

移动 Ad Hoc 网络的跨层优化拥塞控制*

徐伟强^{1,2}, 汪亚明¹, 俞成海¹⁺, 刘良桂¹, 张云华¹

¹(浙江理工大学 信息电子学院, 浙江 杭州 310018)

²(浙江大学 工业控制技术国家重点实验室, 浙江 杭州 310027)

Cross-Layer Optimal Congestion Control Scheme in Mobile Ad Hoc Networks

XU Wei-Qiang^{1,2}, WANG Ya-Ming¹, YU Cheng-Hai¹⁺, LIU Liang-Gui¹, ZHANG Yun-Hua¹

¹(College of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

²(State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: E-mail: yuchenghai@zstu.edu.cn

Xu WQ, Wang YM, Yu CH, Liu LG, Zhang YH. Cross-Layer optimal congestion control scheme in mobile ad hoc networks. *Journal of Software*, 2010,21(7):1667–1678. <http://www.jos.org.cn/1000-9825/3803.htm>

Abstract: This paper identifies all factors impairing TCP performance based on network protocol hierarchy: Lossy wireless channel at the physical layer; excessive contention and unfair access at the MAC layer; frequent routing changes due to node mobility at the network layer; the fundamentally inappropriate mechanisms of TCP at the transport layer, including window-based congestion control, loss-based congestion detection, slow-start and AIMD (additive increase/multiplicative-decrease) of congestion window; reliance on ACK-clocked characteristics. Then, it designs a novel cross-layer optimal congestion control (CCOC) protocol tailored toward the characteristics of ad hoc networks. Cross-Layer design framework is applied in CCOC to improve fair access at MAC layer, to detect false link failure, to reduce the number of route failures, to quick-start during route changes, to transmit reliably based on SACK, and to implement the adaptive optimization strategy guided by the nonlinear optimization theory. Then, this paper outlines the protocol framework of CCOC. Finally, the extensive packet-level simulations is implemented in NS2 environment. The simulation results show that CCOC significantly outperforms TCP and ATCP in many important performances such as throughput and fairness, under a variety of scenario and mobility conditions.

Key words: ad hoc network; congestion control; resource allocation; cross-layer design

摘要: 基于网络协议层框架充分分析了 TCP 应用到 Ad Hoc 网络导致性能下降的原因,包括:物理层中易损耗的无线信道,MAC 层中的过度竞争和不公平接入,网络层中节点移动引起的频繁路由失效,传输层中 TCP 采用的不合适机制,包括基于窗口的传输、基于数据包丢失的拥塞指示,拥塞窗口的慢启动和 AIMD、对 ACK 自定时的依赖;提出了全新的适合 Ad Hoc 网络特性的跨层优化拥塞控制协议 CCOC(cross-layer optimal congestion control).CCOC

* Supported by the National Natural Science Foundation of China under Grant Nos.60702081, 60873020 (国家自然科学基金); the China Postdoctoral Science Foundation of China under Grant No.20080440201 (中国博士后科学基金); the Zhejiang Provincial Natural Science Foundation of China under Grant Nos.Z1080702, Y107309, Y1090980 (浙江省自然科学基金)

Received 2009-09-01; Accepted 2009-12-02

运用了跨层设计框架来改进 MAC 层的接入信道公平性、检测虚假链路失效、减少路由失效次数、加快路由切换后的重启动、运用 SACK 实现可靠传输和实施非线性优化论指导设计的自适应优化策略;描述了 CCOC 的协议体系结构;实施了详细的 NS 仿真实验,仿真结果表明,在几乎所有的仿真场景和移动环境下,CCOC 比 TCP 和 ATCP 在许多重要性能指标,如吞吐量和公平性方面都有了明显的改进。

关键词: ad hoc 网络;拥塞控制;资源分配;跨层设计

中图法分类号: TP393 文献标识码: A

Ad Hoc 网络是指一组带有无线收发装置的移动节点组成的一个多跳的临时性自组织的网络系统.自组织和移动的特点使得 Ad Hoc 网络具有很高的生存能力和灵活性,可以广泛应用于敌对和不易建设固定通信设施的环境中,如野战通信、紧急搜救、临时会议等.传输控制协议 TCP 是专门设计用于在不可靠的 IP 层上为应用层提供了一个可靠的、按序传输、端到端的数据包传输服务.据统计,总字节数的 95%和总数据包数的 90%使用 TCP 传输.TCP 拥塞控制算法已经成为保证互联网稳定性的重要因素.TCP 具有多个版本,本文主要针对当前广泛应用的 Reno 版本.

TCP 最初是针对有线网络设计的,由于有线链路的可靠性高,传统 TCP 认为丢包是由网络拥塞所导致.然而,无线网络中的误码率比有线网络要高好几个数量级.并且在 Ad Hoc 网络中,信道接入的不公平性和节点移动等因素都会导致非拥塞的丢包,节点移动导致的频繁路由切换又会导致乱序.当将传统 TCP 应用到 Ad Hoc 网络时,会把非拥塞因素导致的乱序和包丢失错误地当作是网络拥塞的标志,不必要地启动了拥塞控制,从而导致严重的性能问题^[1,2].对传统 TCP 在 Ad Hoc 网络中的改进,是近年来网络研究的热点之一,研究者们提出了众多改进方案.这些改进方案可以分成 3 类:基于网络的改进机制、基于端节点的改进机制和专用的全新方案.

(1) 基于网络的改进机制的特点是需要中间节点协作,由中间节点负责监控网络状况,并反馈给端节点.端节点以此区分拥塞、路由失效乃至链路错误丢包等网络情况,以便根据网络状况采取适当的措施.这类方案的状态检测准确率比较高,但其准确程度依赖于反馈信息传送的可靠性和速度.典型的改进方案有 TCP-F^[3], TCP-ELFN^[4], TCP-BuS^[5], ATCP^[6].

(2) 改善 Ad Hoc 网络 TCP 性能的另一条途径是对现有的端到端拥塞控制算法进行改进.与基于网络反馈的机制不同,这种方案无需中间节点参与,通过端到端对某些度量值的检测来推测当前通路的状态,并依据各个状态采取相应措施.典型的改进方案有 Fix RTO^[7]和 DOOR^[8].此外,还有确定拥塞窗口的改进方案和控制 ACK 发送节拍的改进方案^[9],也可以归到这一类.

(3) 由于以上这两类改进方案只能取得有限的性能改善,学术界于是提出了专门为 Ad Hoc 网络设计的全新方案,典型的方案有 ATP^[10], WXCP^[11], C³TCP^[12], IHCC^[13], Hop^[14], DiffQ^[15], WCP 和 WCPcap^[16].ATP 通过中间节点的平均队列时延和平均传输时延所指示的当前网络所允许的速率直接反馈给源端来调整发送速率. C³TCP 通过收集链路层的带宽和延时信息来调整源端的信息流输出速率. WXCP 通过可用剩余带宽、接口队列长度和平均链路层重传次数这 3 种参数来估计网络的拥塞程度,提高了拥塞估计精度. IHCC 是一种隐式逐跳拥塞控制协议,它采用隐式背压(backpressure)机制和轻量级的纠错检错机制来快速响应不断变化的无线媒质. Hop 是全新的无线传输协议,其核心理念是采用了可靠的每跳 block 传输机制. DiffQ 设计了基于差分积压的 MAC 调度和路由器辅助的背压机制,以优化多跳无线网络下的拥塞控制性能. WCP 是一种基于 AIMD 的速率控制协议,以实现多跳无线网络资源公平和有效的分配.而 WCPcap 是基于容量估计的无线控制协议,它通过估计每个邻居可用的信道容量和竞争流之间的信道容量分配来优化设计分布式速率控制器.

此外,作为能量受限的 Ad Hoc 网络——无线传感网络的拥塞控制也受到学术界的关注^[17],提出了 FACC^[18], E RTP^[19]和 UAFC^[20]等协议. FACC 是一种基于速率公平感知拥塞控制协议.在 FACC 中,近源节点保持每流状态,并近似公平分配经过该节点的每个流;近目标节点使用轻量级的基于队列占用和访问次数的概率丢弃算法. E RTP 是专为流媒体应用数据设计的能源有效和可靠的无线传感器网络传输协议. E RTP 基于统计可靠性指标,以确保传输给目标节点的数据包数量超过预定的阈值.我们在文献[20]中研究了基于效用优化的能量

有效的异步拥塞控制机制 UAFC(utility-based asynchronous flow control algorithm).

然而,这些改进方案或多或少地都存在如下问题:(1) 未能充分分析传统 TCP 应用到 Ad Hoc 网络导致性能下降的原因所在,因而没有能够“对症下药”,提出合理、高效的改进措施;(2) 缺乏有效的理论指导;(3) 未能充分考虑 Ad Hoc 网络存在着无线多跳连接、节点变动这两个本质特点.本文在充分分析性能下降的原因之后,综合运用“跨层设计”、“优化理论”和“滚动优化”等手段,提出了适合 Ad Hoc 网络的跨层优化拥塞控制协议 CCOC(cross-layer optimal congestion control).

1 TCP 在 Ad Hoc 网络中性能下降的原因分析

传统 TCP 拥塞控制机制采用了如下 4 种机制:

慢启动机制.当拥塞窗口小于慢启动门限窗口时,采用慢启动机制,用于探测网络的可用带宽,每个数据包被确认(ACK)后,拥塞窗口就加 1.

拥塞避免机制.当拥塞窗口大于门限窗口 $ssthresh$ 时,采用拥塞避免机制,避免可能发生的拥塞,并尽可能地探测可用带宽,使用加性增加、乘性减小(AIMD)的方式来改变拥塞窗口的大小.

快速重传快速恢复机制.当源端收到重复 ACK(DupACK)时,采用快速重传机制重发 DupACK 指示的数据包并快速恢复机制对拥塞窗口和 $ssthresh$ 重新赋值,避免进入慢启动阶段.

超时重传机制.在源端发送数据包的同时,启动重传定时器(retransmission timer).如果源端在定时器超时之前还没有收到 ACK,则认为该数据段已经丢失并将进行重传,使用指数 RTO(retransmission timeout)退避算法,并进入慢启动阶段.

由此可见,TCP 拥塞控制机制表现出以下几个特点:(1) 基于窗口的传输;(2) 用推测数据包的丢失来指示网络拥塞;(3) 源端数据的发送要依赖于接收端反馈的 ACK.然而,传统 TCP 应用到 Ad Hoc 网络中是不合适的,主要体现在:

- 高误码率. Ad Hoc 网络应用的无线链路是开放的有损介质,存在着不可预见的外界干扰、多径衰落、阴影效应,加之许多 Ad Hoc 网络的信道选用开放频段,使得 Ad Hoc 网络具有比特误码率高的特点.高误码率容易使数据包传输受损,导致经常性的数据包丢失.如果 TCP 源端在 RTO 超时前没有收到 ACK 数据包,就会引发 TCP 的慢启动,从而源端的 RTO 时间加倍,并减小当前拥塞窗口为 1 个数据包,使得源端发送数据包的速率产生了突降.频繁的错误会使拥塞窗口一直保持在很小的范围内,从而造成吞吐量的降低.

- 信道接入的不公平性. Ad Hoc 网络的共享信道为多跳共享广播信道,由此带来的直接影响就是数据包冲突与节点所处的位置相关,使发送节点和接收节点感知到的信道状况不一定相同,引发了隐终端、暴露终端等一系列的特殊问题.IEEE 802.11 DCF 是 Ad Hoc 网络 MAC 层的事实标准,它采用了二进制指数退避机制:对成功发送后的节点引入较小的竞争窗口,使其处于竞争优先地位,而竞争中失败的节点引入较大的竞争窗口,使其处于更不利的地位,从而导致信道接入的不公平性.隐终端、暴露终端问题又加剧了这种不公平性.严重情况下,某些节点长期占用无线信道,而另外一些节点长期竞争不到信道.MAC 协议把因冲突造成的信道接入失败看作是链路中断,并向上层的路由协议报告;每当路由协议接收到这一通告,就会认为通过该链路的路由已经失效,继而采取一系列的路由恢复措施.而在路由再次恢复之前,所有接收到的要使用该失效路由的数据包均被立即丢弃.

- 节点移动. Ad Hoc 网络中的节点移动会使得网络节点之间的连通性发生变化,导致路由的失效.频繁的路由失效会导致 TCP 采取不必要的拥塞控制措施,可能会出现如下情况:

(1) 路由的失效导致中间节点丢弃数据包,这种行为会被 TCP 源端理解为拥塞发生;

(2) 路由失效后,网络层的路由算法就会去重新发现新路由.发现新路由的时间可能要比 TCP 协议中 RTO 时间要长,源端会进入慢启动过程.如果中断时间较长,则会导致 TCP 源端采用指数退让策略来推迟数据包的重发,重传的时间间隔呈指数上升.经过多次退避后,重传时间会变得很长,这样很可能在路由恢复时,源端不能及时恢复数据的传送.即使很快发现一条新路由,采用慢启动机制去探测网络的可用带宽,吞吐量仍然不会很高;

(3) 频繁的路由失败和重建将使得多数从源端发送出来的顺序数据包按照不同的路由进行转发,致使在到达接收端时产生了乱序^[10].乱序会使 TCP 源端采取不必要的拥塞控制机制,将慢启动门限设置为当前拥塞窗口的一半,并启动快速重传和快速恢复算法.

• 不对称. TCP 是依赖 ACK 自定时的基于窗口协议,即 ACK 的到达触发下一个 TCP 数据包的发送.如果 ACK 是瓶颈,则会引起 ACK 包在中间节点的累积.这些累积的 ACK 连续到达 TCP 源端,将引发源端连续发送一连串的数据包,导致严重的数据包“突发性”的传输,产生 ACK 累积效应^[21],网络出现拥塞的可能性大大提高.

从上面的分析可知,TCP 拥塞控制机制在 Ad Hoc 网络性能严重下降的根源在于:

(1) 基于数据包丢失的拥塞指示.在 Ad Hoc 网络中,引起数据包丢失的因素除了拥塞之外,还有无线链路的高误码率、信道的不公平接入、节点移动引起的“路由失效”.TCP 不能区分数据包丢失的原因,只是简单地频繁唤起不必要的拥塞控制,吞吐量以及链路的利用率等都受到影响,直接导致网络性能的显著下降.因此,以数据包丢失作为网络拥塞的指示在 Ad Hoc 网络中是不合适的;

(2) 慢启动和 AIMD.对 Ad Hoc 网络来说,慢启动机制导致了网络资源利用率的低下.它要经过几个 RTT(round-trip time)的时间才能使连接充分利用可用的信道带宽.由于 Ad Hoc 网络的动态特性,TCP 连接频繁中断,反过来导致频繁的超时,使 TCP 处于慢启动阶段的时间偏长.拥塞避免阶段采用的“加性增加”AI(additive increase)存在着与慢启动机制相同的缺点.采用的“乘性减小”机制也不适合 Ad Hoc 网络.大部分的数据包丢失是“路由失效”的结果.当路由改变时,源端根据既往网络状况计算出的调节参数对于当前网络状况而言已经陈旧而不再适合,从而加剧了网络的拥塞和不稳定.因此,当路由改变时,传输层应立即进入带宽估计阶段,不必再在老拥塞窗口作“乘性减小”的修改;

(3) 基于窗口的传输.文献[22]研究表明,TCP 的优化拥塞窗口通常小于 5 甚至小于 1.这样,在拥塞窗口的任何变化都会导致在每个 RTT 内的吞吐量发生很大的抖动.文献[11]也表明,传统 TCP 的拥塞窗口往往趋于一个较大的值,拥塞窗口控制算法在 Ad Hoc 网络中显得过于激进,这会严重加剧 Ad Hoc 网络中的拥塞;

(4) 对 ACK 的依赖.在 Ad Hoc 网络中,ACK 流在反向路径上与前向路径上的数据包流面临着信道接入的竞争,从而减少了数据包的平均传输速率.同时,容易导致“突发性”数据包传输.

2 跨层优化拥塞控制方案 COCC

由上节可知,TCP 拥塞控制机制在 Ad Hoc 网络性能严重下降的根源有两类:

一类是 Ad Hoc 网络的特点所致,包括:(1) 无线链路的高误码率导致数据包丢失;(2) 信道的不公平接入导致数据包丢失和虚假链路失效;(3) 节点移动引起的“路由失效”而导致的数据包丢失、乱序和频繁的路由切换;(4) 不对称路径导致的 ACK 累积.

另外一类是 TCP 拥塞控制机制的本身特性所致,包括:(5) 基于数据包丢失的拥塞指示;(6) 慢启动;(7) AIMD;(8) 基于窗口的传输;(9) 对 ACK 的依赖.

本文和我们先前的工作^[23,24]综合运用“优化理论”、“滚动优化”和“跨层设计”等手段,设计了适合 Ad Hoc 网络特点的拥塞控制机制 CCOC,以解决 TCP 在 Ad Hoc 网络存在的缺陷.图 1 用框图形象地描述了 CCOC 的设计框架.

在文献[23]中,基于对偶分解优化方法设计了公平、有效的价格协作拥塞控制算法 PCA(price cooperation approach),以针对导致性能下降的根源(8).同时,该算法是基于显式拥塞指示的,以针对根源(1)、根源(2)和根源(3)中数据包丢失导致的性能下降,也克服了根源(5)的缺陷.而且,该算法不像 AIMD 那样仅凭工程经验,而是通过优化理论的严格指导而设计的,这主要针对根源(7).

文献[23,24]针对 Ad Hoc 网络中节点频繁移动导致网络状态不断变化带来的挑战,设计了基于状态检测和滚动优化的自适应拥塞控制策略 AOS(adaptive optimization strategy).

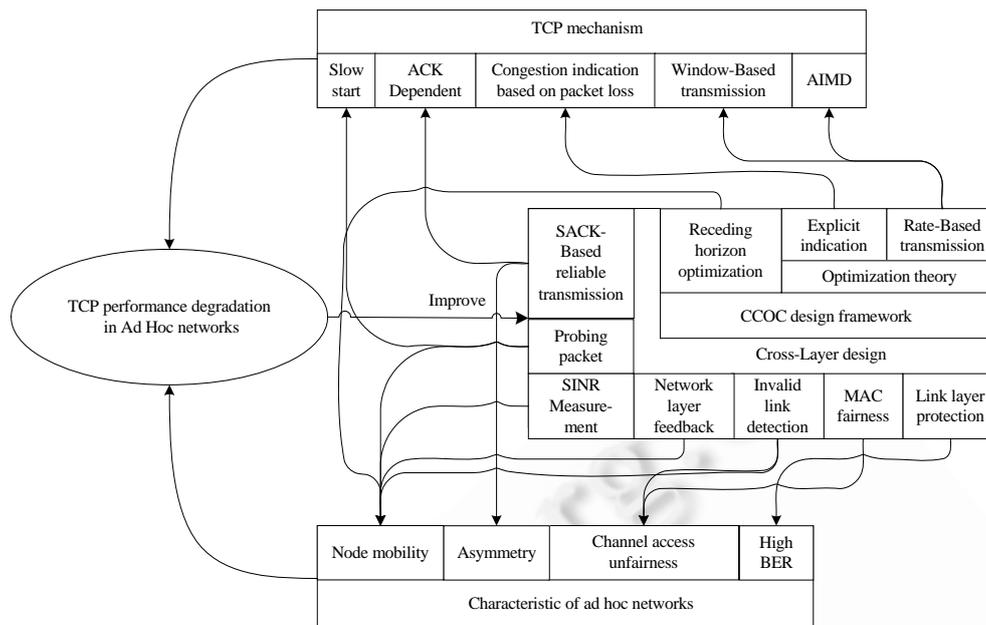


Fig.1 Cross-Layer design framework of CCOC

图1 CCOC 的跨层设计框架

本文运用跨层设计对拥塞控制协议进行优化设计,充分考虑物理层、MAC 层、网络层、传输层对拥塞控制的影响,以解决:根源(2)的虚假链路失效;根源(3)的数据包乱序和频繁路由切换;根源(4)和根源(9)的 ACK 依赖;根源(6).具体说,将采用如下几种策略:(1) 提高 MAC 接入的公平性;(2) 检测虚假链路失效;(3) 增强路由稳定性;(4) 实施网络层反馈机制和探测包机制实现快速启动;(5) 运用 SACK 实现可靠传输.下面详细说明这些改进方案的设计思想.

2.1 提高MAC接入的公平性

在 IEEE 802.11 DCF 中,节点在竞争信道时,如果信道忙,则节点需要在 $[0, 2^i CW_{\min}]$, $i=1, 2, \dots, 7$ 之间进行随机延迟,其中, CW_{\min} 是最小竞争窗口, i 是发送尝试失败次数.这种方法导致了竞争中失败的节点在下次竞争信道时处于更加不利的地位,从而导致信道接入的不公平性.针对 IEEE 802.11 DCF 的这种不公平性,利用链路的拥塞价格(拥塞价格的具体意义和计算方法参见文献[23])来提高其接入的公平性.具体做法是:当节点的下行链路的拥塞价格大于等于拥塞价格的门限 $CP_{\text{threshold}}$ 时,采用 $[0, \alpha CW_{\min} - 1]$, $\alpha \leq 1$ 之间进行随机延迟,让竞争中失败的节点引入较小的随机时延,使其处于竞争有利的地位,增大接入信道的概率,缓解信道接入的不公平性,从而减少出现虚假链路失效问题的可能性.

2.2 检测虚假链路失效

通过上节对 IEEE 802.11 DCF 的改进,可以在很大程度防止节点发生重传次数达到最大值的可能,但不能完全避免虚假链路失效的出现.本文采用了类似于文献[25]的方法来检测链路失效的真假.一旦发送节点的路由协议收到来自 MAC 层的链路失效报告,就触发向对应的接收节点发送 HELLO 消息,同时启动计数器,其初始值为 C_{hello} .每向接收节点发送 HELLO 消息 1 次,计数器的值就减 1.如果在 $C_{\text{hello}}=0$ 之前未收到来自接收节点对该 HELLO 消息的应答,则判定链路失效是真实的,并立即触发节点启动本地路由修复方案(见下节的描述).如果本地路由修复方案不成功,则向源端发送路由失效消息.如果定时器在超时前收到来自接收节点对 HELLO 消息的应答,则复位定时器,判定链路失效是虚假的.

目前,网络层的路由协议都会响应来自 MAC 层的链路失效报告,触发节点向数据包的源端发送路由失效

消息,从而触发漫长的路由发现和重构过程.通过判定链路失效的真实性,让路由协议能够正确区分 MAC 层信道接入失败的真正原因,使得源端的传输层不会对虚假链路失效采取不必要的拥塞控制.

2.3 增强路由稳定性

有效设计拥塞控制的最大难点在于 Ad Hoc 网络系统的动态性. Ad Hoc 网络中节点的自主移动导致网络拓扑结构的时变,导致频繁的路由失效和路由重构过程.在这过程中,源端将停止数据包的发送,严重影响了网络性能.而且路由重构后,在旧路由上迭代得到的发送速率不再适合新路由.如果在新路由上继续使用此发送速率来发送信息,则要么形成拥塞,要么就是没有充分利用允许的带宽.因此,增强路由的稳定性对拥塞控制的有效实施非常重要.为此,本文通过监测接收信号的信噪比 SIR 来预测判断链路的稳定程度,并在链路断开之前尽可能地通过本地路由恢复机制来更新路由以减少路由切换的频繁发生,提高对节点移动和通信中断的抵抗力.

SIR 测量.通过测量物理层的信号信噪比来指示邻居节点是否还在其通信范围内,以此区分出链路是否即将失效.该方法的核心思想是,如果信号信噪比在一段时间内都较低,则认为这是由于节点移动导致了真实的链路失效.具体做法是:每个节点维护一张链路状态监测表 LSDT(link state detection table),每个表项对应于该节点的上行链路,包含两项内容:第 1 项是“发送节点 ID 号”,它和本节点的 ID 号联合唯一标识上行链路;第 2 项是记录对应上行链路信噪比的指数加权滑动平均值 $SIR_i^{ave}(n)$.其中,

$$SIR_i^{ave}(n+1) = \alpha SIR_i^{ave}(n) + (1-\alpha)SIR_i(n+1) \quad (1)$$

α 为滤波系数,本文取值为 0.85.采用信噪比的平均值而不是信噪比瞬时值,是为了减少短暂突发干扰引发的信噪比抖动而导致不必要的误判.

对于相同“发送节点 ID 号”,节点只保留最新 1 条关于 SIR_i^{ave} 的记录.链路状态监测表的每一次刷新,节点根据最新的 SIR_i^{ave} 和阈值($SIR_Threshold$)的比较来判断目前的链路状态是否稳定并采取适当的响应.具体做法是:每当节点接收到一个数据帧时,同时测得相应的 SIR_i ,根据公式(1)计算得到相应的 SIR_i^{ave} ,如果 $SIR_i^{ave} \leq \eta SIR_Threshold$,就启动链路状态判定计数器,其初始值为 $C_{link_detection}$.每当节点接收到一个数据帧时,满足 $SIR_i^{ave} \leq \eta SIR_Threshold$,则计数器减 1;否则,计数器重新设置为初始值 $C_{link_detection}$.并断定链路状态的不稳定是暂时的,此链路继续用来传输信息.如果在计数器等于 0 时,链路的状态一直保持 $SIR_i^{ave} \leq \eta SIR_Threshold$,则断定链路状态的不稳定,触发接收节点向发送节点提出路由修复的请求.

本地路由由恢复.发送节点在接收到接收节点的路由修复请求后,实施本地修复过程,通过设置较小的 TTL(time to live)域值将泛洪范围限定在本地.在本地修复过程中,暂时用较高的传输功率在即将失效的不稳定链路上继续传输信息,使正在进行的通信不会被中断.需要指出的是,在暂时增加信号发射功率时,发送的 RTS(request to send)帧中需要包含增强后的发射功率值,让接收节点也是按照此发射功率值来反馈 CTS(clear to send)包和 ACK 包.一旦发送节点本地修复成功,则将从原来的不稳定链路无缝切换到新的稳定链路.

基于接收信号的信噪比对节点间链路状态稳定性进行预测,在链路可能要断开时,及时触发本地路由恢复机制,减少链路断开导致重新选择路由所花费的时间开销,降低了源端重新选择路由的可能性,提高了网络的包投递率并降低了控制包开销,减少了节点移动引发的拓扑变化对网络性能的影响.为此,使路由协议达到了以下目标:(1) 路由维持时间长,减少了路由更新次数,降低了路由控制开销;(2) 在节点不断移动的情况下,仍然能够获得较高的数据传送率;(3) 对路由故障做出预测,在尽量少的时间内恢复数据传送;(4) 降低了端到端延迟.

2.4 实施网络层反馈机制和探测包机制实现快速启动

尽管通过增强路由稳定性减少了节点移动导致路由中断的频繁程度,但是,如果路由本地修复不成功,则链路中断处的发送节点就启动网络层反馈机制.即一旦某一中间节点检测到下一跳节点移出它的发送范围(表明路由失效)时,便向源端发送路由失效通知 RFN(route failure notification)数据包.沿途的其他中间节点接收到该 RFN 数据包后,便令该路由无效.如果它有另外一条可到达目的节点的路由,则对其进行更新,并丢弃 RFN 数据包;否则,将其继续向源端转发.一旦源端接收到此 RFN 消息,则立刻停止发送后续数据包.因为重发的数据仍会

因为无法路由而被抛弃,这会浪费网络资源.

源端立即开始通过周期性地向接收端发送“探测包”来探测路由是否已经重建,每次发送的方式是一对“探测包”被连续(没有被其他数据包隔开)发送.需要指出的是,“探测包”和普通数据包有相同的包头格式,除了两个域的内容有差别:通过 FLAG 域置为 1 来标识它是“探测包”.中间节点收到“探测包”不需要像处理数据包一样修改拥塞价格域的值,接收端接收“探测包”需要记录其到达时间.假设一对“探测包”的前一个到达接收端的时间是 t_1 ,后一个为 t_2 ,”探测包”大小为 L ,则此路由的带宽被估计为 $BW=L/t_2-t_1$.这个值会被接收端反馈给源端,源端就以这个值作为初始发送速率开始重新发送数据,从而实现了快启动.这种快启动方式在连接初始阶段和路由切换时都可以采用,通过探测包机制及时反馈得到的网络速率估计,可以使该连接很快达到其所允许的速率,解决了 TCP 采用“慢启动”策略导致的网络资源利用率低下的问题.

2.5 运用SACK实现可靠传输

因为 CCOC 不是基于数据包丢失来推断网络是否拥塞的,因此,源端只需要来自接收端关于哪些数据包丢失了需要重传的报告,无需正确判断其原因是因网络拥塞还是路由故障,或者是误码率高造成的.为了避免像在 TCP 中依赖 ACK 的自定时减少数据包的信道接入机会那样,可靠性机制采用了选择性确认(SACK).这样,一个 SACK 包就可以报告多个数据包接收正确与否的情况.因此,接收端只要实施粗粒度的反馈就可以实现可靠性报告.

3 CCOC 的协议体系结构

CCOC 的基本思想是,通过数据包的 CCOC 数据包头将链路拥塞信息显式地反馈给数据源端,从而达到有效控制网络数据流量、防止网络发生拥塞、提高网络带宽利用率的目的.

3.1 CCOC数据包头格式

每个数据包都携带了一个数据包头,其格式如图 2 所示,由下列域构成:源端口、目的端口、序号、拥塞度量、标志位 FLAG,除了拥塞度量和 FLAG 以外,其他项与 TCP 的数据包格式类似.拥塞度量用于携带中间节点拥塞价格信息,具体情况参见文献[23].源端将其设置为 0,中间节点对每个经过的数据包根据输出接口队列长度,调整每个数据流的拥塞价格值.可见,拥塞度量是捎带在数据包中的.通过 FLAG 域来区分数据包和探测包.

SACK 包的数据包头,其格式如图 3 所示.协议具体实现时,每个 SACK 反馈包的头中设有一个字段“SACK 反馈序号”用于 SACK 进行编号,源端依据此编号判断是否收到新的报告以开始新的速率更新周期.

Source port		Destination port	
Sequence No.		Congestion measurement	
SYN	FIN	FLAG	Payload

Fig.2 Head format of data packet
图 2 数据包的头格式

Source port		Destination port	
SACK sequence No.		Congestion measurement	
SACK length value		Payload	

Fig.3 Head format of SACK packet
图 3 SACK 反馈包的头格式

3.2 CCOC的工作过程

源端实施快启动方案与接收端建立连接,取得发送速率的初始值.源端在发送数据包时,在其数据包头设置拥塞度量为 0,中间节点则对每个经过的数据包的拥塞度量累加上相应的链路拥塞价格.SACK 包通过数据包头的拥塞度量反馈给源端,源端据此调整新一轮的发送速率.循环往复,直到路由切换后重新进入快启动.

3.3 CCOC的中间节点

在 CCOC 中,中间节点的主要功能是实现拥塞价格的计算、收集和修改.此外,还实现了以下功能:(1) 提高

MAC 接入的公平性;(2) 检测虚假链路失效;(3) 构建稳定的路由;(4) 实施网络层反馈机制.为了实现这些功能,中间节点是跨越了物理层到网络层,具体如图 4 所示.

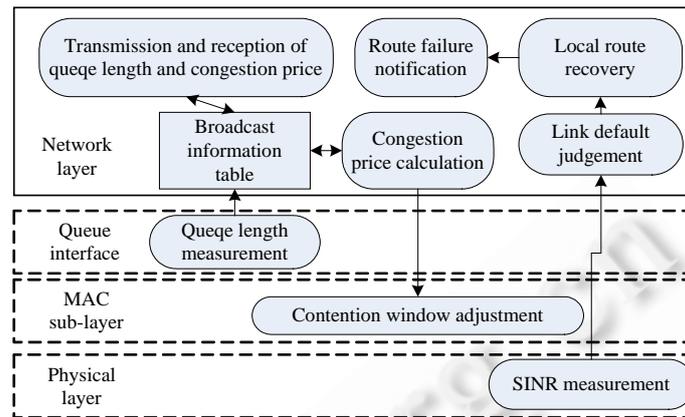


Fig.4 Function of inter-nodes

图 4 中间节点的功能示意图

3.4 CCOC的源端

在 CCOC 中,源端主要实现快启动、拥塞控制和可靠性控制功能.快启动的实施是在连接的初始阶段和路由切换后重新开始传输阶段,其拥塞控制的实现是根据接收端反馈的路径上的中间节点提供的拥塞度量信息.

可靠性控制是依据接收端反馈的 SACK 提供的信息来实施的,SACK 指示的未被准确接收的数据包有优先发送权.需要指出的是,基于速率的传输需要一个定时器,幸运的是,这引起的负荷不会很大^[5].

3.5 CCOC的接收端

考虑到尽可能降低反馈通道上的通信量,接收端摒弃了每数据包确认方式,而是采用 SACK 确认方式.接收端实施周期性反馈以辅助源端实现拥塞控制和可靠性控制,确保每个数据包被准确接收.此外,它也为每个连接记录了前一时刻的拥塞度量值.为了周期性地反馈实施,接收端需要一个定时器,定时器的值在连接或者路由变化后开始的前 n 次是等于一个 RTT 时间,让拥塞控制尽快地收敛到网络的最优解.此后,接收端在可跟踪网络动态特性的情况下,尽可能少地反馈 SACK.对每个属于同一连接的数据包,根据所携带的拥塞度量值来判断

$\left| \frac{\lambda^s(n+1) - \lambda^s(n)}{\lambda^s(n)} \right| \leq \gamma$ 是否成立? $\lambda^s(n)$ 是指第 n 时刻从源段 s 出发的信息流到达接收端所携带的拥塞度量总和^[23].如果其相对变化小于预定的取值 γ ,即 $\left| \frac{\lambda^s(n+1) - \lambda^s(n)}{\lambda^s(n)} \right| \leq \gamma$,则这个 RTT 就不实施反馈给源端,以尽量降低反馈通道上的通信量;如果不成立,则实施反馈给源端.在本文的仿真中,设置 n 为 20, γ 为 0.05.

在一个 SACK 反馈包中,最多可以记录 10 个数据包是否被接收端准确接收的情况.每次 SACK 反馈可以有不同数量的记录,通过域“SACK 长度值”制定.

接收端的应用层是按序处理数据的.如果某个序号为 n 的数据包未被接收端准确接收,则序号大于 n 的数据包就被暂时存放在接收端的缓存中,但是缓存容量是有限的,当缓存中队列长度达到一定值,接收端也会触发发送 SACK,要求源端赶快重新发送未被准确接收的数据包.

4 仿真实验

本节用 NS2 仿真平台来比较 CCOC 和 TCP-Reno,ATCP^[6]的性能.仿真环境如下:各节点使用全向天线,无线传播模型采用 TwoWayGround 模型.信道传输速率为 2Mbps.MAC 协议按第 2 节所述修改的 IEEE 802.11 协议,

有效传输距离为 250m,侦听距离为 550m;路由采用按第 2 节所述修改的 AODV 协议;每个节点的缓存大小为 50.数据包大小等于 1 000byte,传输的业务为 FTP 数据.所有仿真实验的运行时间都为 200s.本文采用的公平性指数 (fairness index,简称 FI)定义为

$$FI = \frac{(\sum_{i=1}^n T_i)^2}{n \sum_{i=1}^n (T_i)^2},$$

其中, T_i 是第 i 个信息流的吞吐量, n 是信息流总数.FI 的取值越接近于 1,则信息流之间的公平性越高.

4.1 网络拓扑

100 个节点形成一个 10×10 的网格拓扑结构.两个相邻节点之间的距离设置为 200m.在此网络拓扑上随机生成 4,6,8,10 个信息流.这些信息流在每个仿真中持续连接 200s.本文重复了 10 次仿真,得到了图 5 显示的关于 CCOC, TCP-Reno 和 ATCP 在 4,6,8,10 个信息流的网络场景下的平均公平性指数.进一步地,图 6 显示了 6 个信息流的网络场景下的瞬时公平性指数.从图 5 和图 6 可以看出,CCOC 比 TCP-Reno 和 ATCP 能更好地达到短期和长远公平性.在 TCP 中,公平性的问题主要是由于过度的链路竞争,由此产生了虚假链路失效导致某些 TCP 流被饿死.而 CCOC 利用链路的拥塞价格来提高 MAC 层接入的公平性.此外,CCOC 能够快速侦测并恢复虚假链接失效,因此能够取得更高的 FI.

网络累计吞吐量是指在整个仿真持续时间内所有信息流的吞吐量总和.CCOC 在 4,6,8,10 个信息流的网络场景下所取得的网络累计吞吐量均高于 TCP-Reno 协议 30%~50%,ATCP 协议 30%~50%.以 6 个信息流的网络场景为例,CCOC 取得的网络累计吞吐量为 726 740 bit/s,而 TCP-Reno 和 ATCP 分别为 455 020 bit/s 和 546 030 bit/s.

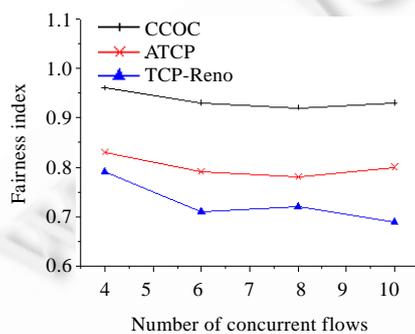


Fig.5 Fairness index in grid-topology (different flows)

图 5 网格拓扑在多种信息流数下的公平性指数

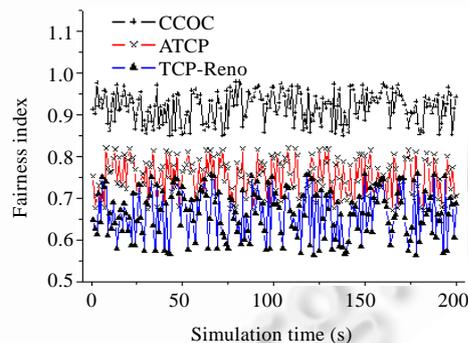


Fig.6 Instantaneous fairness index in grid-topology (six flows)

图 6 网格拓扑瞬时公平性指数(信息流数为 6)

4.2 连十字链路

连十字链路仿真场景如图 7 所示(采用文献[26]中类似的网络拓扑结构),图中 flow 1 跨越了 4 个瓶颈节点,而其他 flow2,flow3 和 flow4 只跨越了一个瓶颈节点.仿真结果如图 8 所示.

图 8 表明,在 CCOC 中,flow 1 能够获得其他 flow 吞吐量的 1/4 左右.其他 flow2,flow3 和 flow4 能够比较公平地共享网络资源,主要是因为 CCOC 实施了比例公平.而在 TCP 协议中,未能将网络资源有效地分配给跨越了 4 个瓶颈节点的 flow 1,短信息流几乎霸占了全部的信道资源,而长信息流几乎被饿死.而且,CCOC 取得的网络总吞吐量是 TCP 的 1.3 倍,体现了 CCOC 的有效性和公平性的改进.

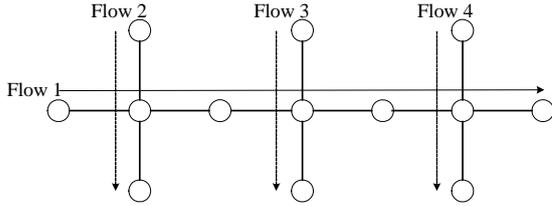


Fig.7 Parking-Lot topology
图 7 连十字链路

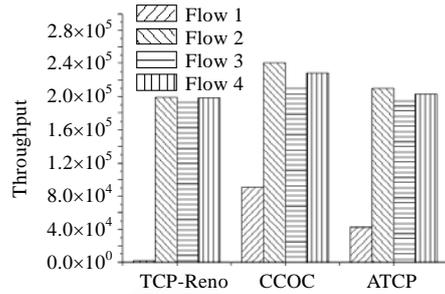


Fig.8 Overall throughput of in a parking-lot topology
图 8 连十字链路下的综合吞吐量

4.3 伪动态拓扑

在如图 9 所示的伪动态网络拓扑中,有 2 个信息流在传输信息.从仿真 0s~50s 之间,只有 flow 1 传输信息;从仿真 50s 开始,flow 2 也开始传输信息,此时,节点 6 从(150,250)以 3m/s 的速率做直线运动到(150,550)后停止运动.考察了 CCOC 和 TCP-Reno 在这样场景下的瞬时吞吐量,其结果分别如图 10 和图 11 所示.

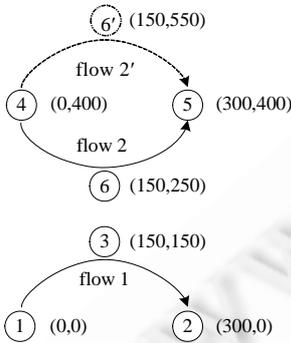


Fig.9 Pseudo-Dynamic topology
图 9 伪动态拓扑

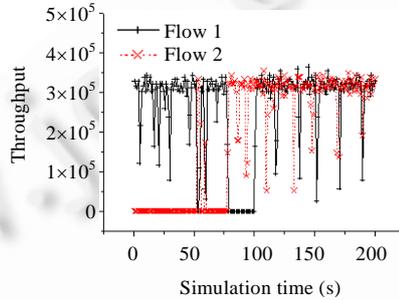


Fig.10 Instantaneous throughput of TCP-Reno
图 10 TCP-Reno 的瞬时吞吐量

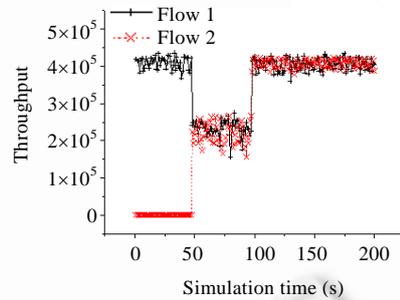


Fig.11 Instantaneous throughput of CCOC
图 11 CCOC 的瞬时吞吐量

从这两个图的结果比较来看,再次说明了 CCOC 比 TCP-Reno 无论在吞吐量还是在公平性上都有了大幅度的改进.TCP-Reno 在仿真场景的 50s~100s 之间,由于节点 3 和节点 6 相互干扰,因此造成了或者 flow 1 完全霸占了信道,或者 flow 2 完全霸占了信道.CCOC 通过节点之间拥塞信息的交流改进了信息流之间的公平性.

4.4 动态网络性能分析

先前的工作^[23]详细考察了 CCOC 中的部分机制(PCA,PCA+AOS),TCP-Reno 和 ATCP 在常见的 3 种移动模型下的性能,包括 RWP(random way point)模型、FW(freeway mobility)模型和 MH(manhattan mobility)模型.NS2 仿真实验结果表明,PCA 和 PCA+AOS 在网络性能上比 TCP 和 ATCP 有了明显的改进^[23].而 CCOC 又取得了优于 PCA 和 PCA+AOS 的 5%~10%左右的性能提高.限于篇幅,本文省略了这部分研究内容的描述.

5 结 论

本文首先基于网络协议层框架充分分析了 TCP 应用到 Ad Hoc 网络导致性能下降的原因所在,详细解释了这些因素是如何降低 TCP 性能的;然后,有针对性地提出了全新的适合 Ad Hoc 网络特性的跨层优化拥塞控制协议 CCOC.之后,本文详细描述了 CCOC 运用的跨层设计框架中改进方案的设计思想;简要描述了 CCOC 的协议

体系结构.最后,详细的NS2仿真实验结果表明,在几乎所有的仿真场景和移动环境下,CCOC取得的性能都要比TCP和ATCP要好.因此,CCOC为Ad Hoc网络实现有效资源分配提供了有效的拥塞控制方案.

References:

- [1] Xu WQ, Wu TJ. TCP issues in mobile ad hoc networks: Challenges and solutions. *Journal of Computer Science and Technology*, 2006,21(1):72–81. [doi: 10.1007/s11390-006-0072-2]
- [2] Feng YJ, Sun LM, Qian HL, Song C. Improving TCP performance over MANET: A survey. *Journal of Software*, 2005,16(3): 434–444 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/434.htm>
- [3] Holland G, Vaidya NH. Analysis of TCP performance over mobile ad hoc networks. *Wireless Networks*, 2002,8(2-3):219–230.
- [4] Chandran K, Raghunathan S, Venkatesan S, Prakash R. A feedback-based scheme for improving TCP performance in ad hoc wireless networks. *IEEE Personal Communications*, 2001,8(1):34–39. [doi: 10.1109/98.904897]
- [5] Kim DK, Toh CK, Choi Y. TCP-BuS: Improving TCP performance over wireless ad hoc networks. *Journal of Communications and Networks (JCN)*, 2001,3(2):1–12.
- [6] Liu J, Singh S. ATCP: TCP for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 2001,19(7): 1300–1315. [doi: 10.1109/49.932698]
- [7] Dyer T, Boppana R. A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In: *Proc. of the ACM MOBIHOC 2001*. Long Beach: ACM, 2001. 56–66.
- [8] Wang F, Zhang Y. Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response. In: *Proc. of the ACM MobiHoc 2002*. Lausanne: ACM, 2002. 217–225.
- [9] Chen K, Xue Y, Shah SH, Nahrstedt K. Understanding bandwidth-delay product in mobile ad hoc networks. *Computer Communications*, 2004,27(10): 923–934. [doi: 10.1016/j.comcom.2004.01.020]
- [10] Sundaresan K, Anantharaman V, Hung YH, Sivakumar AR. ATP: A reliable transport protocol for ad hoc networks. *IEEE Trans. on Mobile Computing*, 2005,4(6):588–603. [doi: 10.1109/TMC.2005.81]
- [11] Su Y, Gross T. WXCP: Explicit congestion control for wireless multi-hop networks. In: *Proc. of the IWQoS 2005*. LNCS 3552, Berlin, Heidelberg: Springer-Verlag, 2005. 313–326.
- [12] Kliazovich D, Granelli F. Cross-Layer congestion control in ad hoc wireless networks. *Ad Hoc Networks*, 2006,4(6):687–708. [doi: 10.1016/j.adhoc.2005.08.001]
- [13] Scheuermann B, Lochert C, Mauve M. Implicit hop-by-hop congestion control in wireless multihop networks. *Ad Hoc Networks*, 2008,6(2):260–286. [doi: 10.1016/j.adhoc.2007.01.001]
- [14] Li M, Agrawal D, Ganesan D, Venkataramani A. Block-Switched networks: A new paradigm for wireless transport. In: *Proc. of the 6th ACM/USENIX Symp. on Networked Systems Design and Implementation (NSDI 2009)*. Boston, 2009. <http://whitepapers.zdnet.com/abstract.aspx?docid=383708>
- [15] Warriar A, Janakiraman S, Angtae S, Rhee I. DiffQ: Practical differential backlog congestion control for wireless networks. In: *Proc. of the INFOCOM*. Rio de Janeiro, 2009. 262–270.
- [16] Rangwala S, Jindal A, Jang K, Psounis K, Govindan R. Understanding congestion control in multi-hop wireless mesh networks. In: *Proc. of the ACM MOBICOM*. San Francisco: ACM, 2008. 291–302.
- [17] Sun L, Li B, Zhou X. A survey of congestion control technology for wireless sensor networks. *Journal of Computer Research and Development*, 2008,45(1):63–72 (in Chinese with English abstract).
- [18] Yin X, Zhou X, Huang R, Fang Y, Li S. Fairness-Aware congestion control scheme in wireless sensor networks. *IEEE Trans. on Vehicular Technology*, 2009,58(9):5225–5234. [doi: 10.1109/TVT.2009.2027022]
- [19] Le T, Hu W, Corke P, Jha S. ERTP: Energy-Efficient and reliable transport protocol for data streaming in wireless sensor networks. *Computer Communications*, 2009,32(7-10):1154–1171. [doi: 10.1016/j.comcom.2008.12.045]
- [20] Chen J, Xu WQ, He S, Sun Y, Thulasiramanz P, Shen X. Utility-Based asynchronous flow control algorithm for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 2010. <http://www.sensor.net.cn/publication.html>
- [21] Zhang L, Shenker S, Clark D. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic. *ACM SIGCOMM Computer Communication Review*, 1991,21:133–147.

- [22] Zhai H, Chen X, Fang Y. Rate-Based transport control for mobile ad hoc networks. In: Proc. of the IEEE Wireless Communications and Networking Conf. (WCNC 2005). New Orleans, 2005.
- [23] Xu WQ, Wu TJ, Wang YM, Zhang YH, Chen JM. Congestion control for high dynamic ad hoc networks: Price cooperation and receding optimization. Journal of Software, 2008,19(9):2389–2402 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/2389.htm> [doi: 10.3724/SP.J.1001.2008.02389]
- [24] Xu WQ, Wu TJ, Wang YM, Zhang YH, Chen JM. Adaptive congestion control strategy for multirate multicast sessions in ad hoc networks. Journal of Software, 2008,19(3):769–778 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/769.htm> [doi: 10.3724/SP.J.1001.2008.00769]
- [25] Li Y, Chen QB, Long KP, Wu SQ. Analyzing and improving the TCP stability in wireless ad hoc networks. Journal of Software, 2003,14(6):1178–1186 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/1178.htm>
- [26] Tan K, Jiang F, Zhang Q, Shen X. Congestion control in multihop wireless networks. IEEE Trans. on Vehicular Technology, 2007, 56(2):863–873. [doi: 10.1109/TVT.2007.891405]

附中文参考文献:

- [2] 冯彦君,孙利民,钱华林,宋成. MANET 中 TCP 改进研究综述. 软件学报, 2005, 16(3):434–444. <http://www.jos.org.cn/1000-9825/16/434.htm>
- [17] 孙利民,李波,周新运. 无线传感器网络的拥塞控制技术. 计算机研究与发展, 2008, 45(1):63–72.
- [23] 徐伟强,吴铁军,汪亚明,张云华,陈积明. 强动态 Ad Hoc 网的拥塞控制: 价格协作和滚动优化. 软件学报, 2008, 19(9):2389–2402. <http://www.jos.org.cn/1000-9825/19/2389.htm> [doi: 10.3724/SP.J.1001.2008.02389]
- [24] 徐伟强,吴铁军,汪亚明,张云华,陈积明. 用于 Ad Hoc 网络的自适应多速率多播拥塞控制策略. 软件学报, 2008, 19(3):769–778. <http://www.jos.org.cn/1000-9825/19/769.htm> [doi: 10.3724/SP.J.1001.2008.00769]
- [25] 李云,陈前斌,隆克平,吴诗其. 无线自组织网络中 TCP 稳定性的分析及改进. 软件学报, 2003, 14(16):1178–1186. <http://www.jos.org.cn/1000-9825/14/1178.htm>



徐伟强(1975—),男,浙江杭州人,博士,副教授,主要研究领域为 Ad Hoc 网络,无线传感器网络,拥塞控制,网络资源优化分配,跨层优化.



汪亚明(1972—),男,博士,教授,博士生导师,主要研究领域为计算机图像,运动估计,计算机网络,智能算法.



俞成海(1975—),男,讲师,主要研究领域为信息安全,网络通信.



刘良桂(1975—),男,博士,讲师,主要研究领域为无线网络,路由协议.



张云华(1965—),男,博士,研究员,主要研究领域为计算机网络,软件工程,计算机控制.