无线车载网络中一种基于跨层优化的网络编码TCP协议

陈 杰 谢显中 黄 倩 黎 佳
(重庆邮电大学宽带接入网络研究所 重庆 400065)

摘 要 无线车载网络(Wireless Vehicular Networks,WVN)具有研究价值和重要应用价值,目前关于WVN中的网络编码TCP协议的研究并不多,且对跨层的优化也没有被重视。文中首先针对无线车载网络中不同的丢包原因,给出了一种基于随机线性网络编码的跨层联合优化方法VC-TPC/NC,可以针对不同丢包原因采取不同的处理方式;进一步,重新设计了网络编码层发送端的发送策略,并通过理论分析说明了VC-TCP/NC在时延和网络吞吐量方面的优势;最后,在不同场景下的仿真结果表明,VC-TCP/NC的性能相比于传统TCP与TCP/NC等性能有较大提高。

关键词 无线车载网络,TCP优化,网络编码,跨层设计,VC-TCP/NC
中图法分类号 TN915 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.02.014

Network Coding TCP Protocol Based on Cross-layer Optimization in Wireless Vehicle Networks

CHEN Jie XIE Xian-zhong HUANG Qian LI Jia
(Institute of Broadband Access Technologies,Chongqing University of Posts and Telecommunications,Chongqing 400065,China)

Abstract Wireless vehicle network (WVN) has important research and application value. At present,there are few researches on network coding TCP protocol in wireless vehicle network. At the same time,cross-layer optimization has not been paid attention seriously. This paper first addressed different causes of packet loss in wireless vehicle networks,and gave a cross-layer joint optimization method VC-TPC/NC based on stochastic linear network coding. Aiming at different reasons for packet loss,different treatment were given. Further,the sending strategy of network coding layer sender was redesigned,and the advantages of VC-TCP/NC in the delay and network throughput were illustrated through theoretical analysis. Finally,simulation results in different scenarios show that the performance of VC-TCP/NC is greatly improved compared to the performance of traditional TCP and TCP/NC.

Keywords Wireless vehicular networks,TCP optimization,Network coding,Cross-layer design,VC-TCP/NC

1 引言

作为第5代移动通信系统(5G)的一种重要应用场景,目前无线车载网络(Wireless Vehicular Networks,WVN)引起了人们的高度关注[1-2]。它是一个典型的无线Ad hoc网络(WAN),无线车载网络不但具有动态网络拓扑、信道的质量不可预测等特点,而且无线车载网络还具有网络拓扑结构变化频繁、运动速度较快等特点,且无线车载网络对延迟和吞吐量的要求也很高。TCP协议早已成为现代网络中不可或缺的网络传输协议,根据无线Ad hoc网络(WAN)和无线车载网络(WVN)的特点,设计其TCP协议在一定程度上具有很大挑战性[3-4]。网络编码(Network Coding,NC)的优点主要体现在减少无线发送的能量消耗、提升网络的吞吐率和增强网络的鲁棒性和容错性等方面。基于网络编码对数据传输的可靠性,将其与TCP协议结合而成的网络编码TCP协议[5-6]为改善无线网络下TCP的性能提供了一种新的方法。

目前,已有一些针对无线Ad hoc网络中网络编码TCP协议的研究[6-16]。在无线链路中TCP由于受到随机分组丢失的影响[7],其性能会下降,利用网络编码的纠删性可以减轻其影响。一种称为TCP/NC的编码传输方式由Sundararajan等[8]提出,其用一种简单巧妙的方法将网络编码结合到现有的协议栈中,能够让TCP协议在无线传输的情况下也可以得到较高的吞吐量。文献[9]给出了TCP/NCW协议,在该方法中,编码窗口可以根据解码时间进行修改和调整,并最终达到最佳的窗口大小,这种方案能有效解决数据传输和解码操作的同步性问题,从而可以使TCP/NC获得更好的网络性能。文献[10]提出了一种TCP/NC隧道方法,其是在单个TCP/NC流上进行端到端的TCP会话;而在两个特殊网关之

到稿日期:2018-02-23 返修日期:2018-05-17 本文受国家自然科学基金(61271259,61471076,61601070),重庆市教委科学技术研究项目(KJ1600411),重庆市基础与前沿研究计划项目(CSTC2016jcyjA0455),长江学者和创新团队发展计划(IRT1299),重庆市教委重点实验室专项经费(JK12010000062),重庆市研究生科研创新项目(CYS17223)资助课题资助。

陈 杰(1993-),男,硕士生,主要研究方向为无线网络中TCP编码与优化,E-mail:854031548@qq.com;谢显中(1966-),男,博士,教授,主要研究方向为无线和移动通信技术,E-mail:xiexz@cqupt.edu.cn(通信作者);黄 倩(1991-),女,博士生,主要研究方向为云计算以及干扰对齐;黎 佳(1993-),女,硕士生,主要研究方向为mMTC高可靠低延迟等。

间穿过的有损网络上则不用进行流量管理。仿真结果表明,TCP/NC 隧道可以有效缓解通过有损网络时端到端 TCP 会话吞吐量下降的问题,但是在每个末端主机上的 TCP 却没有任何改变。在多天线多信道(MRMC)的无线 Mesh 网络(WMN)中,当发生非拥塞分组丢失时,TCP 性能显著下降,因此文献[11]提出了基于 TCP 跳数的网络编码,以提高 MRMCWMN 中基于跳数网络编码的 TCP 性能。它通过将随机线性网络编码中同下一跳节点流内的数据包组合在一块,然后逐条转发并确认数据块,引入了分布式速率和路径选择算法,有效地利用了网络资源,使得来自拥塞路径和非拥塞路径的负载达到平衡。仿真表明该方法不仅实现了吞吐量的增益,在高丢失率下也依然保持端到端延迟和时延抖动相对较小。文献[12]提出了一种在线网络编码(TCP VON),它将在线网络编码整合到 TCP 中,与基于生成的网络编码相比,由于在线网络编码中的数据包可以连续解码但不是逐代生成的,因此在线网络编码的解码延迟较低。比较 TCP VON 和基于 TCP 的自动重复请求(ARQ)网络编码在时延和吞吐量上的性能可知,TCP VON 在平均解码和网络吞吐量上要优于 TCP ARQNC。文献[13]分析了 TCP/NC 对有损链路纠错编码的演化过程。其忽略了慢启动阶段,使用三维马尔可夫链模型来描述拥塞窗口和可用窗口的状态转换过程;以最大窗口的大小、端到端擦除率和冗余因子作为输入的参数,利用数值分析模型计算网络编码 TCP 的吞吐量,并对影响 TCP 网络编码性能的因素进行了定量分析,结果表明数值分析模型达到了相当高的精度。文献[14]提出了一种快速网络编码的 TCP (TCP-FNC),其目的是通过使用网络编码来降低 TCP 的解码时延。TCP-FNC 不仅保留了 TCP/NC 的框架模型,而且还有两个方案用于满足在线网络编码的需求;首先开发了一个名为 FCWL 的基于反馈的方案,其根据接收器的反馈锁定滑动编码窗口,可以有效减少等待延迟,并且同时适用于基于 RTT 和基于丢失的 TCP 的协议;其次,提出了一种优化的渐进译码算法 EFU,评估结果表明,TCP-FNC 可以实现比 TCP/NC 更短的解码延迟,而且不会牺牲其良好的性能,验证了 TCP-FNC 平均可以减少 33% 的解码延迟。众所周知,TCP 在有损信道上的性能较差,尤其当这些链路表现出突发行为时。文献[15]研究了在无线链路上将 TCP 和网络编码技术互结合时的网络性能。其利用 Kodo 网络编码实现,并将与 TCP 协议集成在一起,仿真结果表明网络编码可以有效地隐藏无线信道的损失,当使用突发模型时,网络编码能够提供更好的性能。具有网络编码的 TCP 期望通过发送源处编码的冗余组合分组来恢复丢失的分组,从而代替通过重传的方式。虽然 TCP/NC 可以在理论上工作,但是在时变丢包情况下,仍有两个问题未得到很好的解决:1)如何基于连续观察估计的最近网络状况来确定好的参数(例如冗余度因子 R);2)如何根据决定的参数动态调整正在进行的 NC 层过程。因此,文献[16]提出了一种改进的 TCP/NC 变体(TCP/NCwLRE),它能够自动调整 NC 层行为以适应时变信道,结果表明,TCP/NCwLRE 改善了时变非相关随机丢失信道中的 TCP 吞吐量。

上述文献中的方法在链路随机丢包的情况下效果明显,然而在由于链路拥塞而造成丢包的情况下,通过编码不仅不能提升网络质量,反而会发送更多的冗余量,从而加剧网络负担,这不仅会消耗更多的资源,同时编码和解码还会引入更高的时延,导致 TCP/NC 的性能下降,而且可能产生连接中断现象,因此直接用于无线车载网络中的效果不理想[17]。另一方面,目前关于在无线车载网络下的网络编码 TCP 协议的研究并不多,且跨层优化也没有引起足够的重视。

传统的 TCP 是基于有线信道设计的,有线介质可靠性高,因此将丢包都视为网络拥塞的指示。而在无线车载网络中,网络拓扑结构变化频繁、节点运动速度快以及由于环境原因使得无线信号在传递过程中发生衰减都会使得链路中的随机丢包增加、延迟增大。TCP 如果不能正确区分无线车载网络中环境中的丢包原因而启动拥塞控制机制,必然会造成网络性能的下降,因此根据无线车载网络中的情况特点和要求,文中探讨了在无线车载网络中基于网络编码 TCP 的跨层优化。考虑到 TCP/NC 在链路拥塞情况下的性能低于传统的 TCP 拥塞控制方法,本文首先对 TCP/NC 进行了全面分析;在此基础上,给出了一种基于随机线性网络编码的跨层联合优化方法——VC-TCP/NC,并重新设计了网络编码层发送端的发送策略;进一步,通过理论分析说明了 VC-TCP/NC 在时延和网络吞吐量方面的改进;最后,利用网络模拟器 NS-2 (Network Simulator 2)分别模拟链路拥塞和链路随机丢包率高的情景,对 VC-TCP/NC,TCP/NC 和 TCP-Vegas 协议进行了对比实验,结果表明在丢包率高、信道通信质量不佳的环境下,VC-TCP/NC 的性能优势较为明显,而在拥塞造成的丢包下,VC-TCP/NC 也接近于一个常规 TCP 的性能。

2　TCP/NC 协议分析

网络编码 TCP 协议(TCP/NC)[7]是将一个新的网络编码层加入到传输层与网络层之间,而传统 TCP 协议中的拥塞控制机制被其重新使用了。对传统 TCP 协议进行一些改变,在发送端发送的是网络编码层按冗余比例进行了随机线性编码后的数据包,传输路径上的节点不会加入编码与解码操作。接收方收到编码包后将系数向量加入解码矩阵中,在等待一组数据包被解出和被确认之前,就已经将信息向量存放在解码缓存区中。解码矩阵主要包括可见包、不可见包、已解码包,如图 1 所示。

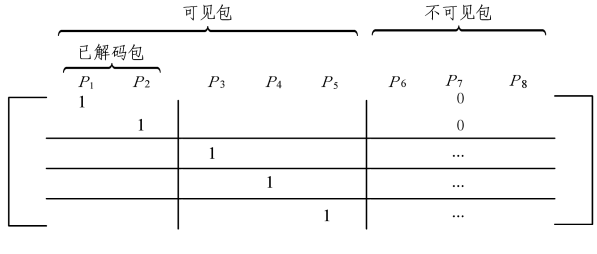


图 1　解码矩阵

Fig. 1　Decoding matrix

即便数据包在传输过程中出现数据包丢失的情况,但每个随机线性组合的数据包都可能使原始数据包被发现,因为,只要在足够大的有限域上,下一个成功接收到的数据包的线性组合能够让下一个没有被确认的数据包可见并确认。重复

的 ACK 会造成拥塞窗口增大。数据链路的丢包率也会通过另外的队列延迟来表现，而且会使得 RTT 的值增加。因此 RTT 值的含义变成有效传输一个自由度并返回 ACK 的用时，新的 RTT 值通常大于在信道质量良好条件下的 RTT 值，并且链路中的信道质量越差，新的 RTT 值就会越大，这种算法能够使 TCP 在超时之前有更多的时间处理数量信息，从而阻止了由于链路通信质量差造成的发送窗口减小，有效地掩盖了丢包，这就是 TCP/NC 协议的合理性。

一方面，在 TCP 协议中引入网络编码是为了在随机丢包较高的链路中使发送方依旧可以保持较高的发送速率和较高的吞吐量。相比于传统的重传机制，发送速率更快，往返时延(RTT)更小。然而在信道质量好、随机丢包率低的链路中，通过编码不但不能提升性能，反而会发送更多的冗余量，从而加剧网络负担，消耗更多的资源，使网络的性能降低，而且编码和解码还会引入更高的时延。收到编码的数据包后需要进行译码操作，CPU 的计算量会相应增加。例如需要传输 n 个信息，则编码的时间复杂度为 $O(n)$，译码的时间复杂度为 $O(n^3)$。节点编码时，在有限域中先随机选择 n 个系数，然后将随机选择的系数分别与 n 个消息进行线性组合，其操作时间复杂度为 $O(n)$，译码时采用的是高斯消去法，其中 $n*n$ 个系数矩阵的求逆过程所需的时间复杂度为 $O(n^3)$，因此译码的时间复杂度为 $O(n^3)$，这相当于额外增加了 CPU 的计算量，尤其对实时应用有较大的影响。

另一方面，将网络编码层加入到 TCP/IP 协议栈中将对 TCP-Reno 拥塞控制机制产生影响[5]。TCP-NC-RED 在延迟或容量增加时变得不稳定，NC 的冗余因子和时延均对 TCP-NC-RED 有影响；且 TCP/NC 太昂贵，无法在一些类型的终端设备中实施[9]，例如一些具有较少存储器和功率的设备；另外，由于有损网络牺牲了冗余组合分组，通过无损网络的 TCP/NC 可能会浪费稀少的链路带宽。TCP/NC 是基于生成的网络编码，故其吞吐量和可靠性要低于基于在线的网络编码[11]，而且解码的延迟也会高于基于在线的网络编码。文献[13]提出了 TCP-FNC，改进了编码算法使用优化的高斯-约旦逐步消除算法，可以进一步减少计算延迟，根据接收器的反馈锁定滑动编码窗口，比 TCP/NC 的解码时延更低，并且不会以牺牲其他的性能为代价。

TCP/NC 性能的好坏取决于当前的网络状况，当信道的通信质量良好时，TCP/NC 的性能表现也较好；当信道的通信质量较差时，TCP/NC 的性能表现也较差。

3 基于跨层优化的 VC-TCP/NC 协议

3.1 VC-TCP/NC 的主要思想

通过对传统的 TCP 进行改进，在 TCP 报文首部的标准格式中，有 6 个没有使用并且默认值为 0 的比特位，更改其中的 2 个保留位，将其命名为 VC 位[18]，用于指示当前链路的信道状态，设置默认值为 00。根据当前的链路状态动态改变 VC 位的值，VC 值为 00 时，表示信道通信质量好，链路没有发生丢包行为；VC 值为 01 时，表示链路中发生了拥塞丢包；VC 值为 10 时，表示链路中丢包为随机丢包；VC 值为 11 时，表示链路中断。引入跨层设计的思想，使得传输层能够根据 VC 值快速判别丢包原因，由 Veno 算法[19]可在发送端测量两个速率：发送速率的期望值(Expected)以及发送速率的实际值(Actual)，计算式如下：

$$Expected=cwnd/BaseRTT$$

$$Actual=cwnd/RTT$$

其中，$BaseRTT$ 是测量到往返时延中的最小值，$cwnd$ 是目前发送窗口的大小；RTT 是最近一次测量得到的往返时延值，两者间的差值定义为 $Diff$：

$$Diff=Expected-Actual$$

当网络的瓶颈带宽处堆积了数据包时，$RTT>BaseRTT$。如果用 N 来表示该处堆积的数据包个数，则有如下表达式：

$$RTT=BaseRTT+N/Actual$$

上式右侧表示网络瓶颈处的数据包堆积会引起往返时延的增加，整理得到如下的表达式：

$$N=Actual\times(RTT-BaseRTT)=Diff\times BaseRTT$$

设置一个阈值 β，通过比较 N 和 β 判断是否出现了拥塞，即判定出是随机丢包还是拥塞丢包。经过实验证明，阈值 β 取 3 较合理。

在传输层和网络层之间加入一层编码层，当出现丢包时，如果 $N<\beta$，表明链路中发生随机丢包，此时将发送端传输层中的 VC 值设置为 10，此时发送端先将数据包传入编码层，通过编码层编码后发送；如果 $N\geqslant\beta$，将丢包原因归于拥塞丢包，此时将发送端传输层中的 VC 值设置为 01，此时待发送的数据包不需要进行编码操作，发送端进行传统 TCP 拥塞控制后，将数据通过传输层传入网络层后发送。引入跨层设计的思想，以使传输层区分出丢包原因后，针对不同的丢包原因做出相应的处理，当判断出 VC 值为 01 时，即判断出是拥塞造成的丢包时，此时无需进行编码操作，直接将数据包从传输层传入网络层。其流程图如图 2 所示。

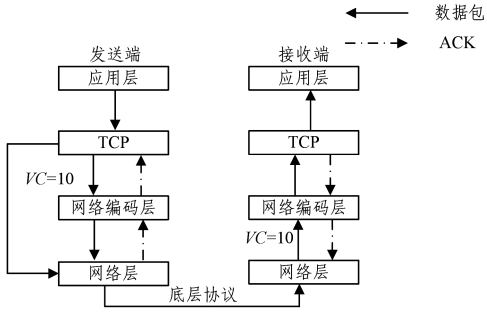


图 2 VC-TCP/NC 的跨层处理流程

Fig. 2 Cross-layer processing of VC-TCP/NC

需要强调的是，新加入的 NC 层没有对 TCP 中的接口做出更改，且 TCP 协议与下层协议栈之间的接口也没有进行改动。当 $VC=10$ 时，即判断出丢包原因为随机丢包时，网络编码层从 TCP 发送端接收 TCP 数据包，对其编码后传入 IP 层。由于编解码操作均在网络编码层完成，因此只需对 TCP 协议进行稍微的改动就可基本实现。

3.2 发送端的算法流程

当链路中出现丢包时，发送端根据 VC 值判断丢包原因，由于当 $VC=10$ 时需要编码操作，因此发送端需要设置一个

编码缓冲区,用于存放将要编码的原始数据包,同时用编码窗口进行随机线性编码操作,具体算法流程如图 3 所示。

1)初始状态为保持等待状态,直到遇到以下情况,执行相应操作,否则返回等待状态。

2)当有数据包从 TCP 层向 IP 层传送时,判断当前 TCP 连接的 ACK 包的 *VC* 值,执行以下步骤:

①当 *VC* 值为 00 时:链路通信质量好,未发生丢包,无需进行编码与拥塞控制,直接发送即可。

②当 *VC* 值为 01 时:a)链路中出现拥塞,数据包不通过网络编码层,TCP 连接进入拥塞避免阶段,执行传统的拥塞控制算法;b)一旦出现丢包,将 ssthresh 设为 cwnd/2,cwnd 设为 1。

③当 *VC* 值为 10 时:a)链路中出现随机丢包,需将数据包传入网络编码层进行编码。b)如果该数据包是用来创建连接的控制包,则将其传入到 IP 层。c)如果数据包没有存放在编码缓存中,则将该包加入到编码缓存中。d)设置参数 $N=N+R$,其中 R 为冗余系数。e)重复以下步骤 floor(N)次:在编码窗口中产生一个随机线性编码包;在编码包中添加一个自定义的编码包头部;将这个编码后得到的包投递到 IP 层;恢复到等待状态。

④当 *VC* 值为 11 时:链路中出现中断,直接放弃发送。

3)如果接收到接收端对数据包的确认,就把 ACK 确认的数据包从编码缓存区中删掉,并向 TCP 层发送 ACK 确认。

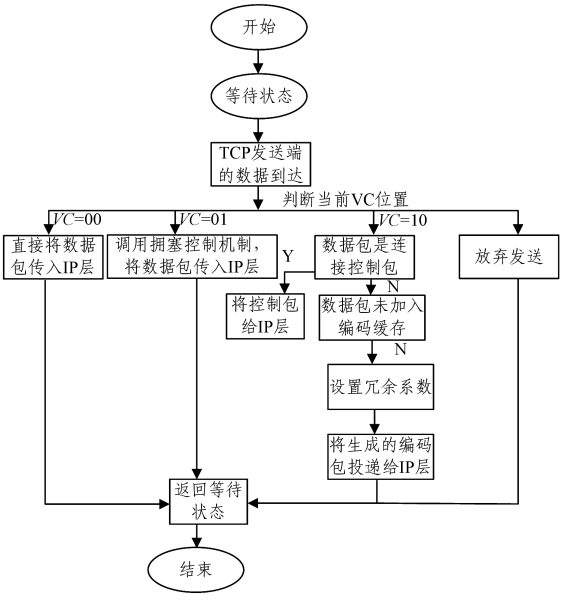


图 3　发送端流程图

Fig. 3　Flowchart of sending side

根据 *VC* 值判断当前的网络状态,并针对不同的状态做出不同的处理,以适应网络环境的实时变化,这是传统 TCP 以及 TCP/NC 无法实现的。

3.3　VC-TCP/NC 的理论性能分析

从前文可知,网络编码 TCP 协议的想法是将一个新的并且含有编码模块和解码模块的网络编码层加入到 TCP 层与 IP 层之间,由于数据包需要进行编解码操作,因此将数据缓冲区设置在编解码模块中,用于临时保存即将进行编码或者解码的数据。编码端中包括编码窗口,用进行一次编码操作所组合的数据包的数量来表示窗口的变化,同时改动解码端中的解码缓冲区的大小。发送端将数据从 TCP 层传入网络编码层进行编码操作后再将其发送给 IP 层,本文提出的 VC-TCP/NC 是在 TCP/NC 的基础上加入一种决策机制,以根据链路的实际情况来决定是否将数据包传入编码层,因为在链路由于拥塞而造成丢包的情况下,通过编码不但不能缓解丢包,反而会发送更多的冗余量,编码的效果不明显。当链路中的丢包为随机丢包时,通过编码能提升网络性能,增加吞吐量,这就是加入决策的意义。由于决策判断的依据是往返时延的大小变化,因此对往返时延的精确测量显得格外重要,引入 Veno 算法,通过计算期望吞吐量与实际吞吐量的差值来估算网络中的可用带宽。

当判定为拥塞时,由于 TCP/NC 新加入的网络编码层没有对 TCP 的接口做出更改,而且 TCP 协议与下层协议栈之间的接口也未改动,因此为引入跨层设计的思想提供了理论上的可行性,此时 VC-TCP/NC 未进行编码操作,其性能与一个在拥塞情况下常规 TCP 的性能接近。

当判定丢包情况为随机丢包时,即 TCP 层将数据传入网络编码层,需要注意的是,所有的编解码操作对 TCP 层和 IP 层都是公开的,网络编码层从 TCP 层获得发送数据包后,对数据进行重新编码,然后再将编码后的数据包以及在编码过程中采用的随机系数等参数进行打包处理后一并发给网络层。接收模块根据从网络层接收的编码包中的编码参数信息进行解码操作,然后再把解码复原出的原始数据发给 TCP 层。此时 VC-TCP/NC 进行了编解码操作,其性能与在随机丢包情况下 TCP/NC 的性能接近。

根据以上分析可知,针对不同的丢包情况选择恰当的处理方式可以获得更优的效果,尤其是对往返时延的测量特别重要,精确往返时延变化不仅能及时反映出链路的实际变化,还能提高网络的性能,避免资源的浪费。

4　仿真评价

4.1　仿真环境

由于在无线车载网络下关于网络编码 TCP 的研究很少,同时现有的关于 TCP 网络编码的研究大部分是基于静态场景的,因此本文分别从静态场景与动态场景两种不同的场景进行仿真比较。

首先在静态场景下根据 TCP/NC[7] 提出的思想及算法,重新完成对网络编码 TCP 协议的仿真,并在网络模拟器 NS-2 对应的目录下加入网络编码 TCP 的代码,将创建的代码文件放入 NS-2 中。在网络编码 TCP 算法的基础上,加入了判决条件以判断丢包原因,以同样的方式放入网络模拟器中,以便利用该软件进行仿真验证。由于网络编码 TCP 协议是在传统 TCP-Vegas 协议上改变而来的,因此选择 VC-TCP/NC 与 TCP/NC 和 TCP-Vegas 协议在不同情景下进行性能比较。其中仿真环境设置如下。

由于网络模拟器 NS-2 便于安装在 Linux 上,因此选用 Linux 操作系统。首先,设置无线链路带宽为 10 Mbps,其延迟为 10 ms,每个节点都采用 DropTail 的队列管理方式。

MAC 层选用 IEEE 802.11b 协议，通过媒体介入控制层的数据包出错的概率来模拟链路中的丢包率 ρ，编码窗口根据当前的网络状况，每次发送 R 个编码包，冗余系数 $R=1/(1-\rho)$，ρ 为丢包率。设丢包服从均匀分布，应用层使用持续的比特流业务 FTP 协议，网络层静态场景下使用 AODV 协议，动态场景下使用 DSR 协议。传输层选择的是 TCP-Vegas 协议，接收方的最大通知窗口设为 100 个数据包，每个数据包为 1500 kB，在静态场景下，选择无线链式拓扑单条数据流场景。仿真中，N_0-N_6 代表无线节点，节点接收半径为 250 m，间距为 150 m。仿真时间为 1000s，仿真结果取 10 次运行得到的平均值。

4.2 静态场景下的结果与分析

仿真拓扑结构如图 4 所示，节点 N_0 与 N_1-N_6 间进行通信。

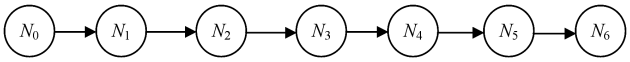


图 4 单条数据流拓扑结构

Fig. 4 Topology of single data flow

1)模拟链路为随机丢包的场景，设置 VC 值为 10，且使丢包率不断增加，仿真结果如图 5 所示。

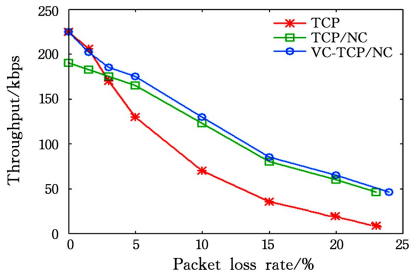


图 5 丢包率与吞吐量的对比曲线

Fig. 5 Comparison of packet loss ratio and throughput

从图 5 可知，3 种协议的网络吞吐量都随着丢包率的增加而下降。当丢包率小于约 3%时，TCP-Vegas 的吞吐量较 TCP/NC 的吞吐量更优。由于 VC-TCP/NC 是通过比较 N 与 β 的大小来确定的，阈值 β 的一般取值为 3，由于丢包率是逐渐增加的，因此当链路中的丢包率很低时，VC-TCP/NC 并不能立即检测出链路出现的随机丢包，此时 VC 值为 00，VC-TCP/NC 无需进行编码操作。但因为 TCP/NC 要花费额外的时间在编解码操作上，所以在刚开始时，TCP/NC 的网络吞吐量会低于 VC-TCP/NC 和 TCP-Vegas。随着丢包率的逐渐增加，VC 会变为 10，因此 VC-TCP/NC 的吞吐率会略低于 TCP-Vegas。当链路中丢包率超过一定值时，由图 5 可知，TCP-Vegas 吞吐率下降迅速，而 TCP/NC 和 VC-TCP/NC 的吞吐率即使有下降，幅度也不是很大，也即，通过设置合理冗余参数并把丢包率控制到一定范围内，并不会使 TCP/NC 和 VC-TCP/NC 由于网络丢包率的上升而使得吞吐量下降，且 VC-TCP/NC 的吞吐量也要高于 TCP/NC。

2)将 TCP/NC 和 VC-TCP/NC 的冗余度均设置为 0.4，分别模拟在丢包率为 5%和丢包率为 10%的网络条件下发送定量的数据包，其传输时间如图 6 和图 7 所示。

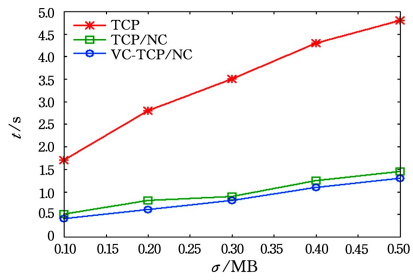


图 6 丢包率为 5%的传输时间

Fig. 6 Transmission time when packet loss is 5%

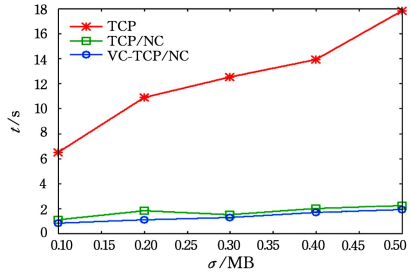


图 7 丢包率为 10%的传输时间

Fig. 7 Transmission time when packet loss is 10%

从图 6 和图 7 可知，3 种协议在同样的丢包率下，VC-TCP/NC 的传输速率最大，且 VC-TCP/NC 的传输速率大于 TCP/NC，因为在刚开始时丢包是随机发生的，并不会立即检测到丢包，因此 VC-TCP/NC 并不会进行编解码操作；而 TCP/NC 则要花费额外的时间进行编解码操作，从而导致传输速率较低。随着丢包率的增加，当 VC-TCP/NC 检测到随机丢包时，即 $VC=10$ 时，两者的传输速率将趋于一致。网络丢包的出现是随机的，在相同的丢包率下发送不同数量的数据包，3 种传输协议的传输速率的比值也会发生变化。总体而言，发生网络丢包时，VC-TCP/NC 协议比 TCP-Vegas 和 TCP/NC 的传输速率大。

3)为了更直观地在 VC-TCP/NC 中展现吞吐量与冗余因子 R 的变化情况，设置链路中的丢包率为 10%，按理论值将编码窗口的大小 W 设置为 3，并进行仿真，结果如图 8 所示。

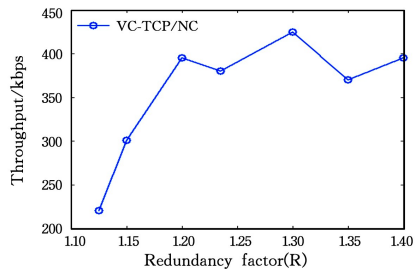


图 8 冗余因子与吞吐量的对比曲线

Fig. 8 Comparison of redundancy factor and throughput

由图 8 可知，网络吞吐量随着冗余因子上下波动，当冗余因子在 1.3 附近时，吞吐量达到最优值，这主要与选择的编码方案有关系，而且选择较好的 R 与编码窗口 W 值对编码操作也有较大好处。加入冗余因子的目的是使接收端的数据速率与 TCP 的发送端速率相匹配，如果冗余因子太小，则不会掩

盖 TCP 损失,且解码速率也会降低,这也是一个值得研究的方向。

4)为了研究吞吐量与解码窗口大小变化的关系,设置链路带宽为 2 Mbps,其数据包的长度为 1 kB,丢包率 γ 为 0.25,图 9 示出了网络吞吐量随着解码窗口大小变化的情况。

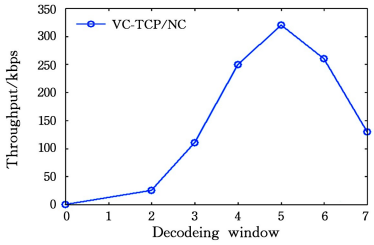


图 9　吞吐率随解码窗口大小变化的情况

Fig. 9　Change of throughput with decoding window size

由图 9 可知,当解码窗口发生改变时,吞吐量也会发生相应的变化。解码窗口小于 5 时,吞吐量与解码窗口是正向变化关系。解码窗口约为 5 时,吞吐量达到最大值。解码窗口大于 5 时,吞吐量与解码窗口是反向变化关系。另外需要说明,由于本文中 VC-TCP/NC 与 TCP/NC 采用了相同的编解码方式,因此只需分析 VC-TCP/NC 中吞吐率随着解码窗口变化而变化的情况。

5)模拟简单链路中由于网络拥塞造成丢包的场景,设置 *VC* 值为 01,仿真的网络结构图如图 10 所示。

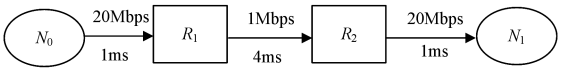


图 10　网络结构图

Fig. 10　Network structure

其中,N_1 和 N_2 表示节点,R_1 和 R_2 表示网络中的路由器。路由器使用 DropTail 队列管理方式,频宽为 1 Mbps,时延为 4 ms。因为 *cwnd* 直接反映 TCP 的传输效果,*cwnd* 越大,表示能发送的数据越多,所以这里将 *cwnd* 作为观察测量的参数,仿真结果如图 11 所示。

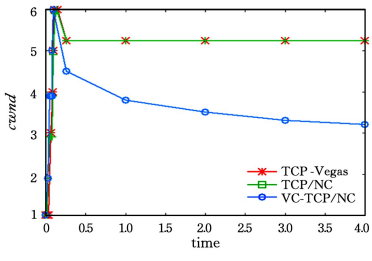


图 11　拥塞控制窗口与时间对比曲线

Fig. 11　Comparison of congestion control window and time

由图 11 可知,TCP/NC 比 VC-TCP/NC 和 TCP-Vegas 先达到拥塞窗口的最大值,这是由于 TCP/NC 对数据包进行了编码操作,在相同时间内发送了更多的编码数据包,此时 *VC*=01,VC-TCP/NC 未进行编码操作。随着时间的增加,由于编码窗口大小并没有改变,即进行编码的数据包没有减少,因此会导致 TCP/NC 的拥塞窗口急剧减小,而 TCP-Vegas 与 VC-TCP/NC 的拥塞窗口在达到最大值后也会减小,但是始终保持在一个稳定的值,且拥塞窗口的大小也要大于 TCP/NC。VC-TCP/NC 未进行编码操作,此时相当于一个常规的 TCP,因此 VC-TCP/NC 拥塞窗口的大小与 TCP-Vegas 的窗口大小接近。

4.3　动态场景下的结果与分析

本文在动态场景下对 TCP 与 VC-TCP/NC 协议进行了对比仿真实验。使用 VanetMobisim 搭建交通仿真场景,如图 12 所示。VanetMobisim 是一款交通仿真器,能够为 NS-2 生成移动跟踪 trace 文件,该场景在 1000 m×1000 m 的矩形区域设置了 60 个车辆节点,以 # 号作为区分。

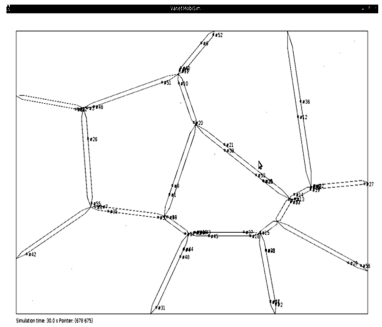


图 12　VanetMobisim 搭建的交通仿真场景

Fig. 12　Traffic simulation scenario built with VanetMobisim

使用了 6 种不同的平均移动速度速度:0 m/s,5 m/s,10 m/s,15 m/s,20 m/s,25 m/s,其中 0 m/s 相当于静止,5 m/s 相当于自行车的速度,10m/s,15 m/s,20 m/s,25 m/s 相当于机动车的移动速度。仿真时间设置为 500 s,参考节点之间有持续的 FTP 传输,节点的接收半径设为 250 m。网络仿真场景的基本参数如表 1 所列,分别在每种平均速度下进行 10 次网络仿真,计算每次仿真的端到端时延以及平均吞吐量。

表 1　参数表

Table 1　Parameters

参数	取值
物理层	WirelessPhyExt
MAC 层	802_11Ext
信道衰落模型	TwoRayGround
频段/GHz	5.9
带宽/MHz	10
通信范围/m	250
监听范围/m	550
分组大小/Byte	1000
路由层	DSR
应用层	FTP

设置 *VC* 值为 10,表示链路中发生随机丢包,并将丢包率设置为 0.1,实验结果如图 13 和图 14 所示。

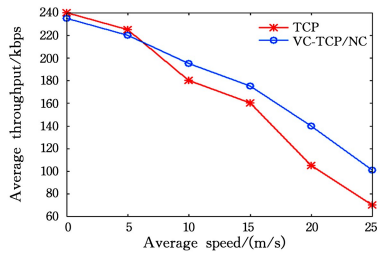


图 13　VC-TCP/NC 与 TCP 的吞吐量与速度变化的对比曲线

Fig. 13　Comparison of throughput and velocity changes for VC-TCP/NC and TCP

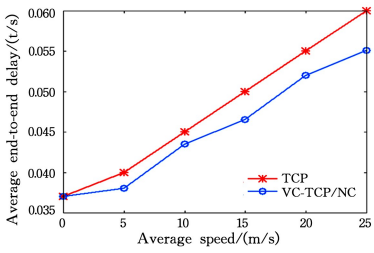


图 14 VC-TCP/NC 与 TCP 的端到端时延与速度对比曲线

Fig. 14 Comparison of end-to-end delay and velocity for VC-TCP/NC and TCP

由图 13 可以看出，TCP 和 VC-TCP/NC 的平均吞吐量都随着速度的增加而逐渐减小，这是因为移动速度增大导致路由中断次数增多，从而降低了链路通信质量。在节点平均速度小于 5 时，TCP 的吞吐量稍高于 VC-TCP/NC 的吞吐量。当节点速度增加时，可以明显看到，在相同的速度下，VC-TCP/NC 与 TCP 性能相比有显著提高，这是由于 VC-TCP/NC 使用了网络编码，减少了不必要的丢包重传以及不正确的调度拥塞处理机制的发生，从而没有迅速减小窗口，因此 VC-TCP/NC 的吞吐量相比于 TCP 有所提高。而从图 14 中 TCP 与 VC-TCP/NC 的端到端时延与速度的对比曲线可知，两种协议的端到端时延与节点的移动速度都为正向关系，由于 VC-TCP/NC 减少了不必要的重传，使得其平均端到端时延低于 TCP 且增加得较为缓慢，这也体现了 VC-TCP/NC 的优势。

结束语 本文针对无线车载网络的特点和要求，给出了一种基于随机线性网络编码的跨层联合优化方法 VC-TCP/NC，重新设计了网络编码层发送端的发送策略。理论分析和仿真结果表明，在丢包率高、信道质量不佳的环境下，VC-TCP/NC 在时延和网络吞吐量上的优势较为明显。为进一步克服解码延迟和反馈延迟，在今后的工作中，考虑将在线网络编码(TCP VON)和快速网络编码(TCP-FNC)引入到 VC-TCP/NC 中，以实现更低的解码时延，获得更高的吞吐量。

参 考 文 献

[1] KARAGIANNIS G, ALTINTAS O, EKICI E, et al. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions[J]. IEEE Communications Surveys & Tutorials, 2011, 13(4): 584-616.

[2] ZHENG K, ZHENG Q, CHATZIMISIOS P, et al. Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2377-2396.

[3] AL HANBALI A, ALTMAN E, NAIN P. A survey of TCP over ad hoc networks[J]. IEEE Communications Surveys & Tutorials, 2005, 7(3): 22-36.

[4] JAMIL F, JAVAID A, UMER T, et al. A comprehensive survey of network coding in vehicular ad-hoc networks[J]. Wireless Networks, 2017, 23(8): 2395-2414.

[5] SUNDARARAJAN J K, SHAH D, MEDARD M, et al. Network Coding Meets TCP: Theory and Implementation[J]. Proceedings of the IEEE, 2011, 99(3): 490-512.

[6] RUIZ H M, KIEFFER M, PESQUET-POPESCU B. TCP and Network Coding: Equilibrium and Dynamic Properties[J]. IEEE/ACM Transactions on Networking, 2016, 24(4): 1935-1947.

[7] FRAGOULI C, BOUDEC J Y L, WIDMER J. Network coding: an instant primer[J]. Acm Sigcomm Computer Communication Review, 2005, 36(1): 63-68.

[8] SUNDARARAJAN J K, JAKUBCZAK S, MEDARD M, et al. Network Coding meets TCP[C]// IEEE Conference on Computer Communications (INFOCOM). 2009: 2809-288.

[9] JUAN L, WEIMIN G, JUNKE W, et al. Enhanced Network Coding for TCP in Wireless Networks[C]// IEEE 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM). 2011: 1-4.

[10] HA N V, KUMAZOE K, TSUKAMOTO K, et al. Masking lossy networks by TCP tunnel with Network Coding[C]// IEEE Symposium on Computers & Communications (ISCC). 2017: 1290-1295.

[11] LIU H, GU Y. TCP with hop-oriented network coding in multi-radio multi-channel wireless mesh networks[J]. Networks Iet, 2012, 1(3): 171-180.

[12] BAO W, SHAH-MANSOURI V, WONG V W, et al. Tcp von: Joint congestion control and online network coding for wireless networks[C]// IEEE Global Communications Conference (Globecom). 2012: 125-130.

[13] LAN X, LI S, ZHANG S F. Network Coded TCP throughput Analysis in Wireless Mesh Networks[C]// IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks (MSN). 2013: 225-232.

[14] SUN J Y, ZHANG Y, TANG D, et al. TCP-FNC: A Novel TCP with Network Coding for Wireless Networks[C]// IEEE International Conference on Communications (ICC). 2015: 2078-2084.

[15] GARRIDO P, GOMEZ D, FITZEK F H P, et al. When TCP and Network Coding meet Wireless Links[C]// European Wireless 2016, European Wireless Conference. VDE, 2016.

[16] HA N V, KUMAZOE K, TSURU M. Making TCP / NC Adjustable to Time Varying Loss Rates[C]// IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS). 2016: 457-462.

[17] WANG Z R, XIE X Z, ZHAO D X. Key Technologies of Vehicle Self-Organizing Network [J]. Telecommunications Science, 2011, 27(1): 44-51. (in Chinese)
王昭然，谢显中，赵鼎新. 车载自组织网络关键技术[J]. 电信科学，2011，27(1)：44-51.

[18] LUO C, YU F R, JI H, et al. Cross-layer design for TCP performance improvement in Vehicular Communication Networks [C]// IEEE 14th International Conference on Advanced Communication Technology (ICACT). 2012: 400-405.

[19] FU C P, LIEW S C. TCP Veno: TCP enhancement for transmission over wireless access networks[J]. IEEE Journal on Selected Areas in Communications, 2003, 21(2): 216-228.