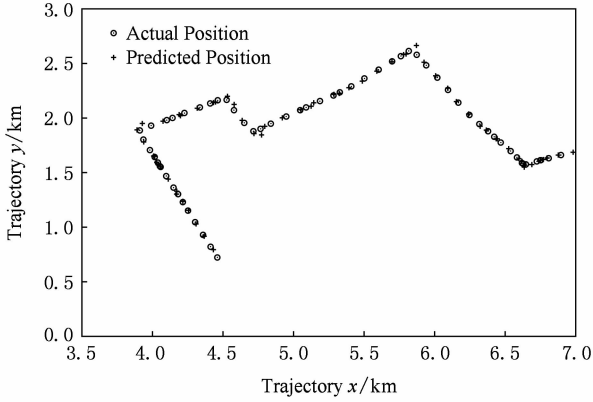


Fig. 6 Trajectory prediction result of simple prediction: local map

图 6 朴素预测算法的轨迹预测结果:局部图

2.2 改进的车辆位置预测算法

为了避免在拐弯预测中产生较大的偏差,一个自然的想法是通过预知拐弯的发生来改善预测结果.对于一段的道路情况,拐弯是一个渐变的过程,拐弯时长通常在 $8 \sim 10$ s. 以一个常见的 90° 拐弯为例,拐弯过程如图 7 所示.这是一张车辆运行的轨迹

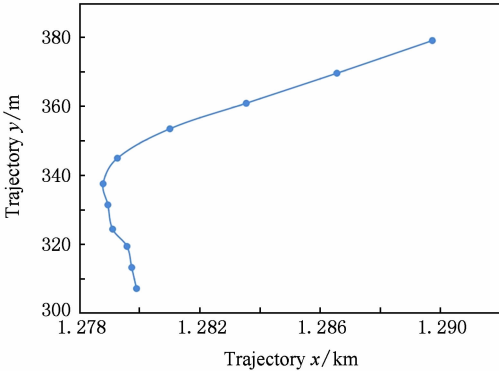


Fig. 7 Examples of right-angle turning

图 7 直角拐弯示例

图,绘制间隔为 1 s,即每一个点表示一辆车在某一时刻的位置,连续的点之间的间隔为 1 s. 通过图 7 可以想到,如果朴素预测算法的预测发生在恰好要拐弯的第 4 秒或之后几秒,此时的预测误差将会非常大.因此,改进这种误差的关键是能够预测到拐弯的发生.

一个直观的想法是,假设拐弯是一个相对均匀的过程.实际上,这个假设符合实际情况,因为在一般道路上拐弯时的速度要小于直线驾驶时的速度.如果一个普通的 90° 拐弯要持续 10 s,则每一秒的转角平均下来为 9° . 如果我们发现了一辆车在第 3, 4 秒的时间内速度方向有连续的改变,那么我们便可以预知一个拐弯.将这种思想进行推广,如果我们每一次预测时的速度方向都是按照前几秒的速度方向来估计,其效果应该远好于之前的朴素预测.最简单有效地利用上述思想的方法就是线性轨迹.在实际中,用车辆运动轨迹的切线方向表示速度方向.而在离散轨迹中,用折线段来近似车辆轨迹,则可用折线段斜率表示速度方向.注意,为了避免出现斜率过大的情形,应适当地在局部进行坐标系的旋转,以防止因为斜率过大导致的计算预测不准确.

现假设已观测到 n 组数据,每一组数据包括连续 m 秒的速度方向数据 $(k_{i1}, k_{i2}, \dots, k_{im})$,并知道接下来的速度方向为 $k_i (i=1, 2, \dots, n)$,则可建立线性回归模型:

$$\begin{cases} k_i = \beta_0 + \beta_1 k_{i1} + \dots + \beta_m k_{im} + \varepsilon_i, \\ \varepsilon_i \sim N(0, \sigma^2), \end{cases} \quad (9)$$

其中, $i=1, 2, \dots, n$. 将式(9)表示成矩阵式,则记:

$$\mathbf{X} = \begin{bmatrix} 1 & k_{11} & \dots & k_{1m} \\ \vdots & \vdots & & \vdots \\ 1 & k_{n1} & \dots & k_{nm} \end{bmatrix}, \quad (10)$$

$$\mathbf{Y} = \begin{bmatrix} k_1 \\ \vdots \\ k_n \end{bmatrix},$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_m \end{bmatrix}. \quad (11)$$

则这个模型可以表示为

$$\begin{cases} \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \\ \boldsymbol{\varepsilon} \sim N(0, \sigma^2 \mathbf{I}). \end{cases} \quad (12)$$

按照最小二乘法估计参数 $\boldsymbol{\beta}$, 应最小化误差平方和,即考虑代价函数:

$$Q(\boldsymbol{\beta}) = \sum_{i=1}^n \varepsilon_i^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \quad (13)$$

求其极值,只需考虑条件 $\frac{\partial Q}{\partial \beta_j}=0$,则应该有参数 $\boldsymbol{\beta}$ 的最小二乘估计:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (14)$$

由此可得到速度方向的线性预测模型:

$$\hat{k} = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i k_i + \boldsymbol{\varepsilon}. \quad (15)$$

在统计学上,为了评价模型的精确性,要引入相关系数来刻画.为此,记:

$$U = \sum_{i=1}^n (\hat{k}_i - \bar{k})^2,$$
$$S = \sum_{i=1}^n (k_i - \bar{k})^2, \quad (16)$$

更一般地,我们定义相关系数

$$R^2 = \frac{U}{S}. \quad (17)$$

容易说明 $0 < R^2 < 1$,且相关系数越大表明模型的线性越好.尽管不存在理论上的一个阈值,使得相关系数大于这个阈值时认为模型的精确度较高,但在实际工程应用中,一般认为当 $R^2 > 0.5$ 时,模型就已经充分可用.为使用该模型对车辆行驶方向进行预测,首先应该确定回归变元数量 m 的取值.相关系数随变元数量变化的结果如图 8 所示.图 8 的横轴表示选取了前 $n(n=1,2,\cdots,13)$ 秒的速度方向来线性拟合这一时刻的速度方向,纵轴表示线性回归系数.从回归之后的结果我们知道这种线性是统计显著的,回归系数在 0.6 左右,表示模型精确度尚可.在图 8 中,我们发现,取 $n=7$ 是比较合适的;若 $n > 7$,则之后的变量影响极小,不必考虑.

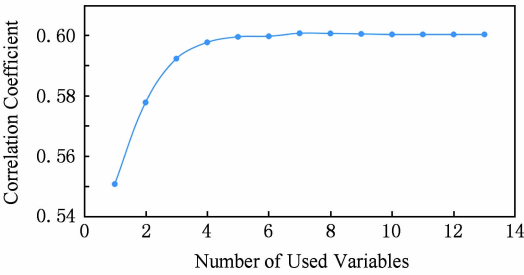


Fig. 8 Correlation coefficients with different number of variables

图 8 拟合变量个数对线性回归系数的影响

基于以上结果,我们将预测的过程细分,预测逐秒进行,每次都通过回归系数计算出一个新的速度方向,可以得到新算法如算法 2 所示:

算法 2. 改进的车辆位置预测算法.

输入:当前车辆速度 v 、加速度参数 μ 和 σ 、积分步长 Δt 、预测时间 T 、前 7 s 的斜率向量 $\mathbf{k}$;
输出: T 时间后的车辆位置 $\mathbf{x}$.

- ① $time \leftarrow 0$;
- ② $\mathbf{x} = \mathbf{0}$;
- ③ For $time \leq T$ do
- ④ $temp_k = linear_predict(\mathbf{k})$;
- ⑤ $\mathbf{x} \leftarrow \mathbf{x} + position_predict(v, \mu, \sigma, \Delta t, 1, temp_k)$;
- ⑥ construct a new $\mathbf{k}$ according to $temp_k$;
- ⑦ $time \leftarrow time + 1$;
- ⑧ End For

其中, $position_predict(v, \mu, \sigma, \Delta t, 1, temp_k)$ 表示利用之前的朴素预测算法预测位置,预测时长 $T=1\text{ s}$,距离为按照朴素预测算法得到的结果 $\mathbf{x}$,方向是 $temp_k$.按照这个算法得到的预测结果如图 9 所示,其局部放大图如图 10 所示.图 10 中的点表示的意思与之前的轨迹图类似.在这里,我们没有考虑线性预测过程的误差 $\boldsymbol{\varepsilon}$,先令 $\boldsymbol{\varepsilon} = \mathbf{0}$.

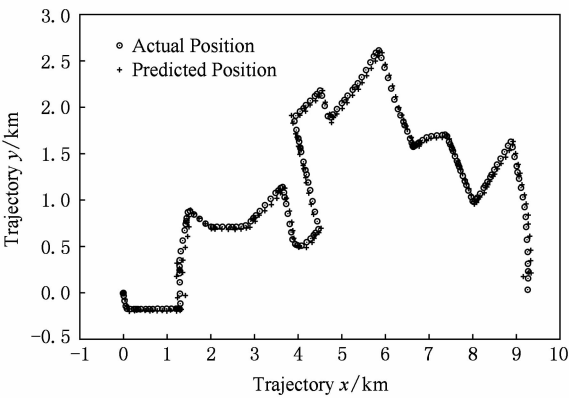


Fig. 9 Trajectory prediction result of improved prediction

图 9 改进预测算法的轨迹预测结果

从图 10 中不难看到,对拐弯的预测已经改善很多.这里要说明的是,这种预测算法的最大误差会发生在预测时车辆刚刚要进入拐弯,即图 7 中提到的拐弯的第 1 秒.但这时的最大误差显然会远小于之前朴素预测算法的最大误差.为说明这一点,我们可以简单计算如下:设一个拐弯总长度为 40 m,拐弯前后各行驶 20 m,车辆匀速行驶,拐弯总时间为 10 s,预测时间 $T=10\text{ s}$.我们假设一维预测是准确的,即忽略一维预测带来的误差,则朴素的位置预测算法的误差发生在恰好拐弯处:

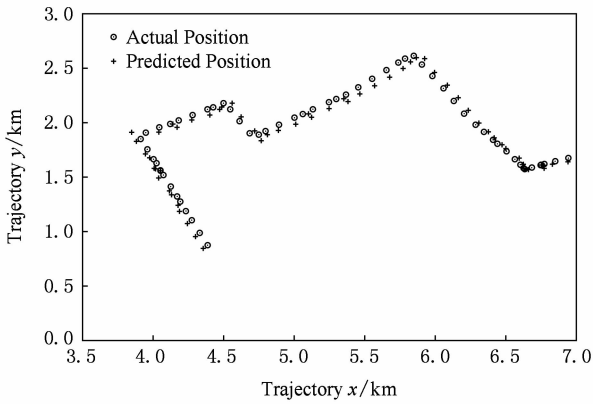


Fig. 10 Trajectory prediction result of improved prediction; local map

图 10 改进预测算法的轨迹预测结果:局部图

$\epsilon_1 = 40\sqrt{2} \text{ m} = 56.57 \text{ m}.$

改进的预测算法的最大误差发生在刚拐弯时:

$\epsilon_2 = 20\sqrt{2} \text{ m} = 28.29 \text{ m}.$

因此,改进算法的误差已得到大幅改善.

仔细观察图 9 和图 10,可以发现这个预测图有一定的误差,是因为未考虑 ϵ 造成的. 一般而言,在线性预测模型中,总是假定 ϵ 服从某种分布(较常见的,如上面模型假设的正态分布),从而给出对 ϵ 的估计. 但是,在实际应用场景中,这个估计一般比较困难. 因此,我们转而直接考察预测结果的误差. 取定一个局部坐标系,用坐标来表示地图上的点,并考虑 x, y 两个方向上的误差. 我们统计了随着预测时间的增长误差的累计情况,如图 11 所示:

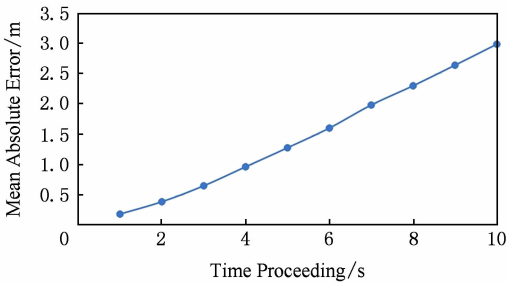


Fig. 11 Error accumulation on x -axis with time varying
图 11 x 方向上误差随预测时间的累积

从图 11 中容易看出误差的累积是高度线性的. 值得说明的是,每一次预测的误差的线性关系不尽相同,图 11 中表示的是一个平均的结果. y 方向的结果与之类似,在此不再给出. 基于该结果,虽然不能很好地估计速度方向线性预测中的误差 ϵ ,但是可以通过修正最终的预测结果来使误差降低. 一个操作上的难题是如何利用这个信息,我们可以从反

馈的角度思考. 如果我们预测 10 s 之后车辆的位置,那么其实可以不断修正这个预测结果,修正的方式就是利用图 11 的结果. 例如,可以在预测后 3 s 内统计出误差的线性增长关系,在最后一并给出最终的预测结果. 最初的 3 s 可以不进行预测,利用旧的位置信息进行路由决策,然后预测系统的误差补偿机制开始运作后,随着窗口滑动,可以得到连续的修正预测结果. 误差修正后的实验结果及局部放大图如图 12 和图 13 所示,预测时长 $T=10 \text{ s}$. 其中,图上各点的意义与之前的轨迹图类似. 在矫正了误差之后,预测结果已经很好.

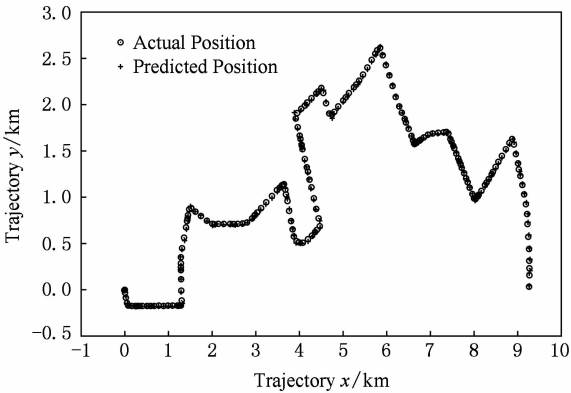


Fig. 12 Trajectory prediction result after error correction

图 12 误差修正后的轨迹预测结果

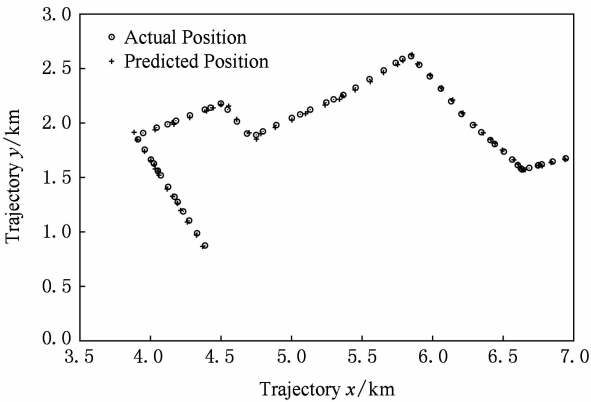


Fig. 13 Trajectory prediction result after error correction; local map

图 13 误差修正后的轨迹预测结果:局部图

我们新提出的预测算法与已有算法有较大不同,体现在 2 方面:1)新算法是基于加速度正态性假设进行的预测;2)新算法是实时性的预测,可以直接给出预测后的地理位置,方便基于位置的路由协议的使用. 第 3 节,将介绍如何将我们的预测方法用于实际路由协议,并验证其在实际路由决策中的有效性.

3 基于位置信息的即时路由协议

即时路由协议 (IR) 是一种基于 GPS 信息的路由协议, 从本质上看也是一种基于地理位置的路由协议. IR 协议专门被设计为适用于高速变化拓扑的车联网, 所以 IR 放弃了像其他协议如 AODV^[11] 那样去修复端到端的连接, 这一过程被证明是很耗时的. 相反, IR 可以采用第 2 节提到的预测算法, 利用保存的周边节点的地理位置信息去计算预估合适的下一跳, 并转发给该下一跳节点. 另外, IR 还可以平衡查找路由和计算路由之间的比例, 在节点位置关系变化不大时, 利用之前计算出的下一跳结果进行转发. 下面, 我们将详细介绍 IR 协议的设计.

IR 共有 3 种形式的数据包, 分别为 Beacon 包、Interest 包和 Data 包. 其中 Beacon 包是节点间用来

交换地理位置信息的, 周期性发送; Interest 包是节点用来请求数据而发送的, 包含所需数据的名称; 而 Data 包顾名思义就是数据包, 用来响应 Interest 包, 里面包含着请求者所需的数据.

IR 工作机制如图 14 所示. 首先, 每个节点周期性地发送 Beacon 包, 携带自身 ID、GPS 信息和运动速度信息, 发送给邻居节点. 所有接收到该 Beacon 包的节点都被视为该节点的邻居节点. 每个节点创建并维护一张邻居列表 (neighbor list), 收到 Beacon 包后就该邻居节点更新到该表中; 若之后一段时间未收到来自某邻居的 Beacon 包, 则将该节点从邻居列表中删除 (表项的时间戳过期, 自动删除). 把请求数据的节点 consumer 称为源节点, 数据提供者 (provider) 称为目的节点, 其他转发了 Interest 包或者 Data 包的节点 forwarder 都称为中间结点. 上述类似的思想在命名数据网络 NDN^[12-17] 中被广泛使用.

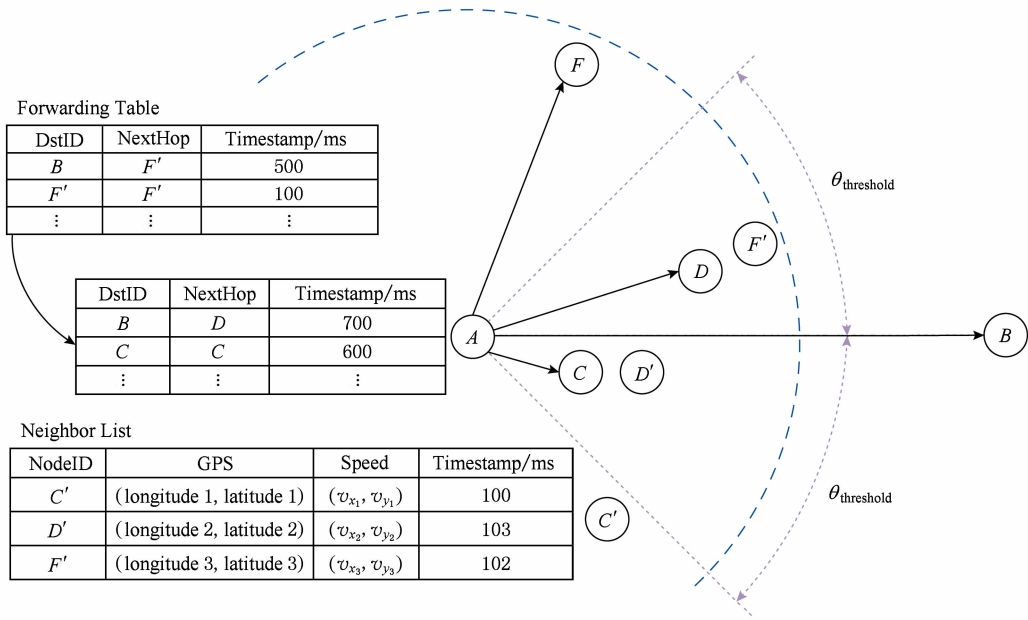


Fig. 14 Forwarding calculation procedure of instant routing
图 14 即时路由的计算转发过程

当源节点想要发送包含所需数据名称的 Interest 包时, 如果是该节点第 1 次请求某一数据, 它并不知道谁拥有内容, 所以该节点会通过广播 Interest 包寻找数据, 周围节点收到广播的 Interest 包后, 则会继续广播 Interest 包, 直到抵达最终目的节点, 然后目的节点将 Data 包按原路径返回源节点. 这样源节点就获得了目的节点的地理位置信息, 从而用于请求后续数据分组, 并且每一次分组带回来的 GPS 信息也都会实时更新目的节点的位置, 以确保目的节点位置信息最新.

如果节点不是第 1 次请求数据, 而是之前已经请求并获得了数据的部分分组, 并且当通过发送 Interest 包请求数据的剩余其他分组时, 就会根据已获得的目的节点 GPS 信息 (之前请求数据包时返回的信息) 去查找路由表, 如果路由表中存在相应路由表项, 并且该表项未过期且合法, 则节点就根据路由表中信息来发送 Interest 包. 如果不存在相应表项或表项不合法, 则协议会启动计算下一跳过程, 根据节点邻居表中的历史 GPS 信息, 来计算预测周围节点中距离目的地最近的节点, 然后将其作为下一跳,

并把 Interest 包发送给那个节点,该节点的转发过程与上述过程相同,查表或计算预测,这样 Interest 包就能经过不断地转发到达目的地,目的节点则会按照 Interest 来时的路径把数据分组反向传递回源节点,每一次返回的数据包都会将目的节点的最新位置带回源节点.

下面详细介绍计算预测算法和建立反向链路过程,这些都是 IR 显著区别于其他路由算法的内容. 计算预测算法是 IR 路由协议的最大特色之一, Beacon 包的发送是具有一定周期的,并且周期不能太短,不然就会造成很大的协议包开销,影响带宽利用率. 并且在 Beacon 包的周期内,节点会移动,在车联网应用场景下移动速度相对较快. 考虑到这个情况,我们初始设计了一个比较简单的线性预测算法来专门预测当前时刻周围节点的地理位置,从而能准确找到合适的下一跳,使得发送数据包更精确和有效.

首先做这样一个假设,即在 Beacon 包的周期范围内,车辆节点的速度是恒定的. 因此我们可以根据车辆的速度来预测位移,进而得出位置. 具体方法为:首先基于节点历史 Beacon 包周期获取的邻居节点地理位置信息,先计算出上一周期邻居节点的速度向量(包含大小和方向),从而计算出从上一个 Beacon 周期到现在为止邻居节点的相对位移,通过向量运算得到邻居节点当前的位置,然后在当前节点和目的节点位置的连线方向上,在预先设定好的夹角阈值 $\theta_{\text{threshold}}$ 范围内,通过计算在邻居节点中找出距目的节点最近的节点,并将计算出的节点作为

新的下一跳节点更新到路由表中. 不过该方法除了需要 GPS 信息的交换之外,每个节点还须维护其邻居节点的速度信息. 如图 14 所示, C', D', F' 是 A 节点在上一 Beacon 周期所获取的邻居节点地理位置信息, F' 是过去路由表中的下一跳. 现在 A 想发送数据给 B, 但发现路由表项过期, 于是启动预测计算算法, 预测出 C', D', F' 所对应的当前邻居节点位置 C, D, F , 并在夹角阈值 $\theta_{\text{threshold}}$ 范围内, 通过计算和判断, 将 D 点作为自己的下一跳.

反向链路,顾名思义,是目的节点收到 Interest 包后将数据返回给源节点所要经过的路径. 对于建立反向链路过程, IR 的做法和 AODV 做法虽看上去相似,但却有本质的不同. 首先,在 AODV 中,是通过协议包事先建立好源和目的节点间的来回路径. 而在 IR 中,是通过查表和计算路由相结合的方法, Interest 包发往目的节点途中在中间节点建立了返回源节点的路由,数据返回过程很快,所以大部分节点位置几乎没变,只有少部分节点变化很大,导致链路中间断开一小部分,这时可以在断开位置利用预测计算来得到下一跳,计算几次后就可能重新找到原来的反向路径. 如图 15 所示, A 点向目的节点 B 点发送 Interest 包,沿途的中间节点创建了返回 A 点的路由信息,从而建立了 A 到 B 的反向链路, B 收到请求包后,准备将数据发回 A, 根据路由表应该发给下一跳 H' , 但根据最新的邻居表显示 H' 已不在 B 点可达范围, 故路由信息不合法, 所以 B 启动计算路由过程找到 G 点作为其下一跳, 由于 G 点没有参与 Interest 转发, 故不存在到 A 点的

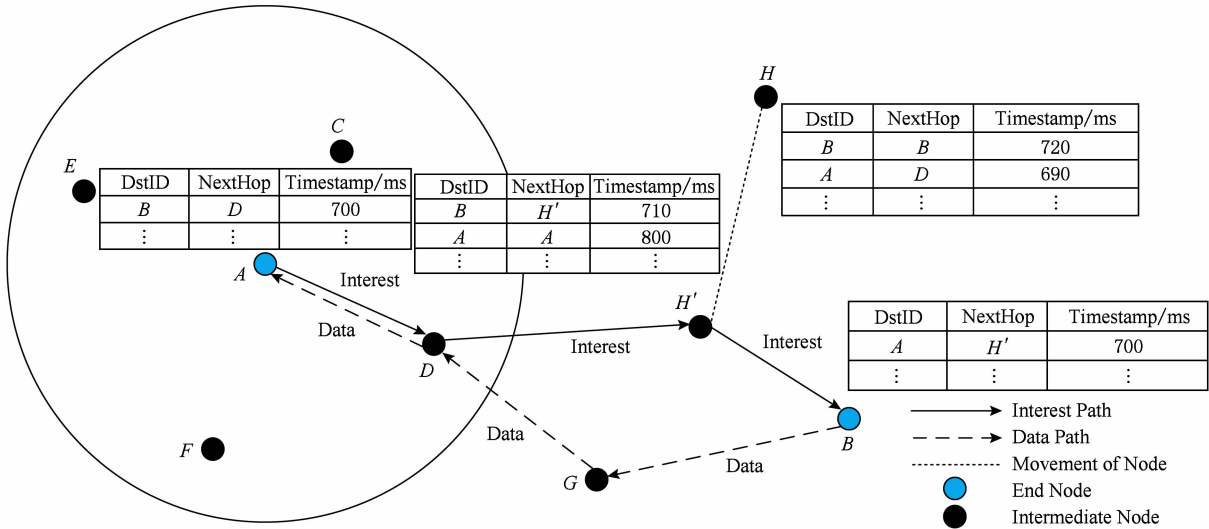


Fig. 15 Procedure of setting up the reverse link

图 15 反向链路建立过程

路由信息,所以 G 点也启动计算路由过程找到 D 点作为其下一跳, D 是反向链路的一部分,所以 D 根据其路由表信息就能将数据返回到 A 点.

以上描述的简单线性预测较初步,我们将使用第 2 节提出的新型预测算法进行节点位置预测,以改进基础版 IR 的性能.

4 实验与结果

仿真实验是在安装于 Ubuntu14. 04 上的 NS3. 26^[18] 和 SUMO0. 29. 0^[19] 平台上进行的. 我们首先在 OpenStreetMap^[20] 官网上选择并下载了上海和重庆的局部地图数据. 其中,上海的局部地图对应了地理范围较大、节点较为稀疏的情形,如图 16 所示. 重庆的局部地图对应了地理范围较小、路网较密的情形,如图 17 所示. 我们根据地图数据同时编写了车流量文件,车辆规模为 20~100 辆车,随机平均分布在地图车道上,并在 0~100 s 内通过设定好的路线运动,遇到红绿灯会等待,并且会根据道路选择拐弯和掉头,以达到尽可能地模拟真实环境的目的. 最后,利用 SUMO 生成了车辆仿真 Trace 文件.



Fig. 16 Local map topology of Shanghai
图 16 上海局部地图拓扑

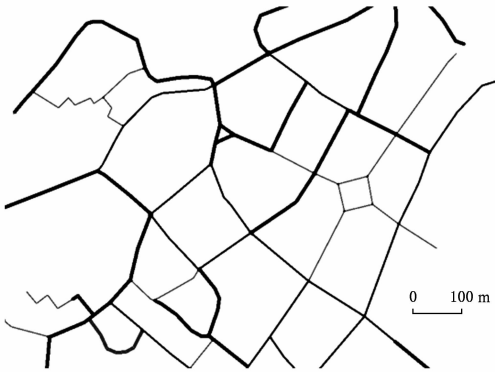


Fig. 17 Local map topology of Chongqing
图 17 重庆局部地图拓扑

在 NS3 中,我们设置了默认的底层模型,选择 802. 11a 作为物理层, AdhocMac 作为链路层, 创建并安装无线网络设备. 然后选择上面 SUMO 生成的 Trace 文件作为节点移动拓扑模型. 接着安装我们设计的基础版 IR 路由协议(记为 IR), 新型预测 IR (记为 IR_Prediction), 还有对比组 GPSR 协议(记为 GPSR). 完成之后, 添加上我们设计的一个简单应用, 该应用采用 client-server 模式, client 节点周期性发送 Interest 包, sever 节点是每收到一个发送给自己的 Interest 包后就返回一个 Data 包, 然后设置仿真时间为 0~100 s 后, 就可以进行仿真了. 具体实验参数如表 1 所示:

Table 1 Simulation Parameter Configuration Table
表 1 仿真参数配置表

Parameter	Value
802. 11a Data Rates/Mbps	6
Channel Bandwidth/MHz	5
Transmit Power/dBm	30
Interest Sending Interval/s	0. 1
Interest Packet Size/B	64
Data Packet Size/B	1 000
Beacon Interval/s	1
Beacon Packet Size/B	24
Vehicle Node Velocity/(m·s ⁻¹)	<25
Number of Nodes	20, 40, 60, 80, 100
Client Number	10
Sever Number	10
Simulation Time/s	100

我们在改变节点规模的情况下, 分别对 IR, IR_Prediction, GPSR 进行了 10 组实验, 然后对实验结果取平均, 统计了每个节点的各项指标. 我们选取了具有代表性的 3 个指标: 收包率(round-trip delivery ratio, RTDR)、往返时延(round-trip time, RTT)和开销率(overhead). 其中, 收包率的定义为接收到的所有 Data 包总数与发送的 Interest 包总数的比值; RTT 的定义为接收到所需 Data 包的时间戳与发送对应的 Interest 包的时间戳的差值; 开销率定义为每当 Sever 收到一个 Interest 包或者 Client 收到 Data 包, 网络中需要额外传输多少字节的数据, 包括协议包以及未抵达目的节点而丢弃的各种 Interest 包和 Data 包. 3 个统计量的定义和计算方法如表 2 所示. 对于开销率, $R_{totalByte}$ 代表某节点收到的

所有包(包括协议包、转发包、请求包和数据包)的字节数, R_{its} 代表某 Server 节点收到的 Interest 包的数目,Interest 包的总字节数是 105(包含 41 B 的包头), R_{data} 代表某 Client 节点接收到发往自己的 Data 包的数目,Data 包的总字节数是 1 041(同样包含 41 B 的包头), Hop_{up} 代表 Interest 包到达目的节点平均所经历的跳数, Hop_{down} 代表 Data 包发回源节点平均所经历的跳数.

Table 2 Definitions of Experimental Indexes

表 2 实验指标定义

Index	Definition
RTDR	Data Packet Received/Interest Packet Sent
RTT	Data Packet Timestamp-Interest Packet Timestamp
Overhead	$\frac{\sum_{n=1}^N R_{totalByte}(n)}{\sum_{n=1}^N (105R_{its} \times Hop_{up} + 1041R_{data} \times Hop_{down})} - 1$

下面,我们将分别给出在不同节点规模下,收包率、RTT 和开销率的实验结果与分析.

4.1 收包率实验

图 18 和图 19 分别表示了 3 种路由协议在上海和重庆两张地图上的收包率仿真结果.从图 18 和图 19 都可以看出,使用新型预测算法的 IR_Prediction 协议效果明显好于基础版 IR 和 GPSR 协议.需要说明的是,使用上海地图拓扑时的收包率整体较低,是因为地图过大及节点过于稀疏造成的.而从收包率趋势来看,随着节点密度增大,图 18 上呈现了一种收包率上升的趋势.这是因为上海地图面积较大,在节点密度较小时,很多链路都可能处于最大通信距离的范围之外.当节点密度增大时,可用连接变

多,收包率有所提高.图 19 中,随着节点密度增大,收包率呈现出下降的趋势.这是因为在重庆这张较小的地图中,许多节点都处在一个 CSMA 冲突域中,随着节点规模增大,冲突避让发生频率增大,使得收包率呈现下降的趋势.不过总体上讲,图 19 中的收包率可以维持在较高的水平,大于 70%.

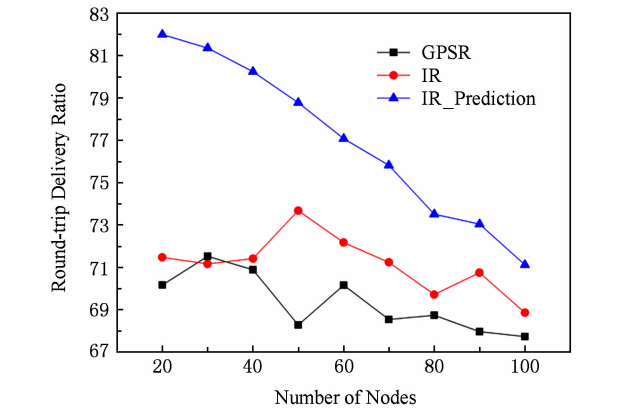


Fig. 19 Round-trip delivery ratio of Chongqing topology
图 19 重庆拓扑的收包率

4.2 RTT 实验

图 20 和图 21 分别表示了 3 种路由协议在上海和重庆两张地图上的 RTT 仿真结果.RTT 从一定程度上反映了网络延迟情况,不过 RTT 的统计与收包率有一定关联,因此不能单纯地从 RTT 断定协议工作状况的优劣.从图 20 和图 21 的纵轴可以看出,上海地图的 RTT 从总体上大于重庆地图.在上海地图上进行实验,IR_Prediction 协议与基础版 IR,GPSR 在 RTT 上并没有拉开差距,互有高低,在节点密度大时 IR_Prediction 协议的 RTT 还大于基础版 IR 和 GPSR,这是由于收包率提高造成的.这部分多收到的包会在一定程度上拉高 RTT 的数值.

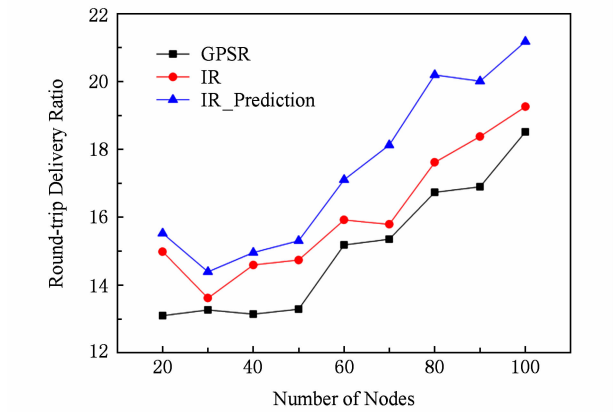


Fig. 18 Round-trip delivery ratio of Shanghai topology
图 18 上海拓扑的收包率

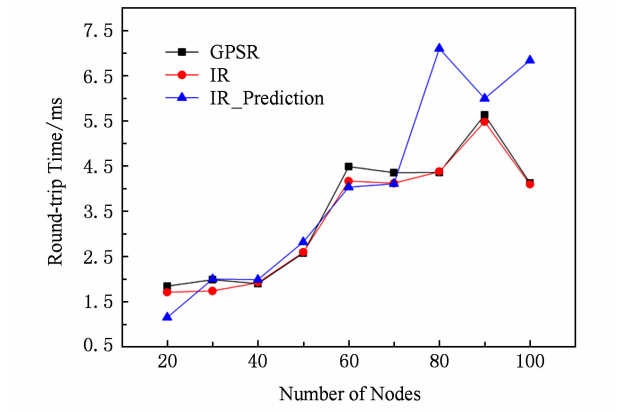


Fig. 20 Round-trip time of Shanghai topology
图 20 上海拓扑的 RTT 结果

而在图 21 中,重庆地图呈现了相反的情况,在节点密度大时,由于收包率降低,部分没收到的包就不再进入 RTT 的统计之中,反映出的结果反而是降低了 RTT. 上述结果的根本原因是 RTT 的统计仅考虑收到的包,而未收到的包并没有加一个较大的值进入统计.

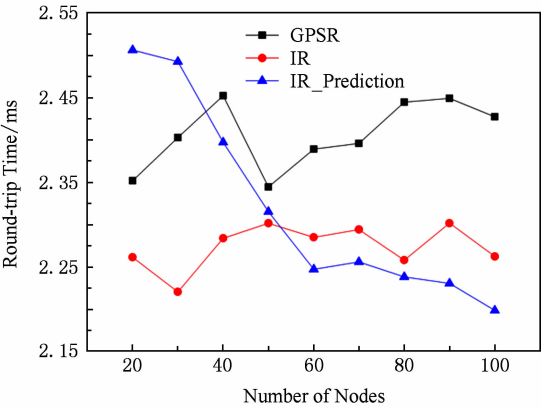


Fig. 21 Round-trip time of Chongqing topology
图 21 重庆拓扑的 RTT 结果

通过上述收包率和 RTT 的实验,可以说明新的预测算法在帮助 IR 协议进行路由决策方面相比基础版 IR 有大的改进,可以显著提升收包率和控制延迟.

4.3 开销实验

从图 22 和图 23 中可以看出,3 种协议的开销率都随着节点密度的增加而逐渐增加. 其中,图 22 表示的上海的开销率大于图 23 表示的重庆的开销率,因为上海地图面积较大,收包率较低. 在两张图中,IR_Prediction 的开销率均大幅优于基础版 IR 和 GPSR,因为 IR_Prediction 协议的重传率低于 IR 和 GPSR,使得开销大大降低.

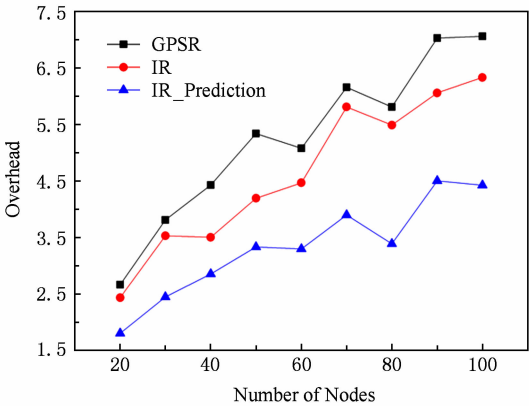


Fig. 22 Protocol overhead of Shanghai topology
图 22 上海拓扑的协议开销

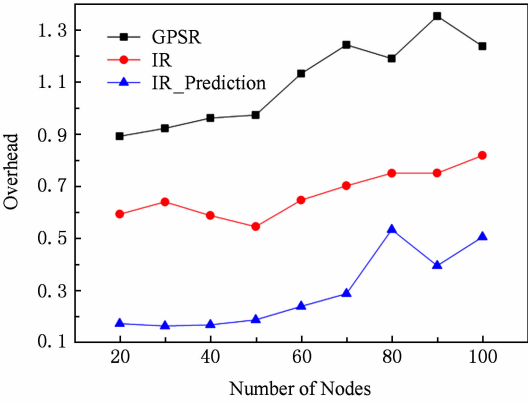


Fig. 23 Protocol overhead of Chongqing topology
图 23 重庆拓扑的协议开销

5 总 结

本文针对车载自组织网络中基于位置的路由协议,提出了一种新的车辆位置预测算法. 新算法首先通过测量得出车辆加速度服从正态分布的结论,利用线性回归方法进行预测,并通过反馈系统修正预测误差,克服了之前预测算法中普适性较差、预测误差大的问题. 然后,提出了一种新型的基于位置的即时路由协议,在新协议上验证了预测算法的正确性. 实验结果表明,新的位置预测算法在保持网络较高收包率、较低延迟的基础上,使得协议开销显著下降. 在未来的工作中,我们期望能够在实车上实现即时路由协议和位置预测算法,实现实际车辆自组织网络系统,利用真实应用场景验证本文预测算法和路由协议的有效性.

参 考 文 献

[1] Awang, A, Husain K, Kamel N. Routing in vehicular ad-hoc networks: A survey on single- and cross-layer design techniques, and perspectives [J]. IEEE Access, 2017, 5(1): 9497-9517

[2] Garrosi M, Kalac M, Lorenzen T. Geo-routing in urban vehicular ad-hoc networks: A literature review [C] //Proc of IEEE ICNC'17. Piscataway, NJ: IEEE, 2017: 865-871

[3] Lin Dan, Kang Jian, Squicciarini A, et al. MoZo: A moving zone based routing protocol using pure V2V communication in VANETs [J]. IEEE Trans on Mobile Computing, 2017, 16(5): 1357-1370

[4] Kung T, Karp B. GPSR: Greedy perimeter stateless routing for wireless networks [C] //Proc of ACM MobiCom'00. New York: ACM, 2000: 243-254

[5] Alsaqour R, Abdelhaq M, Saeed R, et al. Dynamic packet beaconing for GPSR mobile ad hoc position-based routing protocol using fuzzy logic [J]. Journal of Network and Computer Applications, 2015, 47(C): 32-46

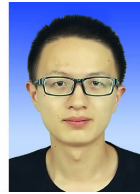
- [6] Li Jia, Wang Ping, Wang Chao. Comprehensive GPSR routing in VANET communications with adaptive beacon interval [C] //Proc of IEEE CPSCOM'16. Piscataway, NJ: IEEE, 2016: 211-216
- [7] Menouar H, Lenardi M, Filali F. Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks [C] //Proc of IEEE VTC'07. Piscataway, NJ: IEEE, 2007: 2101-2105
- [8] Wang Xiufeng, Wang Chunmeng, Cui Gang, et al. Practical link duration prediction model in vehicular ad hoc networks [J]. International Journal of Distributed Sensor Networks, 2015, 2015(1): 311-325
- [9] Eiza M, Ni Qiang. An evolving graph-based reliable routing scheme for VANETs [J]. IEEE Trans on Vehicular Technology, 2013, 62(4): 1493-1504
- [10] Xue Guangtao, Luo Yuan, Yu Jiadi, et al. A novel vehicular location prediction based on mobility patterns for routing in urban VANET [J]. EURASIP Journal on Wireless Communications and Networking, 2012, 222(1): 422-436
- [11] Perkins, C, Belding-Royer E, Das S, et al. Ad hoc on-demand distance vector (AODV) routing [EB/OL]. 2003 [2017-05-23]. <https://tools.ietf.org/html/rfc3561>
- [12] Zhang Lixia, Afanasyev A, Burke J, et al. Named data networking [J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 66-73
- [13] Grassi G, Pesavento D, Pau G, et al. VANET via named data networking [C] //Proc of IEEE INFOCOM'14 Workshop. Piscataway, NJ: IEEE, 2014: 410-415
- [14] Ahmed S, Bouk S, Kim D. RUFs: Robust forwarder selection in vehicular content-centric networks [J]. IEEE Communications Letters, 2015, 19(9): 1616-1619
- [15] Ahmed S, Bouk S, Yaqub M, et al. CODIE: Controlled data and interest evaluation in vehicular named data networks [J]. IEEE Trans on Vehicular Technology, 2016, 65(6): 3954-3963
- [16] Bouk S, Ahmed S, Yaqub M, et al. DPEL: Dynamic PIT entry lifetime in vehicular named data networks [J]. IEEE Communications Letters, 2016, 20(2): 336-339
- [17] Yaqub M, Ahmed S, Bouk S, et al. Interest forwarding in vehicular information centric networks: A survey [C] //Proc of ACM SAC'16. New York: ACM, 2016: 724-729
- [18] NS-3 Consortium. Network simulator 3 [EB/OL]. [2017-05-23]. <https://www.nsnam.org/>
- [19] Institute of Transportation Systems at the German Aerospace Center. Simulation of urban mobility (SUMO) [EB/OL]. [2017-05-23]. http://www.sumo.dlr.de/wiki/Simulation_of_Urban_MObility_Wiki
- [20] OpenStreetMap Community. OpenStreetMap world map [EB/OL]. [2015-05-23]. <http://www.openstreetmap.org>



Li Yang, born in 1989. PhD candidate at Tsinghua University. His main research interests include vehicular ad hoc network and network measurement.



Wang Zhe, born in 1995. PhD candidate at Tsinghua University. His main research interests include vehicular ad hoc network and statistics.



Zhang Chuwen, born in 1992. PhD candidate at Tsinghua University. His main research interests include high-performance switches/routers, named data networking and vehicle networks.



Dai Huichen, born in 1988. Postdoctoral research fellow in the Department of Computer Science and Technology, Tsinghua University. Received his PhD degree from the Department of Computer Science and Technology, Tsinghua University, in 2016. His main research interests include routing protocols in ad hoc wireless networks, content cache analysis, and SDN.



Xu Wenquan, born in 1995. PhD candidate at Tsinghua University. His main research interests include NDN and VANET.



Ji Xuefeng, born in 1994. PhD candidate at Tsinghua University. His main research interests include NDN and VANET.



Wan Ying, born in 1993. PhD candidate at Tsinghua University. His main research interests include named data networking, software defined networking and vehicular networks.



Liu Bin, born in 1964. Received his MSc and PhD degrees in computer science and engineering from Northwestern Polytechnical University, Xi'an, China in 1988 and 1993, respectively. Professor and PhD supervisor of Tsinghua University, Beijing, China. His main research interests include high-performance switches/routers, network processors, named data networking and software-defined networking.