

面向船舶自组网的 Semi-TCP 研究

周 亮^{1,2}, 姜胜明¹, 熊晨霖³

(1. 上海海事大学 信息工程学院, 上海 201306; 2. 上海市教育考试院 网络信息中心, 上海 200433;

3. 华南理工大学 电子与信息学院, 广州 510641)

摘 要: 为提高传输控制协议(TCP)在船舶自组网中的性能, 分析暴露终端问题和高负载网络的拥塞控制问题, 通过开启定时器的方法对拥塞控制算法进行改进, 提出改进的半传输控制协议(Semi-TCP-RTS-V2), 并在 Exata 仿真平台上对 TCP-Lite、Semi-TCP-RTS、Semi-TCP-RTS-V2 进行性能分析, 结果表明, Semi-TCP-RTS-V2 在船舶自组网中传输性能有所提高。

关键词: 船舶自组网; 传输控制协议; 拥塞控制; 网络性能; 定时器

中文引用格式: 周 亮, 姜胜明, 熊晨霖. 面向船舶自组网的 Semi-TCP 研究[J]. 计算机工程, 2018, 44(2): 119-123.

英文引用格式: ZHOU Liang, JIANG Shengming, XIONG Chenlin. Research on Semi-TCP for Ship Ad-hoc Network[J]. Computer Engineering, 2018, 44(2): 119-123.

Research on Semi-TCP for Ship Ad-hoc Network

ZHOU Liang^{1,2}, JIANG Shengming¹, XIONG Chenlin³

(1. College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China;

2. Network and Information Center, Shanghai Municipal Educational Examinations Authority, Shanghai 200433, China;

3. College of Electronics and Information, South China University of Technology, Guangzhou 510641, China)

[Abstract] In order to improve the performance of transmission control protocol on ship ad-hoc network, the problem of congestion control and exposed terminal in high load network is analyzed, the congestion control algorithm is improved by starting a timer, the semi transmission control protocol named Semi-TCP-RTS-V2 is proposed, and the performance of TCP-Lite, Semi-TCP-RTS, Semi-TCP-RTS-V2 is analyzed in the Exata simulation platform, the results show that the transmission performance of Semi-TCP-RTS-V2 in the ship ad-hoc network is improved.

[Key words] ship ad-hoc network; Transmission Control Protocol (TCP); congestion control; network performance; timer

DOI:10.3969/j.issn.1000-3428.2018.02.021

0 概述

传输控制协议(Transmission Control Protocol, TCP)在陆地互联网上表现出良好的性能, 但将其应用于船舶自组网中, 其性能显著下降。传统 TCP 协议总是认为数据包的丢失是由于网络拥塞造成的, 而在船舶无线自组织网络中, 由于信道损耗、动态路由等问题, 很容易造成丢包, 而这种丢包并不是由于拥塞引起的。当 TCP 应用于无线自组织网络中时, 这种非拥塞的丢包就会造成 TCP 丢包即拥塞的误判, 而降低发送速率。最终造成: 非拥塞时信道资源的浪费; 拥塞时不利于快速解除拥塞。针对上述问题, 研究者提出了多种针对无线自组网的 TCP 改进方案^[1-2]。但大多数改进都是基于原 TCP 协议结构进行的, 其拥塞控

制^[3]功能仍在传输层实现^[4], 只是在特定的环境或应用中体现出某一方面的优越性, 而整体性能却并不令人满意, 如移动自组网传输协议(Ad-hoc Transport Protocol, ATP)^[5-6]、频率自适应传输控制协议(TCP with Adaptive Pacing, TCP-AP)^[7]等。

Semi-TCP 是对 TCP 协议体系的突破, 其将原传输层中的拥塞控制功能下放到数据链路层^[8], 通过逐跳控制, 能更准确地判断网络拥塞状况, 提高拥塞控制效率。船舶自组网有节点密度小、单跳距离远、信道损耗大、节点移动速度相对稳定等特点。本文结合船舶自组网的特点, 对基于数据请求发送/清除发送(Request To Send/Clear To Send, RTS/CTS)的拥塞控制方法进行改进, 从而提高 TCP 在船舶自组网中的传输性能。

基金项目: 国家自然科学基金“海洋互联网的开放式架构的基本理论研究”(61472237)。

作者简介: 周 亮(1985—), 男, 工程师、博士研究生, 主研方向为海洋互联网、无线自组网; 姜胜明, 教授、博士; 熊晨霖, 硕士。

收稿日期: 2016-11-08 **修回日期:** 2017-03-11 **E-mail:** lzhou@shmea.edu.cn

1 相关算法

1.1 基于 RTS/CTS 的 Semi-TCP 拥塞控制算法

根据船舶自组网的特点,船舶在水面上以一定的速度移动,信道损耗较大,将拥塞控制下放到 MAC 层是提高网络性能最简便且高效的方法。在 IEEE802.11 无线网络中,基于 DCF 接入控制模式下的 RTS/CTS 有着非常广泛的运用,下面就详细介绍基于 RTS/CTS 的 Semi-TCP 拥塞控制算法。

RTS/CTS 握手协议即请求发送/清除发送协议,是基于竞争的介质访问控制协议 CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance,带冲突避免的载波侦听多址接入)的一种媒体接入控制方式^[9],其采用 4 次握手 (RTS-CTS-DATA-ACK) 的方式^[10]。相比于基本的媒体接入控制方式 (2 次握手 DATA-ACK),它的提出是为了解决隐藏终端^[11]问题。隐藏终端是指在接收节点的覆盖范围内而在发送节点的覆盖范围外的节点,如图 1 所示。

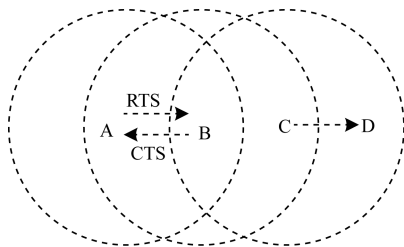


图 1 隐藏终端示例

图中节点 A 向 B 发送数据,节点 C 也有数据需要发往 D,图中所示情况下的 C 节点就是一个隐藏终端,若采用的是二次握手的接入控制方式,那么其在监听到信道空闲后向 D 发送数据,但是 B 同时处在 A 和 C 的接收范围内,因此会造成 B 在接收数据时发生碰撞。采用 RTS/CTS 之后,即使 C 收不到 A 的 RTS 帧,仍可以收到 B 的 CTS 帧然后进行退避,从而解决了这种情况。下面要讨论的 Semi-TCP 算法就是基于该握手协议。

1.2 算法基本思想介绍

基于 RTS/CTS 的 Semi-TCP 算法也分为节点内和节点间拥塞控制^[12],节点内拥塞控制原理这里不再赘述,下面就节点间的拥塞控制详细介绍。

根据 RTS/CTS 协议,假设节点 A 向节点 B 发送数据,在争取得到发送机会后发送数据之前需要向节点 B 发送 RTS 帧,节点 B 收到 RTS 帧后回复 CTS 帧。基于 RTS/CTS 的 Semi-TCP 拥塞控制算法中,利用 RTS 和 CTS 帧中的空闲位携带拥塞信息,以表明自己是否处于拥塞状态^[13-15]。从而引入了 2 种控制帧——带拥塞状态信息的请求发送帧 (Request-To-Send-with-Congestion-status, RTSC) 和带拥塞状态信息的允许发送帧 (Request-To-Send-with-Congestion-status, CTSC),同时增加了一种新的帧——拒绝清除发送帧 (negative-Clear-To-Send,

nCTS)。下面分别分析 3 种帧的作用,RTSC:发送节点 A 向节点 B 发送数据,若 A 处于拥塞,则发送 RTSC,不拥塞时则发送 RTS:

1) nCTS:接收节点 B 在收到 RTS 帧之后,判断自身拥塞状态,若拥塞则回复 nCTS 帧,拒绝接收数据,否则回复 CTS。若节点 A 收到 nCTS,则不能向 B 发送数据包。值得注意的是接收节点为目的节点的情况 (最后一跳),此时对于拥塞的判断不再是对于发送缓存进行判断,而是接收缓存。

2) CTSC:CTSC 帧的引入主要是为了解决死锁的情况,死锁就是两拥塞节点 A 和 B 需向对方发送数据来解除本身的拥塞,但这 2 个节点又处于相互拒绝对方的状态中。

3) 若节点 A 和节点 B 都处于拥塞状态,A 和 B 发送缓存中队首数据都是发送给对方。发送节点 A 会先发送 RTSC 帧,节点 B 会回复 nCTS 拒绝接收数据。由于节点 B 的拒绝,节点 A 和 B 将处于僵死的状态,直到超过重传上限丢掉数据包。因此,当节点 B 在收到 RTSC 之后,若判断出可能出现死锁,就回复 CTSC 帧互相帮助解除拥塞。

利用上述 3 种新的帧,当网络中某一节点拥塞时,就可以快速的向上游节点传递拥塞信息,各上游节点进行逐跳的拥塞控制,最终将拥塞传到源节点,从源头减少发往网络中的数据,以实现更好的网络通信性能。

2 算法性能改进措施

根据在 1.1 节中对 RTS/CTS 握手协议的介绍,RTS、CTS 帧会在网络分配矢量 (NAV) 域 (具体值为控制帧中 Duration 位) 中预约了时隙,以表明收到该帧的节点在多长时间以内需要退避,从而避免发送数据帧时产生碰撞。RTS/CTS 协议是为了解决隐藏终端而提出的,但是暴露终端问题也一定程度上影响网络性能,暴露终端是指在发送节点的覆盖范围内而在接收节点的覆盖范围外的节点,如图 2 所示。

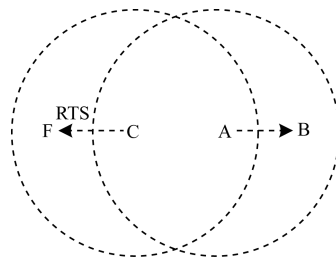


图 2 暴露终端示例

节点 C 在向节点 F 发送数据,A 也需要向 B 发送数据,图示情况中的 A 节点就是一个暴露终端。A 在收到 C 的 RTS 后会进行退避,然而实际上各自的接收节点并不在对方的发送节点的接收范围内,因此,2 种数据并不会在接收端发生碰撞,A 是可以

向 B 发送数据的。这样的暴露终端问题就造成了信道的浪费。

不仅暴露终端问题,对于一个数据流较多、负载较重的拥塞网络而言,自然会有多个 nCTS 帧的交互,这也会带来很大的问题,假设有节点 A 需要发送数据帧给节点 B, B 回复 nCTS, 那么节点 A 在收到之后将不会在原本预约好的时隙中发送数据。如图 3 所示,此时周围的邻居节点可以分情况考虑。

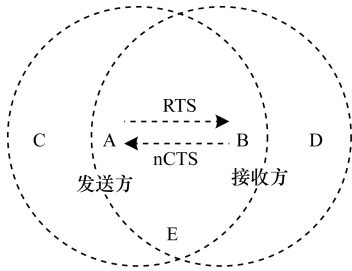


图 3 nCTS 的交互

节点 D 作为隐藏终端,当收到 nCTS 之后,根据其 Duration 会进行相应的退避;节点 C 作为暴露终端,收到 RTS 后也会进行退避;节点 E 能收到 2 个控制帧,会选择 RTS 和 nCTS 更大的 Duration 进行退避。至此节点 A ~ 节点 E 都将不能发送数据,从而造成信道的浪费,使得网络拥塞加剧。

为了解决上述 2 种问题以优化 Semi-TCP 的性能,进行如下的算法设计:

1) 将 nCTS 的 Duration 设为 0, 解决图 2 中隐藏终端的退避问题。

2) 节点在收到不是发给自己的 RTS 帧之后,会记录下该帧的发送地址,同时开启一个定时器暴露终端判定定时器 (Timer for Exposed Terminal, TET), 若在一定时间间隔内没有收到发往该地址的 CTS 或者重传的 RTS, 则判定自己为暴露终端, 可以取消退避。

3) 若在 TET 定时器超时之前收到了 nCTS, 则取消退避。

4) 节点在收到不是发给自己的 RTS 帧之后,在定时器 TET 超时前: 收到别的节点发来的不是给自己的 RTS 帧, 则记录下新的地址并且重设定定时器; 收到不是发给自己的 CTS 帧无条件按照 Duration 进行退避。

5) 节点在 TET 定时器开启的状态下若收到任何发给自己的控制帧, 都按照退避状态处理。

关于设置的 TET 定时器的时长, 必然需要考虑到重传 RTS 的情况, 设置满足要求的最小值。结合 RTS/CTS 协议进行分析, 节点在发送了 RTS 帧之后, 会开启一个重传定时器, 该重传定时器的值如式 (1) 所示, 其中, $PropDelay$ 为传播时延, $SIFS$ 为短帧间间隔, $transmitDelay(C_{CTS})$ 为 C_{CTS} 帧的传输时延, $slotTime$ 为等待的随机时间。

$$holdForCts = PropDelay + SIFS + transmitDelay(C_{CTS}) + PropDelay + slotTime \quad (1)$$

即在这段时间之后, 若在该规定时间内没有收到对方任何回应, 则会重传 RTS 帧。因此, 一个节点在收到不是发给自己的 RTS 帧之后, 如果连重传的 RTS 都没有, 那么判定自身为暴露终端。TET 定时器时长如式 (2), $transmitDelay(R_{RTS})$ 为 RTS 的传输时延。

$$holdForBackup = (holdForCts - PropDelay) + transmitDelay(R_{RTS}) + PropDelay \quad (2)$$

在 TET 定时器没开的状态下, 收到不是发给自己的控制帧的算法流程如图 4 所示。

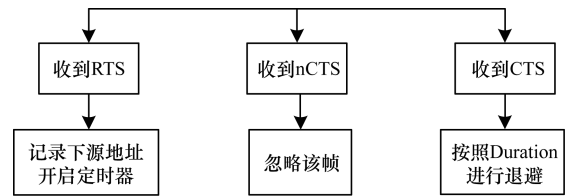


图 4 TET 开启状态下收到非自己的控制帧处理流程

在 TET 定时器开启的状态下, 收到不是发给自己的控制帧的算法流程如图 5 所示。

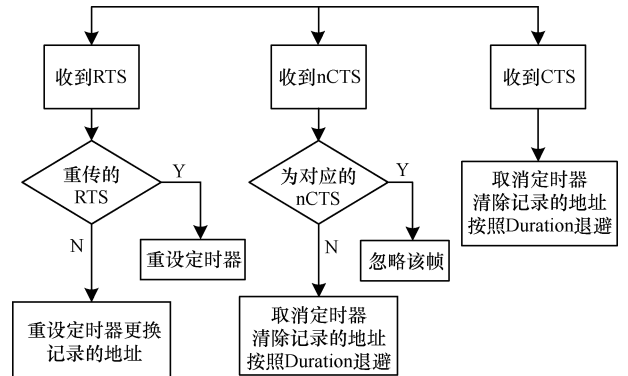


图 5 TET 关闭状态下收到非自己的控制帧处理流程

3 实验仿真及分析

吞吐量和时延作为网络性能的重要评判参数, 在对 Semi-TCP 进行的性能测试中, 本文中将其作为主要的评判标准。网络平均吞吐量直观的表现了网络的传输性能, 是考察网络性能的重要参数, 提高吞吐量也是网络优化协议的最主要的目的。分组时延是指分组从发送节点发出, 到目的节点收到分组时的这段时间间隔, 它包含了队列时延、传输时延和传播时延。在采用的 Exata 仿真软件的统计中, 以上参数都会直接给出方便观察比较。

3.1 仿真场景及参数设置

本节只针对船舶自组网进行仿真, 为了能体现出船舶自组网的特点, 有必要建立一个远距离的动态的仿真场景, 海上的部分节点按一定速度运动, 并配合适当的信道衰减模型, 同时, 要对拥塞控制算法

进行测试,必然需要建立拥塞的场景,因此,加上了多个应用数据流,从而尽可能体现出网络吞吐量、丢包率、平均分组时延的差异。仿真场景抽象场景如图 6 所示。

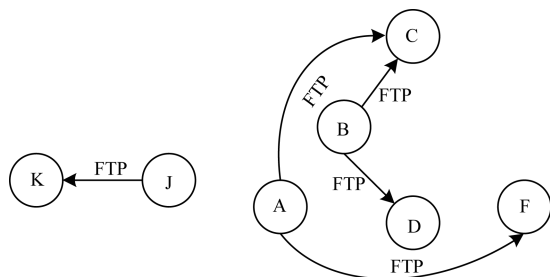


图 6 船舶自组网仿真场景

场景中各个节点的发送范围仅覆盖与其相邻的节点,节点 A 发往节点 C 和节点 F 的数据流都需要经过中间节点的转发。节点 J 到 K 的数据流主要是为了验证 A 作为暴露终端的情况下的 Semi-TCP 改进算法的性能,B 的 2 个数据流主要是为了造成网络的拥塞。场景的基本参数设置如表 1 所示。

表 1 仿真场景主要参数

信道模型	TwoRayGround
衰减模型	Ricean, $K = 5$
无线频段/GB	2.4
MAC 速率/(Mb · s ⁻¹)	11
发送队列	FIFO, 容量 150 KB
路由协议	AODV

为了方便叙述,本文将基于 RTS/CTS 的 Semi-TCP 拥塞控制算法称为 Semi-TCP-RTS,改进算法称为 Semi-TCP-RTS-V2,2 种算法的仿真结果与软件提供的一个 TCP 的轻量版本 TCP-Lite 进行比较。

3.2 仿真结果分析

为了分析门限值对于算法性能的影响,将其作为变量进行多次仿真,同时鉴于有多个数据流,统计了各个数据流平均吞吐量的总和并进行了比较,结果如图 7 所示。

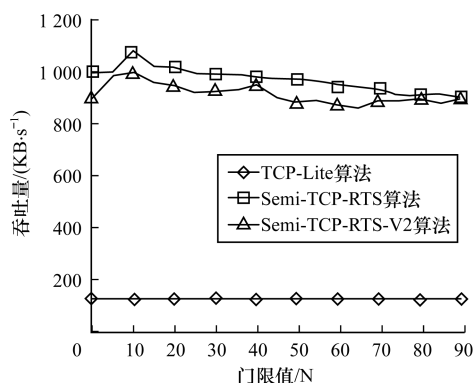


图 7 拥塞场景下吞吐量随门限值的变化

由该结果可以看出,2 种 Semi-TCP 算法相较于 TCP-Lite,在网络总吞吐量上都有大幅的提升,这是因为 TCP-Lite 将由于信道质量不足造成的丢包错判为网络拥塞,错误地降低了发送窗口,限制了网络性能。同时 Semi-TCP-RTS-V2 相较于 Semi-TCP-RTS,总吞吐量上稍有下降,针对该结果结合图 8 中对发送节点 A 的负载的对比做出解释。

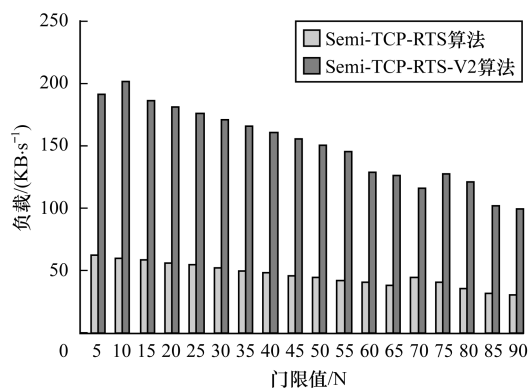


图 8 发送节点 A 负载随门限值的变化

结合图 8 可以发现,节点 A 的负载相较于整个网络的平均吞吐量之和小得多,实际上在 Semi-TCP-RTS 测试结果中,甚至存在节点 A 的一个数据流建立失败的情况。这是因为节点 A 发送数据会同时受到节点 J 和节点 B 的影响,不仅充当了 J → K 以及 B → C 数据流的暴露终端,同时在 B → D 数据流中,由于 D 在其接收范围内也会影响其发包。再加上节点 B 的两个数据流造成的拥塞,更加限制了 A 发送数据包。因此也就出现了上述结果。

另外还可以明显发现,在改进算法中,节点 A 在整个网络吞吐量下降的情况下,自身的负载反而上升。这说明节点 A 与另外 2 个发送节点之间的差距缩小了,也就是说在数据流的公平性上有所改善。这是因为 Semi-TCP-RTS-V2 中,暴露终端的问题得到改善,因此负载也有所提升。至于图 7 中显示的总吞吐量相较于 Semi-TCP-RTS 偏低,是由于改进算法中对于暴露终端的判断需要耗费一些时间,同时节点 A 在发包时会使得另外两个发送节点都进入暴露终端的判断阶段,因此才产生这样的结果。

继续观察图 7,可以发现门限值的变化对于网络整体的吞吐量并没有造成很大的影响,但是随着门限值的增加,队列的平均长度必然也会增加,则平均队列时延一定会相应的变长,下面针对 J → K 数据流中,节点 J 的平均队列时延进行数据统计,结果如图 9 所示,由于门限值一致,Semi-TCP-RTS 和 Semi-TCP-RTS-V2 并没有什么差异,整体上升的趋势也符合预期。

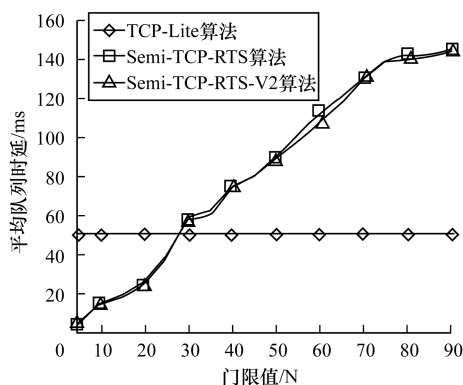


图9 节点J的平均队列时延随门限值的变化

4 结束语

船舶自组网采用 IEEE802.11 协议标准,本文介绍现有的 Semi-TCP-RTS 拥塞控制算法,同时提出针对 Semi-TCP-RTS 的改进算法 Semi-TCP-RTS-2,以解决暴露终端以及由于 nCTS 的交互带来的信道资源浪费等问题。在 Exata 仿真平台分别实现和测试了3种算法,并进行了对比分析。结果显示,Semi-TCP-RTS-V2 算法优于 TCP-Lite 和 Semi-TCP-RTS,同时在节点发送数据的公平性上有所改善。

参考文献

- [1] AFANASYEV A, TILLEY N, REIHER P, et al. Host-to-host Congestion Control for TCP [J]. IEEE Communications Surveys & Tutorials, 2010, 12(3): 304-342.
- [2] HANBALI A A, ALTMAN E, NAIN P. A Survey of TCP over Ad Hoc Networks [J]. IEEE Communications Surveys & Tutorials, 2005, 7(3): 23-36.
- [3] SARDAR B, SAHA D. Survey of TCP Enhancements for Last-hop Wireless Networks [J]. IEEE Communications Surveys and Tutorials, 2006, 8(3): 20-34.

- [4] CHRISTIAN L, BJORN S, MARTIN M. A Survey on Congestion Control on Mobile Ad Hoc Networks [J]. Wiley Wireless Communications and Mobile Computing, 2007, 7(5): 655-676.
- [5] FU Zhenghua, GREENSTEIN B, MENG Xiaoqiao. Design and Implementation of a TCP-friendly Transport Protocol for Ad Hoc Wireless Networks [C]//Proceedings of ICNP'02. Paris, France: [s. n.], 2002: 216-225.
- [6] MONZER H A M, SUHAIDI H. Loss Detection and Recovery Techniques for TCP in Mobile Ad Hoc Network [C]//Proceedings of the 2nd International Conference on Network Applications, Protocols and Services. Washington D. C., USA: IEEE Press, 2010: 48-54.
- [7] 蔡业桂. 多跳无线网络基于 RTS/CTS 的 Semi-TCP 拥塞控制算法研究 [D]. 广州: 华南理工大学, 2010.
- [8] 刘梅, 姜胜明, 陆以勤. 无线多跳网络中 Semi-TCP 协议的实现 [J]. 计算机工程, 2012, 38(9): 79-82.
- [9] 张伟. 无线移动 Ad Hoc 网络违规行为分析 [J]. 中国科技纵横, 2012, 13: 62-64.
- [10] 虞万荣. 无线自组网 MAC 协议关键技术研究 [D]. 长沙: 国防科学技术大学, 2006.
- [11] 刘志敏, 徐凤英. 移动 Ad Hoc 网络隐藏和暴露终端问题的研究 [J]. 计算机与网络, 2010, 6(1): 30-37.
- [12] LIU J, SINGH S. ATCP: TCP for Mobile Ad Hoc Networks [J]. IEEE Journals on Selected Areas in Communications, 2001, 19(7): 1300-1315.
- [13] SUNDARESAN K, ANANTHARAMAN V, HSIEH H Y, et al. ATP: A Reliable Transport Protocol for Ad Hoc Networks [C]//Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing. New York, USA: ACM Press, 2003: 64-75.
- [14] 毛华超. 多跳无线网络服务质量与拥塞控制的联合设计 [D]. 广州: 华南理工大学, 2013.
- [15] TASSIULAS L. Adaptive Back-pressure Congestion Control Based on Local Information [J]. IEEE Transactions on Automatic Control, 1995, 40(2): 236-250.

编辑 刘冰

(上接第118页)

- [9] JANGRA A, SHALINI, GOEL N. e-ARAN: Enhanced Authenticated Routing for Ad Hoc Networks to Handle Selfish Nodes [C]//Proceedings of International Conference on Advances in Engineering, Science and Management. Washington D. C., USA: IEEE Press, 2012: 144-149.
- [10] JHAVERI R H, PATEL N M. A Sequence Number Based Bait Detection Scheme to Thwart Grayhole Attack in Mobile Ad Hoc Networks [J]. Wireless Networks, 2015, 21(8): 1-18.
- [11] XUE Y, REN H L. Greenshields Model Based Congestion Pricing in Bottleneck Road Section [J]. Shandong Science, 2013, 26(5): 84-90.
- [12] PARAMESWARY S, ROSLI H M, CHE P O, et al. Influence of Darkness Dry and Darkness Rainfall on

Malaysian Expressway for Traffic Characteristics Using Greenshield's Model [J]. Journal of Basic and Applied Scientific Research, 2015, 5(11): 34-42.

- [13] DUA A, KUMAR N, BAWA S. A Systematic Review on Routing Protocols for Vehicular Ad Hoc Networks [J]. Vehicular Communications, 2014, 1(1): 33-52.
- [14] VERMA K, HAASBULLAH H, KUMAR A. An Efficient Defense Method Against UDP Spoofed Flooding Traffic of Denial of Service (DoS) Attacks in VANET [C]//Proceedings of IACC'13. Washington D. C., USA: IEEE Press, 2013: 550-555.
- [15] 王海龙, 杨岳湘. 基于信息熵的大规模网络流量异常检测 [J]. 计算机工程, 2007, 33(18): 130-133.

编辑 索书志