

无线自组织网络中 TCP 稳定性的分析及改进*

李云^{1,2+}, 陈前斌², 隆克平², 吴诗其¹

¹(电子科技大学 信息研究所, 四川 成都 610054)

²(重庆邮电学院 光互联网及无线信息网络研究中心, 重庆 400065)

Analyzing and Improving the TCP Stability in Wireless Ad Hoc Networks

LI Yun^{1,2+}, CHEN Qian-Bin², LONG Ke-Ping², WU Shi-Qi¹

¹(Institute of Information, University of Electronic Science and Technology, Chengdu 610054, China)

²(Special Research Centre for Optical Internet and Wireless Information Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

+Corresponding author: Phn: 86-23-62460222, E-mail: liyun@cqupt.edu.cn

<http://www.cqupt.edu.cn>

Received 2002-11-19; Accepted 2003-01-20

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.

<http://www.jos.org.cn/1000-9825/14/1178.htm>

Abstract: Due to the MAC (medium access control) protocol, routing protocol and TCP itself, the TCP flow is very unstable in the wireless Ad Hoc networks based on IEEE802.11. However, the main reasons that make the TCP flow unstable are the unfairness of MAC protocol and the false link failure message that causes the long route discovery procedure. Combining with the IEEE802.11 MAC protocol and DSR routing protocol, these causes are analyzed in detail and the enhanced algorithm to IEEE802.11 MAC protocol and DSR routing protocol is given in this paper. The simulational results show that the proposed algorithm can ultimately avoid the instability of TCP flow and increase its average throughput in the wireless Ad Hoc networks.

Key words: wireless Ad Hoc network; MAC (medium access control) protocol; routing protocol; TCP flow; stability

摘要: 在无线自组织网络中,基于 IEEE 802.11 的 TCP 流存在严重的不稳定性,其原因与 MAC 协议、路由协议和 TCP 本身均有一定的关系,但最根本之处在于 MAC 协议的不公平性以及假的链路失效消息导致了不必要

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA120303 (国家高技术研究发展计划(863)); the Program of Chongqing Science and Technology Committee of China under Grant No.D2002-35 (重庆市科委计划项目); the Young Foundation of Chongqing University of Posts and Telecommunication of China under Grant No.A2002-20 (重庆邮电学院青年教师基金); the Open Foundation of the Key Laboratory of Mobile Communication of Chongqing of China under Grant No.YDFS-2001-02 (重庆市移动通信重点实验室开放基金); the Key Subject Developing Foundation of Communication and Information System of Chongqing of China under Grant No.YDJS-2002-02 (重庆市通信与信息系统重点学科建设基金)

第一作者简介: 李云(1974—),男,四川西充人,博士生,主要研究领域为 MAC 协议性能分析和改进,无线自组织网络 QoS.

的耗时的路由发现过程.结合 IEEE802.11 的 MAC 协议和 DSR 路由协议,对这些原因进行了深入的理论分析和仿真实验,并提出了针对 MAC 协议和路由协议的改进算法.仿真结果证明,提出的改进算法不仅能基本上避免 TCP 流的不稳定性,还能够极大地提高 TCP 流的平均吞吐量.

关键词: 无线自组织网络;MAC 协议;路由协议;TCP 流;稳定性

中图分类号: TP393 **文献标识码:** A

无线自组织网络(wireless ad hoc networks)是一种新型的无线网络.与传统的带固定设备(如基站)的无线网络相比,其显著特点是网络中没有固定的通信设施,网络中所有通信节点都是移动的,每个移动节点既是终端又是路由器,能够提供包的存储转发功能.由于无须固定通信设施的支持,因此,无线自组织网络具有很高的可靠性和灵活性,可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.近年来,无线自组织网络已引起了人们的广泛注意,并成为一个新的研究热点.

无线自组织网络的关键技术包括实现高效的路由算法、设计有效的分布式介质访问控制(media access control,简称 MAC)协议、对区分业务和服务质量(quality of services,简称 QoS)的支持以及网络安全等等.目前还没有很好的 QoS 及网络安全的解决方案.但对于动态选路问题,目前已有一些较好的路由协议,如基于目的节点生成序号的距离矢量路由(destination-sequenced distance-vector routing,简称 DSDV)^[1]、分布式按需距离矢量路由(ad hoc on-demand distance vector routing,简称 AODV)^[2]、动态源路由(dynamic source routing,简称 DSR)^[3] 等等.对已有的路由协议,可大致分为两类,即表驱动(table-driven)路由协议和按需(on-demand)路由协议.前者如 DSDV 路由协议,后者如 DSR,AODV 路由协议.文献[4]对表驱动路由协议和按需路由协议的性能进行了比较,总的来说,当网络拓扑频繁变化时,按需路由协议的性能优于表驱动路由协议;当网络拓扑变化相对较慢时,表驱动路由协议的性能优于按需路由协议.

在分布式 MAC 协议方面,已有的机制也很多,如传统的载波侦听多址接入(carrier sense multiple access,简称 CSMA)、带冲突避免的载波侦听多址接入(carrier sense multiple access with collision avoidance,简称 CSMA/CA)以及无线局域网介质访问协议(media access protocol for wireless LAN,简称 MACAW)^[5]、IEEE802.11 的分布式协调功能(distributed coordination function,简称 DCF)^[6]等.其中,IEEE 802.11 DCF 本身是为无线局域网(wireless LAN,简称 WLAN)制定的 MAC 规范,然而目前,IEEE802.11 已被广泛应用于无线自组织网络的各种仿真和测试床中,IEEE802.11 DCF 正成为无线自组织网络的事实上的 MAC 规范.因此,本文在对无线自组织网络中的 TCP 流的稳定性进行仿真分析时采用了 IEEE802.11 DCF 作为 MAC 协议,其相应的改进也是针对 IEEE802.11 DCF 的.

基于 IEEE 802.11 的无线自组织网络中的 TCP 流存在严重的不稳定性,这一点已逐渐引起了人们的注意.文献[7]对 IEEE802.11 下的 TCP 流的稳定性进行了仿真分析,并引出了 TCP 流的不稳定性问题,但将这一问题仅仅归因于 IEEE802.11 MAC 协议的不公平性,而且主要停留在对 IEEE802.11 DCF 的分析上,而没有将 MAC 协议、路由协议和 TCP 有机地结合在一起进行分析,尽管 IEEE802.11 DCF 本身的不公平性是造成 TCP 流不稳定的根本原因.同时,它没有从理论上揭示出造成 IEEE802.11 DCF 不公平性的原因,也没有给出相应的改进方案.

本文结合 IEEE802.11 DCF、路由协议和 TCP 本身,对无线自组织网络中的 TCP 流的稳定性进行仿真,揭示了造成 TCP 流不稳定的根本原因.进而对 IEEE802.11 DCF 的公平性进行了理论分析,并提出了一种对 IEEE802.11 DCF 和路由协议的改进算法.通过仿真分析,证明了改进算法不仅能基本上避免 TCP 流的不稳定性,同时能在很大程度上提高 TCP 流的平均吞吐量.

1 无线自组织网络中 TCP 流稳定性的分析

为了对无线自组织网络中 TCP 流的稳定性进行仿真分析,我们采用了 LBNL(Lawrence Berkeley National Laboratory)开发的仿真平台 NS-2^[8].仿真拓扑如图 1 所示,相邻节点间的距离为 150m,相邻节点在彼此的发送和

接收范围之内.这一线状拓扑可代表无线自组织网络中的典型的多跳连接.仿真的物理层和数据链路层采用 IEEE802.11,路由协议用 DSR,TCP 版本为 Newreno^[9].有关这些协议的主要参数见表 1.



Fig.1 Simulation topology

图 1 仿真拓扑

Table 1 Simulation parameters

表 1 仿真参数

Parameter name	Value	Parameter name	Value	Parameter name	Value
Channel bit rate	2Mbit/s	MAC header length	144bits	TCP segment length	5600bits
Slot time	20μs	RTS length	160bits	ACK Length	112bits
SIFS	10μs	CTS length	112bits	DSR initial Backoff time	0.05s
DIFS	50μs	CW_{min}	16	DSR maximal Backoff time	10s
PHY header length	192bits	CW_{max}	256	DSR Backoff factor	4

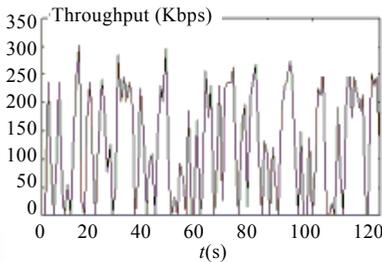


Fig.2 Throughput of TCP flow from node 1 to node 5

图 2 节点 1 到节点 5 的 TCP 流的吞吐量

我们在节点 1 和节点 5 之间建立一条 TCP 连接,其上承载 FTP 流量,FTP 流从 1.0s 开始,120s 后结束,每隔 1s,计算该秒内 TCP 的平均吞吐量,结果如图 2 所示.由图 2 可知,TCP 流的性能严重不稳定.在 120s 的时间里,TCP 流的吞吐量有 30 次为 0,这表明,在相应的 1s 内,成功发送的 TCP 报文段数为 0.

下面,我们将根据仿真记录文件(trace file),结合 IEEE802.11 DCF,DSR 和 TCP,对造成 TCP 流吞吐量不稳定的原因进行分析.表 2 列出了 23.684021s 到 25.787918s 这段时间内发生的事件.在表 2 中,RET 表示 RTS 重传次数达到最大值(在 IEEE802.11 中,缺省为 7),DSR ARP 表示生命周期为 1 跳的路由请求消息,RE 表示路由差错,RREQ,RREP 分别表示路由请求和路由应答.

Table 2 The events at about 24s

表 2 24s 前后发生的事件

Time (s)	Events	Time (s)	Events
23.684 0	RET at node 4, node 4 send RE to node 5	23.747 7	Because of collision, node 4 dropped the DSR RREQ MAC frame from node 5
23.685 6	Node 5 received RE	24.695 5	TCP timeout,send duplicated TCP segment
23.695 5	Node 1 received the last ACK and send the last TCP segment	25.757 4	Node 5 send DSR RREQ again
23.713 0	Node 5 send DSR ARP	25.757 4	Node 5 send DSR RREQ MAC frame again
23.716 1	Node 4 received DSR ARP	25.759 3	Node 4received and forwarded DSR RREQ
23.746 8	Node 5send DSR RREQ	25.759 7	Node 3 received DSR RREQ and send RREP
23.746 8	Node 5 send DSR RREQ MAC frame	25.762 7	Node 5 received DSR RREP
23.747 5	Node 3 send RTS	25.767 9	Node 1 received new ACK and send TCP segment

由表 2 可知,在 2s 多的时间里(23.695 540s~25.767 918s),节点 1 没有发送任何新的 TCP 段,即在这段时间内,TCP 流的吞吐量为 0.TCP 的发送方之所以没有发送新的 TCP 段,原因在于发送方没有收到任何接收方发来的 ACK,而导致 ACK 不能到达 TCP 发送方的原因在于节点 4 因 RET(在 23.684 021s 发生)而错误地认为到节点 3 的链路故障,它丢弃所有已缓存的下一跳为节点 3 的数据包,并发送 RE 消息到这些包的源节点,从而触发了耗时的路由发现过程(路由请求消息的碰撞和 DSR 的二进制指数回退机制进一步增加了该过程的时间),直到节点 5 重新发现到节点 1 的路由(从 23.713 087s~25.762 721s),导致这一连串事件的根本原因是节点 4 发生了 RET.即节点 4 在连续 7 次发送 RTS 后均没有收到节点 3 发回的 CTS.

在无线局域网中,由于移动节点彼此在接收范围之内,因此,我们可以认为节点 4 没有收到节点 3 的 CTS 是由于节点 4 与其他节点“同时”发送,在节点 3 处出现碰撞,使节点 3 不能收到节点 4 的 RTS.根据图 1 所示的拓扑,即在节点 2、节点 3 与节点 4 这 3 个节点中有两个以上同时发送.根据文献[10]对 IEEE802.11 DCF 的性能分析可知,设有 n 个移动节点相互干扰,当每个节点的发送队列均为非空时,其中任一节点在任一间隙发送的包出现碰撞的概率 p 为

$$p = 1 - (1 - \tau)^{n-1} \tag{1}$$

其中, τ 为任一给定移动节点在任一间隙发送包的概率,其解析式为

$$\tau = \frac{2(1-2p)}{(1-2p)(w+1) + pw[1-(2p)^m]} \tag{2}$$

在式(1)和式(2)中, n 为相互干扰的节点数, m 为最大回退等级, w 为最小竞争窗口,即 CW_{min} . 根据式(1)和式(2),显然,由于这种碰撞使任一节点连续 7 次尝试均失败的概率为

$$p_{RET} = [1 - (1 - \tau)^{n-1}]^7 \tag{3}$$

式(2)和式(3)构成非线性方程组,可根据具体的 n, w, m , 用计算机数值解法求相应的 p_{RET} . 根据如图 1 所示的仿真拓扑和表 1 中的仿真参数,我们有 $m=3, w=16, n=3$ (对节点 4 而言,节点 2、节点 3 和节点 4 相互干扰,因此, $n=3$). 代入 n, w, m , 用计算机数值解法,我们求得节点 4 的 p_{RET} 为 1.3×10^{-8} . 由此可知,因同时发送引起节点 4 相继 7 次发送均不成功的概率是很小的,这与上面的仿真结果不符. 因此,我们有必要对导致节点 4 连续 7 次发送尝试均失败的原因作进一步的分析. 为此,我们在表 3 中列出了节点 4 发生 RET 之前的事件.

Table 3 The events before node 4 RET
表 3 节点 4 发生 RET 之前发生的事件

Time (s)	Events	Time (s)	Events
23.664 5	Node 1 send RTS to node 2	23.673 3	@Node 2 send node 1 CTS
23.664 7	@Node 2 send CTS to node 1	23.673 5	Node 1 send node 2 data frame
23.664 9	Node 1 send node 2 data frame	23.674 9	* Node 4 send node 3 RTS
23.665 2	*Node 4 send node 3 RTS	23.674 2	** Node 3 received RTS
23.665 3	**Node 3 received RTS	23.680 8	Node 1 send node 2 CTS
23.666 1	*Node 4 send node 3 RTS	23.681 0	@ Node 2 send node 1 CTS
23.666 3	** Node 3 received RTS	23.681 2	Node 1 send node 2 data frame
23.665 9	* Node 4 send node 3 RTS	23.682 3	* Node 4 send node 3 RTS
23.666 7	** Node 3 received RTS	23.682 4	** Node 3 received RTS
23.668 4	Node 2 send node 3 RTS	23.683 6	* Node 4 send node 3 RTS
23.668 5	* Node 4 send node 3 RTS	23.683 8	** Node 3 received RTS
23.668 7	Because of collision, node 3 dropped RTS from node 4	23.684 0	Node 4 RET

由表 3 可知,在节点 4 RET 之前,节点 4 进行了 7 次发送 RTS 的尝试(如符号*所示),其中,只有一次与节点 2 的发送发生碰撞,而其他的 6 次发送,节点 3 都正确收到了 RTS(如符号**所示),然而,由于 IEEE 802.11 DCF 采用虚拟载波侦听方式,在收到 RTS 之前,节点 3 已收到节点 2 的用于预留信道的 CTS,并设置网络分配矢量 NAV(network allocation vector),因此,节点 3 只是简单地将收到的 RTS 帧丢弃,并不发送 CTS. 当节点 1 成功发送完一个数据帧之后(收到节点 2 的 ACK 确认),它将在 $[0, CW_{min}-1]$ 之间按均匀分布随机选择一个回退时隙. 此时,节点 4 由于已经进行了多次重试,按照 IEEE 802.11 DCF 的指数回退机制,它将在 $[0, 2^m \times CW_{min}-1]$ 之间回退, m 为已做的尝试次数. 这样,节点 1 在竞争信道时处于更有利的地位,它将早于节点 4 发送 RTS,节点 2 在收到 RTS 后发送 CTS,节点 3 收到该 CTS 设置 NAV,此后,节点 3 即使收到节点 4 的 RTS 也不能回送 CTS. 这样,节点 3 将始终没有向节点 4 发送 CTS 的机会,最终导致节点 4 RET.

对图 2 中其他吞吐量为 0s 的情况,原因同 23s~25s 时基本相同. 不同的是发生 RET 的节点不同. 如 70s 处的 0 吞吐量是因为在 68.141 099s 时节点 1 RET 引起的,而节点 1 的 RET 又主要是因为它发送到节点 2 的 RTS 同节点 3 到节点 4 的 TCP 数据帧碰撞(也包括同节点 3 到节点 4 的 RTS 帧碰撞,因收到节点 3 到节点 4 的 RTS 而预留信道等原因).

为了对上述现象进行理论分析,我们对其抽象,归结出更一般的情形:设移动节点 A, B, C, D 构成如图 3 所示的线形拓扑,有数据流从 C 到 D,则当 C 的发送队列总为非空、相应的 MAC 参数如表 1 所示时, A 到 B 的发送将以很大的概率出现 RET 事件. 下面,我们从理论上对其进行证明.



Fig.3 A linear topology constituted by mobile nodes A, B, C and D

图 3 由移动节点 A, B, C, D 构成的线形拓扑

不失一般性,设节点 3 从时刻 0 开始发送,节点 1 从时刻 $t_0(0 < t_0 < \text{节点 C 发完第 1 帧的结束时间})$ 开始发送,则根据 IEEE802.11 DCF 机制,可得节点 C 到 D 的发送和节点 A 的发送如图 4 所示.

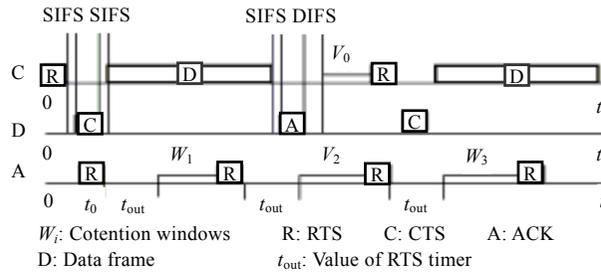


Fig.4 Node C, D and A send MAC frame

图 4 节点 C,D,A 的 MAC 帧发送

在图 4 中,设 $Y[n]$ 表示节点 A 发送第 n 个 RTS 帧的起始时刻的离散时间随机过程, $T[n]$ 表示节点 C 发送第 n 个 RTS 帧的起始时刻的离散时间随机过程,则

$$Y[n] = \begin{cases} t_0, & n = 1 \\ t_0 + (n-1) \times (t_{RTS} + t_{out}) + \sum_{i=1}^{n-1} W_i, & 1 < n \leq m \\ t_0 + (n-1) \times (t_{RTS} + t_{out}) + \sum_{i=1}^m W_i + (n-1-m) \times W_m, & m < n \leq R_{max} \end{cases}, T[n] = \begin{cases} 0, & n = 1 \\ (n-1) \times t_b + \sum_{j=1}^{n-1} W_0, & n > 1 \end{cases}$$

其中, R_{max} 为 RTS 帧的最大重传次数, W_i 为做第 i 次 RTS 重传尝试时的随机竞争窗口,它在 $[0, 2i \times w - 1]$ 之间均匀分布, t_{RTS} 为发送 RTS 帧所需的时间, m 为最大重传等级; $t_b = t_{RTS} + 3 \times t_{SIFS} + t_D + t_A + t_{DIFS}$, t_{SIFS} , t_{DIFS} , t_D , t_A 分别为 SIFS, DIFS, 发送数据帧和发送 ACK 帧的时间.

由图 4 可知,节点 A 发送 RTS 帧的第 n 次尝试取得成功的充分条件是此次发送在节点 C 的第 $i-1$ 次发送结束后开始,同时在节点 C 的第 i 次发送开始前 SIFS 结束;反之,如果节点 A 的第 n 次发送在节点 C 的第 $i-1$ 次发送结束后开始,同时在节点 C 的第 i 次发送开始前 SIFS 结束,则节点 A 的第 n 次发送一定能取得成功.因此,我们有节点 A 发送 RTS 帧的第 n 次尝试取得成功的充要条件为

$$T[i] - W_0 - t_{DIFS} < Y[n] < T[i] - t_{RTS} - t_{SIFS}, \forall i \in [N1, N2] \tag{4}$$

其中,

$$N1 = \left\lceil \frac{(n-1) \times (t_{RTS} + t_{out}) + t_0}{t_b + w} \right\rceil$$

$$N2 = \begin{cases} \left\lfloor \frac{(n-1) \times (t_{RTS} + t_{out}) + \sum_{k=1}^{n-1} 2^k w + t_0}{t_b} \right\rfloor, & n \leq m \\ \left\lfloor \frac{(n-1) \times (t_{RTS} + t_{out}) + \sum_{k=1}^m 2^k w + \sum_{k=m+1}^{n-1} 2^m w + t_0}{t_b} \right\rfloor, & m < n \leq R_{max} \end{cases}$$

$\{x\}$ 表示大于 x 的最小正整数; $[x]$ 表示小于 x 的最大正整数.对节点 A 发送 RTS 帧的第 n 次尝试, $N1, N2$ 分别给出了 i 的上界和下界.

节点 A 发生 RET 的概率即为节点 A 做 R_{max} 次尝试均失败的概率.根据式(4),即节点 A 第 n 次尝试成功的条件,我们可以求得节点 A 发生 RET 的概率 p_{RET} .在本文中,由于我们的主要目的是为了证明 p_{RET} 是一个相对较大的值,因此,我们省略了推导 p_{RET} 的过程.

显然,式(4)成立的必要条件是

$$W_0 + t_{\text{DIFS}} > t_{\text{RTS}} + t_{\text{SIFS}} \quad (5)$$

即只有在式(5)成立的条件下,节点 A 发送 RTS 的某次尝试才有可能成功;相反,只要式(5)不成立,节点 A 发送 RTS 的尝试均不会成功,在节点 A,事件 RET 必定发生.这样,我们有

$$p_{\text{RET}} \geq P\{W_0 + t_{\text{DIFS}} \leq t_{\text{RTS}} + t_{\text{SIFS}}\} = \frac{t_{\text{RTS}} + t_{\text{SIFS}} - t_{\text{DIFS}}}{w} \quad (6)$$

在式(6)中, w 的单位为时隙, $t_{\text{RTS}}, t_{\text{SIFS}}, t_{\text{DIFS}}$ 的单位为秒,因此,我们在计算式(6)时,需要将 $t_{\text{RTS}}, t_{\text{SIFS}}, t_{\text{DIFS}}$ 换算成时隙.根据表 1 所示仿真参数,我们可得 $p_{\text{RET}} \geq 0.5$.这样,我们便从理论上证明了在节点 A 确实会频繁地发生 RET 事件,导致节点 A 很难成功地发送数据帧.

根据上面的仿真和理论分析可知,导致无线自组织网络中的 TCP 流不稳定的原因可归结为两个:一是某节点长期占用无线信道(我们将其称为信道锁定),导致其他节点无法在规定的最大 RTS 重传次数内接入信道;二是如果节点在规定的重传次数内不能接入信道,即完成数据包到下一跳的转发,则该节点向数据包的源节点报告链路失效消息(而这是假的链路失效消息),从而触发漫长的路由发现过程,在这一过程中,源节点将停止数据包的发送.

2 对 IEEE 802.11 DCF 和路由协议的改进算法

针对 IEEE802.11 DCF 的不公平性和假链路失效问题,我们提出对 IEEE802.11 DCF 和路由协议的改进算法.

2.1 对 IEEE802.11 DCF 的改进

我们知道,在 IEEE802.11 DCF 中,为了避免某节点锁定信道,要求其在完成单个数据帧的发送之后,在 $[0, CW_{\text{min}} - 1]$ 之间随机回退.在无线局域网中——更确切地说,是在无线节点彼此能够相互侦听的网络环境中——这种机制是有效的.然而,在无线自组织网络中,这种机制将不能避免某节点对信道的锁定.针对这一问题,我们对 IEEE802.11 DCF 的改进算法如下:

一个移动节点在完成某个数据帧的发送之后,即收到接收方对该数据帧的确认之后,在发送下一数据帧之前,需在 $[0, \alpha \times CW_{\text{min}} - 1], \alpha \geq 1$ 之间进行随机延迟,而并非如 IEEE802.11 DCF 那样,在 $[0, CW_{\text{min}} - 1]$ 之间进行随机延迟.在本文中,我们将 σ 称为竞争窗口系数.这样,通过适当增大 σ 值来增大移动节点在连续发送两次数据帧之间的随机时延,从而增大其他移动节点接入信道的概率,并在很大程度上减小移动节点连续 7 次发送尝试失败的概率.

显然,不同的 σ 值将对 TCP 流的稳定性有不同的影响.在本文第 3 节将要介绍的性能仿真中,我们将对不同 σ 值进行仿真,并给出 σ 的建议值.

2.2 对路由协议的改进

通过本文第 2.1 节对 IEEE802.11 DCF 的改进,我们可以在很大程度上减小无线自组织网络中移动节点发生 RET 事件的概率,但不能完全避免该事件的发生,一旦发生该事件,对于目前已有的路由协议,该移动节点将向数据包的源节点发送路由差错消息,并在源节点触发漫长的路由发现过程.

对此,我们提出对路由协议的改进算法如下:

当某节点的路由协议收到来自链路层的到下一跳的链路失效指示时,并不马上向数据包的源节点发送路由差错消息,而是向对应的下一跳发送 Hello 消息,该消息的目的地址就是下一跳的地址,同时设置定时器,如果在定时器超时之前没有收到来自下一跳对该 Hello 消息的应答,则重传 Hello 消息.如果定时器的超时次数达到预设的最大次数 C_{max} 仍未收到来自下一跳的应答消息,则判定到下一跳的链路失效,并向源节点发送路由差错消息.如果定时器的超时次数在达到设置的最大次数前收到来自下一跳的应答消息,则复位定时器,取消向源节点发送路由差错消息,并继续向该下一跳转发数据包.在本文中,我们将以上过程称为链路失效判定过程.它能有效排除假的链路失效,从而避免不必要的路由发现过程.在链路失效判定过程中,移动节点缓存需转发到对应下一跳的数据包,直到对应的判定过程结束为止.如果通过判定过程确定链路失效,则丢弃这些缓存的包,否

则,将它们转发到下一跳.

对 Hello 消息的最大重传次数 C_{\max} ,我们将从尽量改进无线自组织网络的 TCP 流稳定性出发,在本文第 3 节的性能仿真中给出其建议值.

顺便指出,在无线自组织网络中,由于几乎所有路由协议都需要响应来自链路层的链路失效指示,并触发复杂的路由计算和路由发现过程,而上述链路失效判定过程能区分出假链路失效,从而避免不必要的路由计算和路由发现过程,因此,从这个意义上说,链路失效判定过程对其他类型的路由协议也是有用的.

3 性能仿真

在本节中,我们将通过仿真对本文第 2 节提出的 IEEE802.11 DCF 的改进算法和路由协议的改进算法进行评估,并给出竞争窗口系数和 Hello 消息的最大重传次数的建议值.

在仿真时,我们仍采用 NS-2,仿真拓扑如图 1 所示,对应的仿真参数如表 1 所示.为方便改进前的 IEEE802.11 DCF 和路由协议进行对比,我们仍采用本文第 1 节建立的 TCP 连接,并用相同的方法统计 TCP 流的吞吐量.

首先,我们取 $C_{\max}=0$,对不同的 σ 值分析 TCP 流的稳定性.这等效于未对路由协议作任何改进,而仅对 IEEE802.11 DCF 作改进.在 $\sigma=8,16,24,32$ 时,对应的节点 1~节点 5 的 TCP 流的吞吐量和 120s 内的平均吞吐量分别如图 5 和表 4(对应 $C_{\max}=0$ 的行)所示.

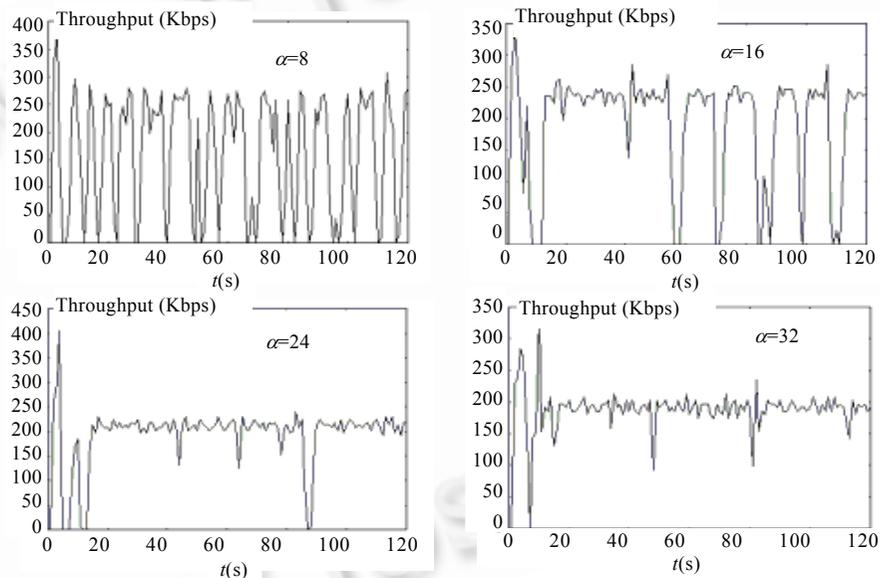


Fig.5 Throughput of TCP flow from node 1 to node 5, with $C_{\max}=0$ and different σ

图 5 $C_{\max}=0$ 时对不同的 σ ,节点 1~节点 5 的 TCP 流的吞吐量

将图 5 和图 1 作比较,再根据表 4 的仿真结果可知,通过对 IEEE802.11 DCF 的改进,适当增加 σ 的值,不仅能在很大程度上改进无线自组织网络中 TCP 流的稳定性,同时能在较大程度上提高 TCP 流的平均吞吐量.当 σ 大于 16 时,过大的 σ 值已对改进 TCP 流的稳定性没有多大意义,相反,会在一定程度上降低 TCP 流的平均吞吐量.我们对出现这样的结果的原因分析如下:适当地逐渐增大 σ 的值,可逐渐减小 TCP 连接上的节点发生 RET 事件的概率,从本文第 2 节的分析可知,这样能逐渐改进 TCP 流的稳定性,减少 TCP 的重传超时次数.显然,这对提高 TCP 流的平均吞吐量也是有利的.当 σ 达到一定数值时,TCP 连接上的节点发生 RET 事件的概率已趋近于 0,再增大 σ 的值已对改进 TCP 流的稳定性没有多大意义,相反地,过大的 σ 值会延迟 MAC 数据帧的发送时间,从而引起 TCP 流的平均吞吐量的降低.

其次,我们取 $\sigma=1$,对不同的 C_{\max} 进行仿真.这等效于未对 IEEE802.11 DCF 作任何改进,而仅对路由协议作改进.在 $C_{\max}=1,2,3$ 时,对应的节点 1~节点 5 的 TCP 流的吞吐量和 120s 内的平均吞吐量分别如图 6 和表 4(对

应 $\sigma=1$ 的行)所示.

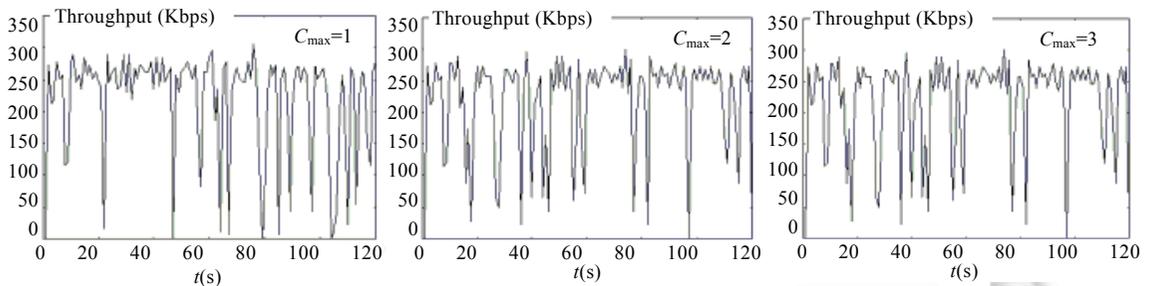


Fig.6 Throughput of TCP flow from node 1 to node 5, with $\sigma=1$ and different C_{max}

图 6 $\sigma=1$ 时,对于不同的 C_{max} ,节点 1~节点 5 的 TCP 流的吞吐量

Table 4 The average throughput of TCP flow from node 4 to node 5 during 120s, with different σ and C_{max}

表 4 在 C_{max} 和 σ 的不同取值下,节点 1~节点 5 的 TCP 流在 120s 内的平均吞吐量

C_{max}	σ	Average throughput (Kbps)	C_{max}	σ	Average throughput (Kbps)
0	1	120	3	1	219
0	8	161	1	8	248
0	16	189	1	16	210
0	24	194	1	24	203
0	32	188	2	16	230
1	1	218	2	24	205
2	1	219			

将图 6 和图 1 作比较,再根据表 4 的仿真结果可知,通过对无线自组织网络中路由协议的改进,我们能在一定程度上改进 TCP 流的稳定性,并在很大程度上增加 TCP 流的平均吞吐量.只要 C_{max} 不等于 0,它都能达到改进 TCP 性能的目的.当 C_{max} 的值为 2 和 3 时,我们得到相同的仿真结果,这说明,最多通过对 Hello 消息的 2 次~3 次重传,我们就能判断出这种假的链路失效.

进一步地,由图 5、图 6 和表 4 的仿真结果可知,通过对 IEEE802.11 DCF 的改进,我们能基本上解决 TCP 流的不稳定性问题,通过对路由协议的改进,我们能在很大程度上提高 TCP 流的平均吞吐量,因此,可以想象,如果我们同时对两者进行改进,并选择适当的 σ 和 C_{max} ,将能达到既改进 TCP 流的稳定性,又提高 TCP 流的平均吞吐量的目的.图 7 的仿真结果证明了这一点.

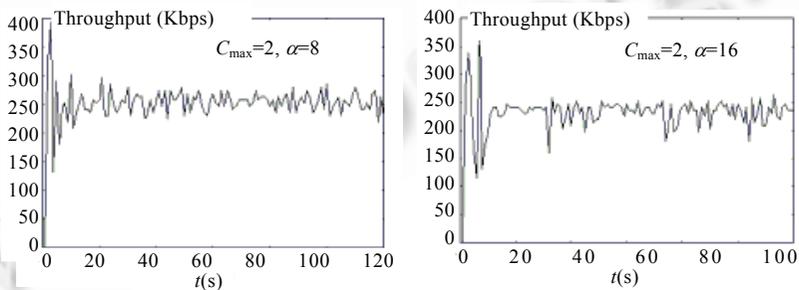


Fig.7 Throughput of TCP flow from node 1 to node 5, with different σ and C_{max}

图 7 对不同的 σ 和 C_{max} ,节点 1~节点 5 的 TCP 流的吞吐量

根据以上的仿真结果,我们给出 σ 和 C_{max} 的建议值:

$$\begin{aligned} 8 \leq \sigma \leq 16, \\ 2 \leq C_{max} \leq 3. \end{aligned}$$

对于如图 3 所示的线形拓扑, σ 越大,节点 C 连续发送 MAC 数据帧之间的时间间隔越大,节点 A 成功接入无线信道的概率越大,但正如表 4 所示,过大的 σ 会减小 TCP 流的吞吐量,从而降低系统的性能, σ 的建议值[8,16]正是两者折衷的结果.从图 7 和表 4 可知, σ 可在较大的范围[8,16]内变化而不明显影响 TCP 流的稳定性和吞吐量.

另外,我们对 IEEE802.11 DCF 的改进只是简单地增加节点连续发送数据帧之间的随机时延,不涉及对帧类型的增减、帧结构和收发双方协议的修改,其实现简单,且能同原 IEEE802.11 兼容。

4 结束语

本文对无线自组织网络中 TCP 流的稳定性进行了深入的分析,找出了造成 TCP 流不稳定性的根源,并在此基础上,提出了一种针对 IEEE802.11 DCF 和路由协议的改进算法,通过仿真分析证明,相应的改进算法不仅能基本上解决 TCP 流的不稳定性问题,还在很大程度上提高了 TCP 流的平均吞吐量。

References:

- [1] Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In: Proceedings of SIGCOMM'94. New York: ACM Press, 1994. 234~244.
- [2] Perkins C, Royer E, Das S. Ad Hoc on demand distance vector (AODV) routing (work in progress). Internet Draft, Internet Engineering Task Force, 2003. <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt>.
- [3] Johnson D, Maltz D, Hu Y, Jetcheva J. The dynamic source routing protocol (DSR) for mobile Ad Hoc networks (work in progress). Internet draft, 2003. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>.
- [4] Broch J, Maltz DA, Johnson DB, Hu Y-C, Jetcheva J. A performance comparison of multi-hop wireless Ad Hoc network routing protocols. In: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking. New York: ACM Press, 1998. 85~97.
- [5] Bharghavan V, Demers A, Shenker S, Zhang L. MACAW: A media access protocol for wireless LANs. In: Proceedings of the ACM SIGCOMM'94. New York: ACM Press, 1994. 212~225.
- [6] IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications. ANSI/IEEE Std 802.11, 1999.
- [7] Xu SG, Saadawi T. Does the IEEE802.11 MAC protocol works well in multi-hop wireless Ad Hoc networks? IEEE Communications Magazine, 2001,39(6):130~137.
- [8] The networks simulator ns-2. 2003. http://www.isi.edu/nsnam/ns/ns_doc.pdf.
- [9] Floyd S, Henderson T. The newreno modification to TCP's fast recovery algorithm. Technique Report, RFC 2582, Internet Engineering Task Force, 1999.
- [10] Binachi G. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications, 2000,18(3):535~547.