

Ad Hoc 多跳网络的不公平性研究*

苏爱国, 董虹宇, 张光昭

(中山大学电子与通信工程系, 广东 广州 510275)

摘要: 在分析基于 802.11 的 Ad Hoc 多跳网络中存在的 TCP 不公平性问题及其产生原因的基础上, 提出了一种新的方案——DWDB (double windows dynamic backoff) 算法, 对基本 802.11 MAC 协议进行了退避算法和重传机制的修改。仿真结果表明该方案有效地解决了不公平性问题。

关键词: Ad Hoc 网络; TCP 协议; DWDB 算法

中图分类号: TP393.17 **文献标识码:** A **文章编号:** 0529-6579 (2004) 01-0032-05

Ad Hoc 网络是一种无中心的网络, 其最关键的就是数据包的可靠传输和拥塞控制, 这些大多是由 TCP 协议支持的。TCP 是 Internet 协议簇中传输层的重要部分, 它在不可靠的下层网络的基础上提供面向连接的可靠的服务。

遗憾的是 TCP 是针对有线网络设计的, 它并不能很好地运行在 Ad Hoc 多跳网络上。由于 Ad Hoc 网络主要使用 IEEE 802.11 作为底层协议, 而且存在着隐藏节点和暴露节点等问题。因此, 当 TCP 在 Ad Hoc 上运行时, 将会出现不稳定和不公平等问题, 严重影响了数据传输质量^[1,2]。不稳定问题可以通过对 TCP 协议优化得到改善, 但不公平性问题却不能因此得到解决, 因此不公平性问题是当前一个研究热点^[3-5]。本文重点是研究多跳网络下 TCP 的公平性能, 选取的研究对象是静态多跳 Ad Hoc 网络, 即暂不考虑 Ad Hoc 网络拓扑的动态变化。

1 不公平性分析

1.1 基本 802.11 的 BEB 退避算法

IEEE 802.11 的 MAC 层包括两种不同的访问方式, 分布式访问控制方式 DCF (distributed coordinator function) 和集中式访问控制方式 PCF (point coordinator function)。DCF 使用载波监听多路访问/冲突避免 (carrier sense multiple access with collision avoidance, CSMA/CA) 来访问共享信道, PCF 则使用集中式的、基于轮询的机制。DCF 是 802.11 MAC 协议的基本媒体访问方法, 而 PCF 由于需要中心控制器, 不像 DCF 那么灵活, 绝大多

数的网卡并不支持 PCF, 本文主要研究 DCF 方式下竞争信道的问题。

当有数据帧需要发送时, MAC 层监听信道是否处于空闲状态。如果信道空闲且空闲期间大于 DCF 帧间间隔 DIFS (DCF interframe space), 则立即发送数据帧, 否则继续监听信道直到检测到需要的空闲帧间隔, 然后选择随机退避时间进入退避, 退避时间结束后如果信道空闲则立即发送, 否则重新开始上述的坚持监听过程, 并按退避算法选择退避时间。

基本 802.11 采用二进制指数退避算法 BEB (Binary Exponential Backoff), 其随机退避时间是用下面的公式计算得到的^[6]:

$$\text{Backoff time} = \text{Random}(\cdot) \times \text{aSlotTime} \quad (1)$$

其中, $\text{Random}(\cdot)$ 是一个平均分布在 $[0, CW]$ 段上的伪随机整数, CW (竞争窗口) 介于管理信息库 (MIB) 中 aCWmin (CW 最小值) 和 aCWmax (CW 最大值) 之间的一个整数, 取自这个范围的随机数是独立于各个节点的。 aSlotTime 由物理层定义的。 CW 的初始值等于 aCWmin 。

退避算法提供 Finc 和 Fdec 两个函数对 CW 进行计算:

$$\begin{aligned} \text{Finc} &= \min \{2 \times CW - 1, \text{aCWmax}\} \\ \text{Fdec} &= \text{aCWmin} \end{aligned} \quad (2)$$

在发送数据帧时使用了 RTS-CTS-DATA-ACK 机制, 源节点在退避时间计时器结束以后, 等待 DIFS 后发送 RTS 帧, 通知目标节点接收数据帧。目标节点收到 RTS 帧后等待 SIFS (Short InterFrame Space, 短帧间间隔) 后发送 CTS 帧, 收到 RTS 或

* 收稿日期: 2003-07-15

基金项目: 国防预研基金资助项目

作者简介: 苏爱国 (1971 年生), 男, 博士研究生, 工程师; 通讯联系人: 张光昭; E-mail: isde04@zsu.edu.cn

CTS 帧的其他节点则启动虚拟载波检测, 在 RTS 或 CTS 指定的时间内保持静默。源节点收到 CTS 帧以后等待 SIFS 则开始发送数据帧, 目标节点在收到数据帧后等待 SIFS 则回复 ACK 帧。竞争窗口在 ACK 帧结束以后经过 DIFS 后启动, 开始下一轮的信道竞争。每次 RTS-CTS 对话失败, 帧没有发送成功, 对退避计数器执行 Fine 操作; 而每次 RTS-CTS 对话成功, 对退避计数器执行 Fdec 操作。为了控制重发次数, MAC 设定了最大短帧重传次数 $\text{dot11ShortRetryLimit}$ 和最大长帧重传次数 aLongRetryLimit 的值, 如果短帧重发次数超过了 $\text{dot11ShortRetryLimit}$ 或者长帧重发次数超过了 aLongRetryLimit , 则丢弃该帧, 并向上层报告错误。

1.2 多跳网络的不公平性问题分析

首先用仿真实验来看一下 Ad Hoc 多跳网络中存在的 TCP 流的不公平性问题。本文采用工作于 Linux 平台上的 ns2 仿真环境, 多跳 Ad Hoc 仿真网络拓扑如图 1 所示, 使用 TCP Sack1 协议和动态源路由协议 DSR (dynamic source routing), 链路层采用 IEEE 802.11 标准 MAC 协议 DCF, 所有节点都是半双工, 带宽为 1 Mb/s, 传送半径为 250 m, 网络由 8 个节点组成, 相邻两节点间距离为 200 m, 即每个节点只能直接和相邻节点通信。而且只有源到目的节点间的 TCP session, 没有其他的背景负荷, 即不存在网络拥塞问题。ns2 中相关参数的默认值如下: TCP 的最大通告窗口值 window- 为 4, TCP 分组数据包大小是 1 460 bytes, 初始窗口值为 2。

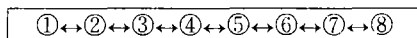
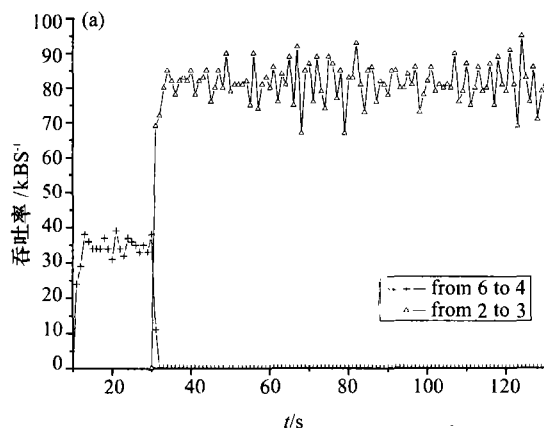


图 1 仿真网络拓扑结构

Fig.1 The simulation topology

在同样的 7 跳仿真网络中, 先后产生两个 TCP 流, 如图 2(a)所示, 先产生的 2 跳(从节点 6 到 4)



TCP 流在第 2 个数据流产生前保持稳定吞吐率, 但当节点 2 到 3 的 1 跳 TCP 数据流连接建立起来后, 虽然两个数据流不占用同一个信道, 但是第一个 TCP 数据流吞吐量却迅速降到 0, 只剩下第 2 个数据流保持着稳定的吞吐率。

分析表明, 在 20 s 时节点 2 开始发送数据包给节点 3, 这时隐藏节点问题出现了 (节点 4 是隐藏节点), 节点 5 不能察觉到节点 2 和节点 3 正在通信, 于是它发送 RTS 给节点 4 请求发送数据, 节点 4 因为之前收到了节点 3 发出的 CTS 知道此时信道不空闲, 所以节点 4 不返回 CTS 给节点 5, 节点 5 在规定时间内没有收到节点 4 的 CTS 后, 便调用后退算法, 重新计算 CW, 选择一个随机退避时间等待。由于 MAC 的二进制指数后退算法总是优先考虑最近成功发送的节点, 因此节点 5 很难继续获得发送数据帧给节点 4 的机会。在超过 MAC 最大重传次数后, MAC 层将 5 到 4 的链路中断信息报告给上层, 并通知源节点 6 链路失败。节点 6 将启动路由发现机制寻找新的路由, 但是由于同样的原因节点 5 很难发送 DSR 路由报文给节点 4, 节点 6 也就不能找到路由。最后, TCP 连接被迫中断, 信道完全被第二条 TCP 连接占据。当然, 当节点 5 发送数据帧给节点 4 时节点 2 也是一个隐藏节点, 但是由于节点 2 和 3 的 TCP 连接不需要中间节点转发数据包, 发送数据帧的频率比两跳的 TCP 连接高, 获得信道的机会也远高于第一条, 因此就出现了不公平现象。

实验发现, 这种不公平性不仅出现在短跳数流和长跳数流之间, 而且对于同样的跳数, 如图 3 (a) 所示, 先建立的流 (从节点 4 到 6) 比后建立的流 (从节点 1 到 3) 也具有绝对的优先级, 后建立的流完全没有机会成功发送。无法通过改变 TCP 参数或 MAC 重传次数来解决这个问题。

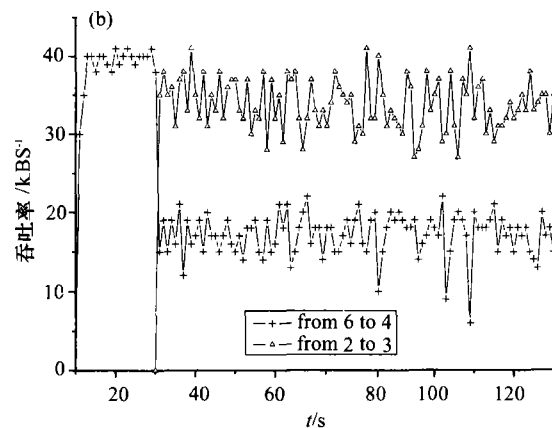


图 2 不同跳数 TCP 流的不公平性改善

Fig.2 The unfairness's improvement of the different hop's TCP connections

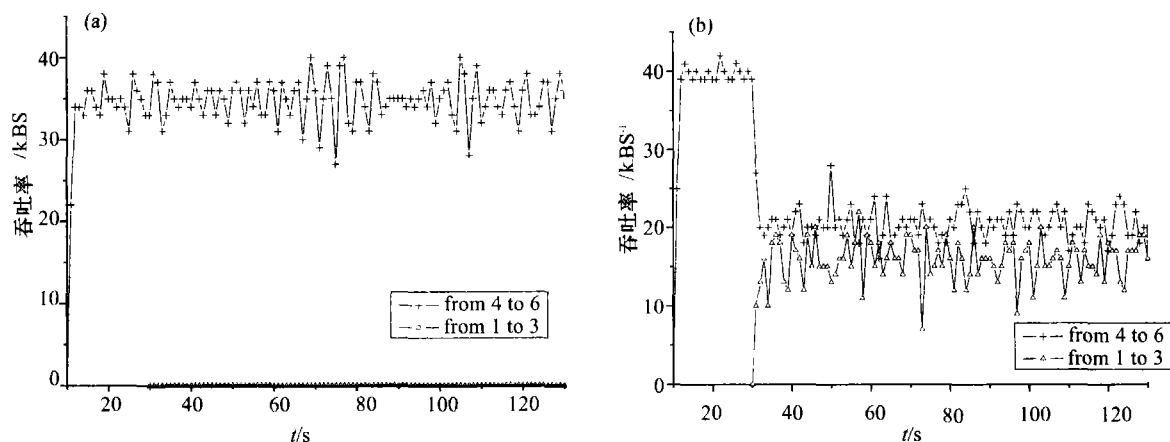


图 3 同跳数 TCP 流的不公平性改善

Fig.3 The unfairness's improvement of the same hop's TCP connections

进一步的研究表明,不公平问题的根本原因在于 802.11 介质访问控制机制 BEB 退避算法的缺陷和无线环境下的隐藏节点和暴露节点问题,主要存在的问题如下:

(1) 基本 802.11 MAC BEB 退避算法存在的问题:节点成功发送后, CW 被置为最小值,不仅不能反应当前的帧碰撞的情况,在一个冲突激烈的网络中,节点就有可能再多次调用后退算法,造成效率较低;而且成功发送帧的节点将会有较大的机会发送下一帧,甚至持续占有信道,而竞争中失败的节点将会处于更不利的地位。

(2) 隐藏节点和暴露节点问题:当跳数增多时隐藏节点和暴露节点问题变得相当严重,影响了帧的正常发送和接收。

(3) 节点不能了解周围网络状况,退避窗口由各个节点单独计算,不能根据网络状况动态地调整,导致不公平性的加剧。

2 DWDB 算法及结果分析

2.1 改进的 802.11 MAC 机制-DWDB 算法

针对不公平性问题,在文献[7]提出一种解决方法——在同一网段中使用两个信道,一个控制信道,一个数据信道。在控制信道上交换 RTS 和 CTS,数据信道专用于发送数据报文。节点就能发现自己隐藏节点后采取相应的措施。这种方案能较好地解决隐藏节点导致的不公平性问题。但是由于使用双信道,降低了网络的容量,而且需要改造物理层,很难获得推广。文献[8]和[9]提出了另一种解决 802.11 的 CSMA/CA 的不公平性的方法——当节点的 backoff time 减为 0 时,节点以概率 p 发送数据帧,如果节点不发送数据帧,则等待相同的 backoff time 后再以概率 p 发送数据帧,概率 p

的取值是按照当前网络状况(通过采用节点之间定时地交换竞争间隔)计算得到的。这种方法较好地解决了不公平问题,但是由于节点延时被增大并且要定时交换竞争间隔信息,因此吞吐量大为下降。

针对基本 802.11 MAC 存在的问题和现有算法的不足,我们提出了一种适用于 Ad Hoc 网络的 802.11 的改进算法——双窗口动态退避 DWDB (double Windows dynamic backoff) 算法,以改善基于 IEEE 802.11 的 Ad Hoc 网络的网路性能。DWDB 算法主要包括两方面的内容:根据网络状况采用竞争窗口通告机制进行动态地调整初始窗口值,以提高 IEEE 802.11 的 MAC 吞吐量;新的重传机制和双窗口机制克服隐藏节点带来的问题。

该算法的主要机制具体描述如下。

2.1.1 双窗口机制 不同于 BEB 的单窗口, DWDB 采用两个窗口值 CW1 和 CW2,其随机退避时间是用下面的公式计算得到的:

$$\text{Backoff time} = \begin{cases} \text{Random1}(\cdot) \times \text{aSlotTime} & \text{for } \text{rtime} \leq N \\ \text{Random2}(\cdot) \times \text{aSlotTime} & \text{for } \text{rtime} > N \end{cases} \quad (3)$$

其中, $\text{Random1}(\cdot)$ 为一个平均分布在 $[\text{CW1}, \text{CW2}]$ 段上的伪随机整数, $\text{Random2}(\cdot)$ 为一个平均分布在 $[0, \text{CW1}]$ 段上的伪随机整数。CW1 为一个常整数, CW2 是介于 $\text{aCW2}_{\min}$ 和 $\text{aCW2}_{\max}$ 之间的一个整数, rtime 为重传次数, N 为重传调整系数,是一个介于 aLongRetryLimit 和 $\text{dot11Short RetryLimit}$ 之间的整数。

当节点重传次数 rtime 小于 N 时,发送失败后 CW2 将被重新计算,等于 $\min \{2 \times \text{CW2} - 1, \text{aCW2}_{\max}\}$,节点将在一个更大的范围内选择随机整

数,再次发生冲突的可能性就减少。当节点重传次数大于等于 N 时,可以认为这时节点是因为隐藏节点的原因而不能发送数据帧的,此时节点将从 $[0, CW1]$ 中选择随机整数,由于其它节点都是从 $[CW1, CW2]$ 中选择随机数,一旦其它节点发送完数据帧,该节点就能够尽快捕获信道。当节点成功发送完一帧后,将在 $[CW1, aCW2_{min}]$ 中选择一个随机数等待发送下一帧。这样就使得隐藏节点问题出现时,节点都有机会发送数据帧。

2.1.2 竞争窗口通告机制 在 RTS 和 CTS 帧中增加一个字段——CW,图 4 为新的 RTS 和 CTS 帧结构,这个字段中写入节点当前的窗口值 $CW2$ 。并采用竞争窗口通告机制,当周围的节点接收到 RTS 和 CTS 后检查其中的 CW 字段的值,并重新计算接收窗口值 RC。新的接收窗口值 RC 等于原接收窗口值与 RTS/CTS 中包含的窗口值的平均。

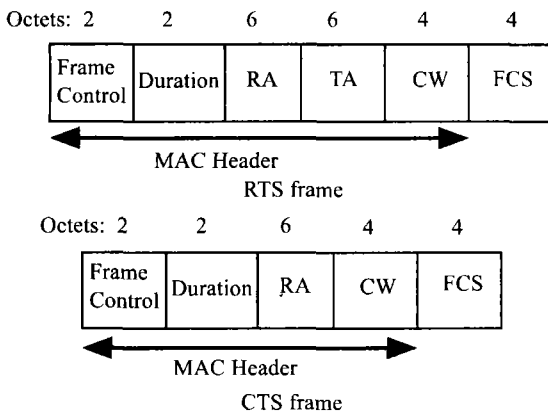


图 4 修改后的 RTS 和 CTS 帧结构

Fig.4 The Modified Format Frame of RTS and CTS

在成功发送完一帧后 $CW2$ 被设置为初始窗口值 $aCW2_{min}$ 。 $aCW2_{min}$ 将被按下列算法重新计算:
①每次成功发送后 $aCW2_{min}$ 等于 $CW2$ 与 RC 的平均;
②如果节点连续成功发送数据帧 M 次后,则 $aCW2_{min}$ 减半。 M 为网络状态平衡因子(用于在冲突环境下保持吞吐量,以及体现对网络状态变化的灵敏度)。

字段 CW 的值反映了节点冲突的情况,通过交换 $CW2$ 窗口值节点可以了解到网络当前状况,并根据网络状况动态地调整 $aCW2_{min}$ 。这一方面根据前一部分的分析提高了 MAC 的吞吐量;另一方面使得一定范围内的节点有大致相同的窗口值,这样可以避免窗口值悬殊而造成的不公平性。节点发生冲突后, $CW2$ 在一段时间内保持不变,可以减少节点再次发生冲突的可能性,从而使得节点能够稳定地发送数据帧。

在本文仿真实验中, DWDB 算法中相关参数的取值见表 1。

表 1 DWDB 算法参数值

Tab.1 The Parameters of DWDB Arithmetic

N	M	CW1	$aCW2_{min}$	$aCW2_{max}$
5	10	10	40	1024

2.2 仿真结果分析

按照 DWDB 算法,修改了 ns2 中的 mac-802-11.cc、mac-802-11.h、mac-timer.cc 和 mac-timer.h 等源文件,形成新的 MAC 协议对不公平性能进行仿真分析。图 2(b)和图 3(b)分别是采用 DWDB 算法后不同跳数和同跳数 TCP 流的吞吐量曲线。

从图 2 和图 3 可以看到应用了新的方案后处于公平性弱势的 TCP 流能够和强势 TCP 流一起同时运行,吞吐量都仅有小幅度的抖动,没有再出现 TCP 流被中断或根本无法发送数据包的情况。本文对实验拓扑网络中不同节点不同跳数的各种不公平性问题都进行了比较,综合结果表明通过双窗口动态调节,使弱势链路取得与强势链路均衡的成功发送机会,一定程度上减轻了隐藏节点的影响,使不公平性得到很大的改善。

为了定量的比较 DWDB 算法对不公平的改善程度,我们定义了链路公平性系数 LFI (link fairness index) 来进行比较, LFI 等于两条连接中较小相对吞吐量与较大相对吞吐量的比值,相对吞吐量 RT (relative throughput) 是指该连接在当前状况下的吞吐量与在无竞争环境下的吞吐量之比, RT 的大小与跳数无关。LFI 在 $[0, 1]$ 之间取值, LFI 的值越接近 1, 表示协议流的公平性越好。表 2 列出了不同 MAC 算法情况下两个 TCP 连接的吞吐量情况和公平性指数。仿真结果表明采用 DWDB 算法后,公平性系数 LFI 都提高到接近 1, TCP 的不公平性得到明显改善。

3 结 论

TCP 在基于 802.11 的 Ad Hoc 多跳网络上运行出现的不公平问题的根本原因在于无线环境下的隐藏节点问题和 802.11 介质访问控制机制的缺陷。为此本文提出了一个新方案—DWDB 算法,对基本 802.11 MAC 协议进行了退避算法和重传机制的修改。实验结果表明,采用 DWDB 算法后, TCP 的不公平性得到明显改善。

表 2 不公平性的改善

Tab.2 The improvement of the TCP unfairness

先后启动的 TCP 连接	the standard 802.11 (BEB)					the modified 802.11 (DWDB)				
	第一个 TCP 连接		第二个 TCP 连接			第一个 TCP 连接		第二个 TCP 连接		
	平均	相对	平均	相对	LFI	平均	相对	平均	相对	LFI
	吞吐率	吞吐率	吞吐率	吞吐率		吞吐率	吞吐率	吞吐率	吞吐率	
node 6→4	0.30	0.0086	81.48	0.998	0.01	17.33	0.44	34.01	0.432	0.98
node 2→3										
node 4→6	34.50	1	0	0	0	20.75	0.52	15.99	0.40	0.77
node 1→3										

参考文献:

- [1] XU S G, SAADAWI T. Reveal TCP incompatibility problem in 802.11-based wireless multi-hop network [C]. IEEE Communication Magazine, 2001: 2847 - 2851.
- [2] XU S G, SAADAWI T. Does the IEEE 802.11 MAC protocol work well in multihop wireless Ad Hoc networks [C]. IEEE Communication Magazine, 2001: 130 - 137.
- [3] GERLA M, TANG K, BAGRODIA R. TCP performance in wireless multi-hop networks [C]. New Orleans, LA: IEEE WMCSA'99, 1999: 41 - 50.
- [4] GERLA M, BAGRODIA R. TCP over wireless multi-hop protocols: simulation and experiment [C]. IEEE Communication Magazine, 1999: 1089 - 1094.
- [5] TANG K, GERLA M. Fair sharing of MAC under TCP in wireless Ad Hoc networks [C]. Venice Italy: Proc IEEE MMT'99, 1999.
- [6] ANSI/IEEE Std 802.11-1999. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification. 1999: 75 - 76.
- [7] LIU K, FUKUDA A. Adaptive acquisition collision avoidance multiple access for multi-hop Ad Hoc wireless network [C]. IEEE Communication Magazine, 2001: 2880 - 2884.
- [8] OZUGUR T, NAGHSHINEH M. Fair media access for wireless LANs [C]. Global Telecommunications Conference, Globecom'99, 1999.
- [9] OZUGUR T, NAGHSHINEH M. Optimal MAC-layer fairness in 802.11 networks [C]. IEEE Communication Magazine, 2002: 675 - 681.

Study of the Unfairness on Ad Hoc Multi-Hop Networks

SU Ai-guo, DONG Hong-yu, ZHANG Guang-zhao

(Department of Electronic and Communication Engineering, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: The unfairness problem of TCP connections was analyzed, and the DWDB (double Windows dynamic backoff) mechanism to solve this problem was proposed to modify the exponential back-off algorithm and retransmission scheme in 802.11 MAC sub-layer. The proposed mechanism is dynamic and sensitive to the change in the network condition and topology. Also, the simulation results show that the problems mentioned above were solved by using our new method.

Key words: Ad Hoc; TCP; double Windows dynamic backoff (DWDB)