

无线自组织网络中的 IEEE802.11 MAC 协议的研究

李春时^{1,2} 王光兴¹

(东北大学信息科学与工程学院 沈阳 110004)¹ (大连民族学院计算机学院 大连 116600)²

摘要 IEEE802.11 DCF 作为无线局域网的 MAC 层规范被广泛应用于无线自组织网络的仿真和测试中。通过研究发现即使在拓扑不变的条件下,使用表驱动路由协议 DSDV, TCP 流仍然存在不稳定和不公平现象,本文分析了问题产生的原因,提出了基于网络负载的解决问题方法。

关键词 无线自组织网络, IEEE 802.11 DCF 协议, TCP 流, 稳定性, 公平性

A Survey of the IEEE 802.11 MAC Protocol in Wireless Ad Hoc Networks

LI Chun-Shi^{1,2} WANG Guang-Xing¹

(Faculty of Information Science and Engineering of Northeastern University, Shenyang 110004)¹

(Faculty of Computer Science of Dalian University Technology, Dalian 116600)²

Abstract The IEEE 802.11 MAC protocol is the standard for wireless LANs, it is widely used in test-beds and simulations for wireless multi-hop ad hoc networks. Although it can support some ad hoc network architecture, it is not intended to support the wireless mobile ad hoc network. The simulations show that, given a specific network topology, using table driven router protocol DSDV, there still exist unfairness problem and instability problem among the TCP flows. In this article we reveal the causes of the problem and offer the possible solutions based on network overload.

Keywords Wireless Ad Hoc networks, IEEE 802.11 DCF protocol, TCP flow, Stability, Fairness

1 引言

MANETs (mobile ad hoc networks) 是一个复杂的分布式系统,由无线自由移动的节点组成,没有地面基础设施,可以临时搭建,具有自组织、多跳和抗摧毁的特点,被广泛应用于作战、救灾等紧急情况。

无线自组织网络与有线网络、蜂窝无线网络相比具有动态拓扑、链路带宽受限、容量时变、动力受限、物理上的安全有限和没有诸如基站的基础设施等特点,但所有这些不同都可以归结为节点的移动性和传输介质的特殊性,这使得原有的网络的路由协议和媒体访问控制协议在无线自组织网络中不再适用,从而成为研究的热点。本文研究了 IEEE802.11 DCF^[1] 应用于无线自组织网络中存在的问题,分析了产生问题的根本原因,提出了解决问题的可行方法。

2 无线自组织网络现有的协议以及存在的问题

TCP 是一个流行的传输控制协议,提供可靠的数据传输,具有拥塞控制能力和适应的带宽调整功能,所以被 MANETs 用作 TCP 传输层协议,运行在网络层和 MAC 层之上。IEEE 802.11 DCF 协议作为无线局域网的 MAC 层协议被广泛用于无线自组织网络的试验和仿真中,成为无线自组织网络事实上的 MAC 层规范。通过仿真发现基于 IEEE802.11 DCF 的 TCP 流之间存在严重的不稳定和公平问题,分析了产生这些问题的根本原因在于 IEEE802.11 DCF 本身。

2.1 运行在 IEEE802.11 DCF 基础上的 TCP 流的不稳定问题

IEEE 802.11 DCF^[1] 的基本的访问机制是 CSMA/CA, 采用二进制指数退避的方式。为了克服隐藏终端问题,协议提供了 RTS/CTS/DATA/ACK 四次握手的方式。一个节点要发送数据帧,在 $[0, CW_{min}]$ 随机选取退避时间。侦听信道,如果信道忙,则继续侦听;如果信道空闲,则退避计时器开始计时,减少到 0 时发送 RTS,如果发生冲突,则在 $[0, 2^m CW_{min}]$ 随机重新选取退避时间,其中 m 为最大回退等级。RTS 包括源、目的地址和数据传送的时间。目的节点收到 RTS,如果信道空闲,目的节点发送 CTS 分组,包含同样的时间信息。所有其它的邻居节点收到 RTS 或者 CTS 设置 CS 指示器,这样锁定了信道,降低了冲突的可能性。源节点收到 CTS 后发送 DATA,目的节点 CRC 检验证实收到的 DATA 包正确,然后发送应答包 ACK。接收方收到 ACK 表示发送成功。如果发送方没有收到 ACK,则重传该帧直到收到 ACK 或一定次数重传之后丢掉该帧。同时报告路由失败。如果发送方发送 RTS 7 次不能收到 CTS,则丢掉该帧并报告链路断连。

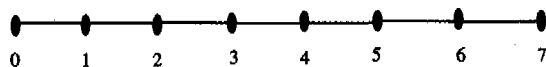


图 1 线性拓扑

采用 LBNL 开发的仿真平台 NS-2^[2], 在试验和仿真过程中假设节点是静止的,因为相对于信号的传送来说节点的运动速度是很小的,采用 DSDV(destination-sequenced distance-vector routing)^[3] 路由协议,物理层和数据链路层采用 IEEE 802.11 DCF, TCP 版本为 Newreno。仿真拓扑如图 1 所示,相邻节点间的距离为 150m,线性拓扑是无线自组织网络中的

典型连接。

2.2 运行在 IEEE802.11 MAC 基础上的 TCP 流的不稳定问题

实验 1 在节点 0 和节点 7 之间建立一条 TCP 连接,每隔 1s 计算 TCP 的平均吞吐量,结果如图所示。由于使用的是 DSDV 路由协议,从开始到 93s 这段时间发现路由,吞吐量为 0。由图 2 可以看出在 93s 到 180s 这段时间里 TCP 流严重的不稳定,并有几次吞吐量为 0,这说明在相应的 1s 内,成功发送的 TCP 包数量为 0。

造成 TCP 流不稳定的原因在于暴露节点长时间不能获得信道而丢弃数据包。暴露终端是无线多跳网络固有的特性。由于无线多跳网络的丢包率始终处在低水平,因此 TCP 的发送窗口总是很大,在稳定状态下,数据包均匀的分布在链路节点的缓冲区内,使得链路上各个节点的队列非空,这样不可避免地造成竞争丢包^[4]。

文[5]针对于发送节点锁定信道的情况提出了改进的办法,如果移动节点完成一个数据帧的发送,发送下一个数据帧之前,需在 $[0, \alpha CW_{min}-1]$, $\alpha > 1$ 之间进行随机延迟,这样增大了其它节点接入信道的机会。但是因为数据分组远比控制分组大,仍然会出现多次发送 RTS 都不能接入信道的情况,并且采用二进制指数退避的方式,在 $[0, 2^m CW_{min}-1]$ 之间随机选取,仍然有利于最后发送成功的节点,并且 α 的取值范围很难确定,如果 α 的取值过大,低负载的情况下降低了网络的效率,如果 α 的取值偏小,网络负载高的情况下仍然提高不了公平性,且文[5]只给出了一个仿真的估计值,不具有普适性。

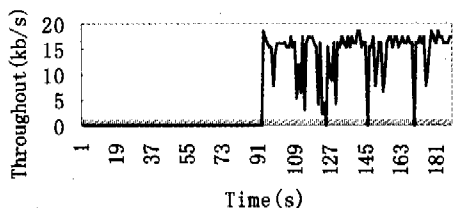


图 2 TCP 流的吞吐量

2.3 运行在 IEEE802.11 DCF 基础上的 TCP 流间的不公平问题

实验 2 我们在节点 1~3 和节点 4~5 之间建立两个 TCP 连接,节点 1~3 之间 FTP 流从 10s 开始,4~5 之间的 FTP 流从 30s 开始。每隔 1s 计算 TCP 的平均吞吐量,结果如图 3 所示。由图 3 可以看出,1~3 为 2 跳,从 38s 开始吞吐量为 44kb/s,然而到第 40s 时节点 5 开始发送包,1~3 的吞吐量完全下降为 0,不再有重新开始的机会。这时的吞吐量完全属于 4~5,约 88kb/s。1~3 即使首先接入信道也会完全丧失带宽。节点 4-5 锁定了无线信道,造成严重的不公平。

解释为什么发生这种情况。表 1 列出了实验 2 的部分 trace 文件,从 40s 开始节点 1 到节点 3 的 TCP 分组不能从节点 2 成功到达节点 3。发送 7 次 RTS 仍然不能获得信道,则丢弃该数据包,节点 1 超时重传 TCP 分组,分组排列在节点 1 的队列中等待转发。这就是为什么一些 TCP 分组到期被丢弃。不幸的是,40s 后一些 TCP 分组根本不能到达节点 3。通过分析仿真结果,发现问题的根源在于 MAC 层。节点 2 不能到达节点 3 的主要原因是冲突。既然节点 3 能检测到节点 4,当节点 4 发送时节点 3 就不得不推迟。结果是尽管正

确的从节点 2 收到了 RTS 节点 3 不能返回 CTS。然而,节点 4 到 5 只有一跳。节点 4 从 5 收到 ACK 后,发送 RTS 请求分配信道,准备发送其它的 TCP 分组。一旦节点 5 收到 RTS 并且应答 CTS,节点 4 开始发送 TCP 分组。正常情况下,数据分组比控制分组大。一旦节点 4~5 开始通信锁定了信道,节点 3 在这期间设置了 CS 指示器,不能应答节点 2 的 RTS,因此只有在节点 4 发送 RTS 之前节点 3 获得机会发送 CTS,这必须是在节点 4 发送完数据分组收到 ACK 之后。节点 3 访问信道的的时间很少,因为二进制指数退避机制总是有利于最后成功的节点(节点 4 就是如此)。节点 3 收到节点 2 的 RTS 后几乎不能获得信道,使得 1~3 的 TCP 流的吞吐量为 0。

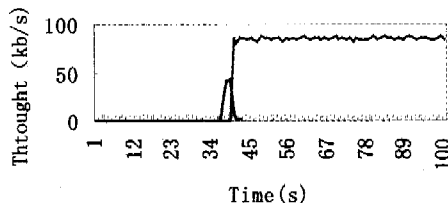


图 3 2 个 TCP 流的吞吐量对比

表 1 实验 2 部分 trace 文件

Time (s)	Events	Time (s)	Events
40.00287	node 2 send RTS to node 3	40.00323	node 3 received CTS
40.00805	node 4 send RTS to node 5	40.00841	node 5 send CTS to 4
40.00849	node 2 send RTS to 3	40.00872	node 4 send TCP to node 5
40.00885	node 3 received CTS	40.00975	node 2 send RTS to node 3
40.00962	node 5 received TCP	40.00963	node 5 send ACK to node 4
40.00993	node 3 received CTS	40.00993	node 4 received ACK
40.1242	node 4 send RTS to 5	40.01278	node 5 send CTS to 4
40.01291	node 2 send RTS to 3	40.01309	node 4 received CTS
40.0131	node 4 send TCP to node 5	40.01327	node 3 received CTS
40.1611	node 2 send RTS to 3	40.01678	node 2 RET and dropped packet

引起这种严重的不公平现象的根本原因在于 IEEE802.11 DCF 协议本身,采用二进制指数退避的目的是为了减少再次冲突的概率,文[5]已经证明由连续的冲突导致的丢包的概率为 1.3×10^{-3} ,二进制指数退避总是有利于最后发送成功的节点,造成了信道的锁定,使得其它节点难以接入信道。以严重的数据流不公平为代价换取冲突避免,而由连续冲突导致的丢包的概率又如此小,可见简单的二进制指数退避在网络 QoS 的收益上是用西瓜换芝麻。

文[6]提出了基于 2 个退避区间的算法,设定标准退避区间为帧长的三分之一,即一个帧的发送时间为 $\frac{packetlength}{3 \text{ 比特率}} = 6ms$,发送节点按照 $CW_{slot} = 2ms, 2CW_{slot} = 4ms$ 循环退避,如图 4 所示,节点无论在什么时间接入信道,都能在 7 次之内成功发送 RTS。在网络负载低的情况下避免了 RET 现象,有效改善了不稳定和不公平的问题。根据文[6]中对 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

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

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

$$\tau = \frac{2(1-2p)}{(1-2)(w+1)+pw(1-(2p)^m)}, m \text{ 为最大回退等级, 在没有指数退避的情况下 } m \text{ 为 } 0,$$

$$\tau = \frac{2}{w+1} \quad (2)$$

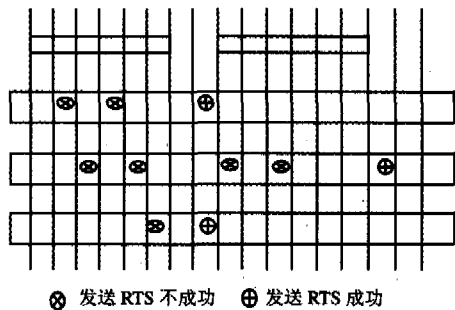


图 4 方法的图示描述

在文[6]的仿真条件下, $CW_{slot} = 2ms, slot = 10\mu s, w = 200$, 代入(2)式和(1)式计算得, 当 $n = 8$ 时, $p = 0.08$; 当 $n = 16$ 时, $p = 0.15$, 可以看出当网络负载较高时, 任一节点在任一时隙发送的包出现碰撞的概率是很高的, 尤其对于随机的拓扑结构。

上述的解决办法都没有考虑在时变的条件下网络负载的变化, 所以不具有普适性。

3 对 IEEE 802.11 DCF 的改进办法

文[4]通过分析和仿真发现, 对于给定的网络拓扑和流量模型, 存在最优的 TCP 发送窗口 w^* , 这时信道的利用率最高, TCP 的吞吐量最大。继续增大发送窗口降低了信道的重用率, 由于信道的竞争反而导致数据包的丢失。另外, 在发送窗口没有约束的情况下, 因为无线信道的丢包特性, 标准的 TCP 协议中平均窗口 \bar{w} 总是大于 w^* 。这使得链路中每个节点的发送队列均为非空, 根据上面的计算出现包碰撞的概率较高, 尤其对于随机的拓扑结构。

产生问题的根本原因在于无线多跳网络的低丢包率引发的 TCP 大的发送窗口。文[3]通过分析和仿真发现, 对于给定的网络拓扑和流量模型, 存在最优的 TCP 发送窗口 w^* , 这时信道的利用率最高, TCP 的吞吐量最大。继续增大发送窗口降低了信道的重用率, 由于信道的竞争反而导致数据包的丢失, TCP 的吞吐量下降。

假设一个 h 跳的链路, 节点间距为 l , 对于图 1 所示的链路, 由一个节点向下一个相邻节点发送数据时, 只有距离接收节点 $550m$ 以外才能同时成功发送数据, 即两节点相距 $h' = \lfloor \frac{550}{l} \rfloor + 1$ 跳时能同时成功发送数据, 这时带宽利用率最高, $l = 150m, h' = 5$ 。在稳定条件下, 数据分组在各个节点上均匀分布, 所以当 $w^* = \frac{h}{5}$ 时链路的利用率最高, TCP 的吞吐量最大。但是在发送窗口没有约束的情况下, \bar{w} 总是超出 w^* 很多。

$$P_l(m) = \frac{b(m)}{m} \cdot (1 - \frac{b(m)}{c(m)})^r \quad (3)$$

$$P(m) = 1 - (1 - P_l(m))^{c(m)} \quad (4)$$

其中 m 为链路上的节点数, $b(m)$ 为能同时成功发送数据包的最大节点数, $c(m)$ 为能同时发送 RTS 的最大节点数。

原因是在发送窗口没有约束的情况下, 根据式(3)和式(4)计算的丢包率总是很小, 使得 TCP 的平均发送窗口 \bar{w} 总是大于 w^* 。这使得链路中每个节点的发送队列均为非空, 出现了竞争丢包, 造成 TCP 流的不稳定, 而二进制指数退避总是有利于发送成功的节点, 产生 TCP 流间的不公平性。

解决问题的办法是根据信道的负载适当增大丢包率, 从而降低 TCP 的发送窗口, 减小因信道竞争引起的冲突, 提高信道利用率。在 MAC 层计算 RTS 的平均重传次数 avg-retry 以此来反映网络的负载, 当 avg-retry 小于某个门限值 retry_th 时, 网络的负载较低, 信道处于不饱和状态, 执行原来退避算法, 保证信道的吞吐量; 当 avg-retry 大于 retry_th 时, 这时网络的负载较高, 设定包被丢弃的概率为原来的二倍, 丢包率取 p 和 p -th 中较小的, 这样减小 TCP 发送窗口, 降低网络负载。发送成功的节点退避时间在原有的基础上增加一个帧长的时间, 避免发送成功的节点对于信道的锁定。算法的伪代码如图 5。

```

Require: avg-retry is the average RTS retries for
each pocket
if avg-retry < retry_th then
    p ← p
    back off ← random_backoff
else p ← 2p
    p ← min{p, p-th}
    if received ACK then
        back off ← random_backoff + TX_Time(DATA)
    end if
end if
MacLayerSend(p)
retry = GetMacRetriesRTS()
avg-retry = 2avg-retry/3 + retry/3
    
```

图 5 算法的伪代码

结束语 本文分析了基于 IEEE802.11 DCF 协议的无线自组织网络中 TCP 流的不稳定性和不公平性, 指出了产生问题的最根本原因在于无线信道不同于有线网络的丢包特性。由于丢包率低使得信道总是稳定于饱和状态(即链路上节点的队列为非空), 信道的竞争导致了丢包和信道锁定, 产生不稳定和不公平现象, 指出了已有方法的局限性, 提出了自适应的基于网络负载的解决问题方法。

参考文献

- 1 IEEE Std. 802.11. Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications, 1999
- 2 The networks simulator ns-2. 2003. http://www.isi.edu/nsnam/ns/ns_doc.pdf
- 3 Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In: Proceedings of SIGCOMM 4. New York: ACM Press, 1994. 234~244
- 4 Fu Zhenghua, et al. TCP over Multihop Wireless Networks; [UCLA Computer Science Tech. Rep]. www.cs.ucla.edu/wing/publication/tech010702.ps
- 5 Li Yun, Chen Qianbin, et al. Analyzing and Improving the TCP Stability in Wireless Ad Hoc Networks. Journal of Software, 2003, 14(6)
- 6 Kanth T K, Ansari S, Mehkri M H. Performance Enhancement of TCP on Multi-hop Ad Hoc Wireless Networks