基于NS2的网络负载自适应DCF实现及性能分析

宋 军[1] 金艳华[1] 宋 文[2]

(重庆交通大学信息科学与工程学院 重庆 400074)[1] (重庆邮电大学通信与信息工程学院 重庆 400065)[2]

摘 要 根据网络负载自适应DCF机制的工作流程,对NS2网络仿真软件的IEEE802.11无线局域网仿真模块-Mac_802.11类进行了修改和扩展,实现了NS2对网络负载自适应DCF机制的仿真。结果表明,与标准DCF机制相比,网络负载自适应DCF机制对IEEE802.11无线局域网的吞吐量、时延、时延抖动和公平性等方面的性能均有所提高。

关键词 网络仿真,NS2,IEEE802.11无线局域网,网络负载自适应DCF

中图法分类号 TP393.04 **文献标识码** A

Load Self-adaptive DCF Implement and Performance Analysis Based on NS2

SONG Jun[1] JIN Yan-hua[1] SONG Wen[2]

(College of Information Science and Engineering, Chonqing Jiaotong University, Chongqing 400074, China)[1]

(School of Communication and Information Engineering, Chonqing University of Post and Telecommunication, Chongqing 400065, China)[2]

Abstract According to the running process of load self-adaptive DCF mechanism, the simulation module of IEEE802.11 WLAN- Mac_802.11 class in NS2 was modified and extended to implement the simulation of load self-adaptive DCF mechanism by NS2. Simulation results indicate, compared with standard DCF mechanism, the load self-adaptive DCF mechanism can enhance the performance of IEEE802.11 WLAN in throughput, delay, delay jitter and fairness.

Keywords Network simulation, NS2, IEEE802.11 WLAN, Load self-adaptive DCF

随着网络结构和规模的复杂化以及网络应用的多样化,网络仿真在协议设计、性能分析等方面发挥着越来越重要的作用。NS2,OPNET,QualNET是目前比较流行的网络仿真工具,其中NS2具有源代码开放、扩展灵活等优点,被广泛应用于网络分析、研究和教学中[1]。据统计,在ACM和IEEE各种期刊上发表的相关学术论文中,近50%的论文采用NS2进行网络仿真与性能分析,如表1所列。

表1 网络仿真工具使用率比较

仿真器	NS2	OPNET	QualNET
传输层及以上层	75%	18%	7%
网络层	70%	18%	12%
MAC层及物理层	43%	36%	21%

数据来源于: ACM Digital Library and IEEExplore searches

IEEE802.11无线局域网采用DCF机制作为基本接入方式[2],实现比较简单,得到了广泛应用,并成为当前的研究热点。但在站点数量较多的情况下,标准DCF机制将导致IEEE802.11无线局域网的吞吐量、时延、公平性等性能明显恶化[3]。为此,国内外学者从建模分析推导优化参数[4,5]和改变竞争窗口调节机制[6-10]等方面提出了大量改进措施,以提高IEEE802.11无线局域网的性能。

本文在深入分析IEEE802.11无线局域网网络负载自适应DCF(Load Self-Adaptive DCF,LSAD)改进机制的基础上,通过对NS2网络仿真软件的IEEE802.11无线局域网仿真模块进行修改和扩展,设计和实现了网络负载自适应DCF机制仿真模块,并通过大量仿真实验,对IEEE802.11无线局域网标准DCF机制和网络负载自适应DCF机制的吞吐量、平均时延、时延抖动和公平性等性能进行了详细分析和比较。

1 NS2仿真软件

NS2是一个基于离散事件的网络仿真器,具有丰富的构件库,提供在无线或有线网络上的TCP、路由、多播等多种协议的模拟,支持广域网、局域网、移动通信网等类型的网络运行全过程仿真。NS2中常用的网络组件包括节点(Node)、链路(Link)、代理(Agent)、应用(Application)、队列(Queue)、跟踪(Trace)对象等,分别仿真网络上的主机和网络设备、包缓冲和传输延时、传输层协议、网络业务流量和应用层协议,以及配置跟踪参数、填写跟踪文件。

NS2采用OTcl和C++两种语言进行开发,其中仿真网络环境布置采用OTcl脚本语言编写,以快速完成网络组件、环境参数的设置和改变;事件调度器和基本网络组件对象采用C++语言编写,以提高具体协议的仿真效率;OTcl解释器则通过OTcl链接对编译后的C++网络组件对象进行调

到稿日期:2008-07-26 本文受重庆市自然科学基金(CSTC2006BB2413),重庆市科技攻关项目(CSTC2008AC2075),重庆市教委科技研究项目(KJ080425)资助。

宋 军(1971—),男,博士,副教授,硕士生导师,研究方向为宽带网络技术、嵌入式系统、智能交通系统,E-mail:songjun_cq@163.com;金艳华(1984—),男,硕士研究生,研究方向为无线网络技术、嵌入式系统;宋 文(1983—),男,硕士研究生,研究方向为移动通信技术、嵌入式系统。

用。利用 NS2 进行网络仿真的基本流程如下：

1. 问题定义。用户首先需要考虑仿真的网络拓扑结构应该如何布置，是否需要对源代码进行修改或添加等；

2. 是否需要修改源代码。一般情况下，用户可以利用 NS2 中已有的构件完成仿真工作，此时只需要编写 Tcl 脚本对网络仿真环境进行设置；仿真新设计的网络协议或算法，则需要修改 NS2 源代码或者编写新的 C++和 Otcl 实现代码，并将它们编译、链接到 NS2 内核中，使 NS2 支持新的协议或算法；

3. 编写 Tcl 仿真脚本。根据所需仿真的网络环境，编写相关 Tcl 脚本文件；

4. 进行仿真。利用 NS2 执行 Tcl 脚本完成仿真工作；

5. 分析结果。仿真程序运行结束后，将生成相应的 Trace 文件，用户可以利用 Trace 文件中的数据对网络性能进行分析和研究。

2 网络负载自适应 DCF

IEEE802.11 无线局域网成功发送一次数据帧的过程如图 1 所示，t_v 为站点成功传送一个数据帧所经历的时间；t_coll 为本站点在 t_v 时间段内因冲突所花费的时间；t_free 为本站点在 t_v 时间段内因退避或延迟而产生的空闲时间，t_pack 为本站点成功发送数据帧的开销，t_succ 为其他站点成功发送数据帧的开销。

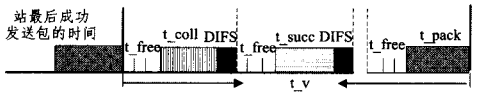

图 1 数据发送过程

在当前竞争窗口相对于站点数过小时，将产生频繁碰撞和退避，退避站点的竞争窗口短期内维持较大值，成功发送站点的竞争窗口经常重置而保持在较小值，其接入信道的概率比退避站点大，从而占用更多的信道带宽；在当前竞争窗口相对于站点数过大时，将产生不必要的退避时延，使整个网络的时延增大。因此，标准 DCF 机制中，站点采用固定的初始竞争窗口接入信道，随着网络中站点数量的增加，将造成冲突过多和频繁退避，最终导致网络的吞吐量、时延和公平性等性能明显恶化。

网络负载自适应 DCF 机制中，站点通过对网络负载情况的实时监测和初始竞争窗口的动态调整，减少冲突和频繁退避，提高网络性能。其主要工作步骤如下：

(1) 网络中的每个站点在数据发送过程中，独立地实时监测和统计信道冲突时间 t_coll 和信道空闲时间 t_free。

(2) 站点在每次成功传输一个数据帧后，首先根据式(1)和式(2)计算出信道平均冲突时间 t_coll_avg 和信道平均空闲时间 t_free_avg，然后根据式(3)计算出实时网络负载 L：

$$t\_coll\_avg=\lambda * t\_coll\_avg+(1-\lambda) * t\_coll \quad (1)$$

$$t\_free\_avg=\lambda * t\_free\_avg+(1-\lambda) * t\_free \quad (2)$$

$$L=t\_coll\_avg/t\_free\_avg \quad (3)$$

其中，λ 为平滑随机抖动参数，取值 0.925。

(3) 站点根据实时网络负载 L 与理论最优值 l_{opt} 的比较结果，调整初始竞争窗口：

①当 $L>l_{opt}+\sigma$ 时，站点将初始竞争窗口减少一倍，$cw\_=(cw\_+1)/2-1$。

②当 $L<l_{opt}-\sigma$ 时，站点将初始竞争窗口增大一倍，$cw\_=(cw\_+1)*2-1$。

③当 $l_{opt}-\sigma<L<l_{opt}+\sigma$ 时，站点的初始竞争窗口保持不变。

其中，σ 为触发窗口调节机制的门限值，取值 0.3。

3 仿真模块设计与实现

在 NS2 中，IEEE802.11 无线局域网的仿真由 Mac802_11 类实现。Mac802_11 类定义了 IEEE802.11 无线局域网标准 DCF 机制的所有操作方法，但不支持各种 DCF 改进机制的仿真。本节将详细描述对 NS2-2.29 版本中的 Mac802_11 类的修改和扩展，以实现对网络负载自适应 DCF 机制的仿真。

3.1 信道冲突时间 *t_coll* 统计

当站点在传输一个数据帧过程中产生冲突时，将调用 Mac802_11 类的 RetransmitRTS()或 RetransmitDATA()函数重传 RTS 帧或数据帧。因此，可以在这两个函数中对信道冲突时间 t_coll 进行统计。

调用 RetransmitRTS()函数重传 RTS 帧时，则上次传输该 RTS 帧所消耗的时间为信道冲突时间，即 t_coll = RTS+SIFS；若调用 RetransmitDATA()函数重传 DATA 帧时，则上次传输该 DATA 帧所消耗的时间为信道冲突时间，即 t_coll=DATA+SIFS+ACK+DIFS。实现代码见 3.2 节中修改后的 RetransmitRTS()和 RetransmitDATA()函数。

3.2 信道空闲时间 *t_free* 统计

在站点成功发送一个数据帧的过程中，可能产生以下延迟或退避时间：

1. 当数据帧由上层到达 IEEE802.11 MAC 层时，将调用 Mac802_11 类的 send()函数。send()函数首先完成发送数据的准备，然后检测退避定时器状态，如果退避定时器已启动，则等待退避定时器超时，退避定时器的剩余时间将计入 t_free；如果退避定时器没有启动，将检测信道状态，如果信道忙，则启动退避定时器，所确定的退避时间将计入 t_free；如果信道空闲，将检测延迟定时器状态，如果延迟定时器没有启动，则启动延迟定时器，所确定的延迟时间将计入 t_free；如果延迟定时器已启动，则等待延迟定时器超时，延迟定时器的剩余时间将计入 t_free。

2. 当退避定时器或延迟定时器超时后，将调用 backoffHandler()或 deferHandler()函数，这两个函数将进一步调用 check_pktRTS()或 check_pktTx()函数进行数据发送。check_pktRTS()或 check_pktTx()函数将检测信道状态，如果信道空闲，则调用 transmit()函数，将 RTS 帧或数据帧发送到物理信道上；如果信道忙，则退避窗口增大一倍，并重新启动退避定时器，所确定的退避时间将计入 t_free。当退避定时器再次超时，将重复执行本环节，直至将 RTS 帧或数据帧发送到物理信道上。

3. 当 RTS 帧或数据帧在发送过程中产生冲突时，将调用 RetransmitRTS()或 RetransmitDATA()函数进行重传。此时，将退避窗口增大一倍后启动退避定时器，所确定的退避时间将计入 t_free。

因此，统计一个数据帧成功发送过程中总的信道空闲时

间 *t_free*，需要修改 Mac802_11 类的 send()，check_pktRTS()，check_pktTx()，RetransmitRTS()和 RetransmitDATA()等函数。限于篇幅，仅给出 RetransmitRTS()和 RetransmitDATA()函数的部分修改代码：

```
Void Mac802_11::RetransmitRTS()
{
  assert(mhBackoff_.busy() == 0);
  macmib_.RTSFailureCount++;
  t_coll+=(txtime(phymib_.getRTSlen(),basicRate_)+phymib_.
getEIFS());
  ssrc_ += 1;
  if(ssrc_ >= macmib_.getShortRetryLimit())
  {
    ...
  }
  else
  {
    struct rts_frame *rf;
    rf = (struct rts_frame *)pktRTS_->access(hdr_mac::offset
_);
    rf->rf_fc.fc_retry = 1;
    inc_cw();
    mhBackoff_.start(cw_,is_idle());
    t_free+=((Random::random() % cw_) * phymib_.getSlot-
Time());
  }
}
Void Mac802_11::RetransmitDATA()
{
    struct hdr_cmn *ch;
    struct hdr_mac802_11 *mh;
    u_int32_t *rcount,thresh;
    ch = HDR_CMN(pktTx_);
    mh =HDR_MAC802_11(pktTx_);
    t_coll+=(phymib_.getDIFS()+txtime(phymib_.getACKlen
(),basicRate_)+phymib_.getSIFS()+txtime(pktTx_));
    if((u_int32_t)ETHER_ADDR(mh->dh_ra) == MAC_
BROADCAST)
    {
      Packet::free(pktTx_);
      pktTx_ = 0;
      inc_cw();
      mhBackoff_.start(cw_,is_idle());
      t_free+=((Random::random() % cw_) * phymib_.getSlot-
Time());
      return;
    }
    macmib_.ACKFailureCount++;
    if((u_int32_t) ch->size() <= macmib_.getRTSThreshold())
    {
      ...
    }
    (*rcount)++;
    if(*rcount >= thresh)
    {
      ...
    }
    else
    {
      struct hdr_mac802_11 *dh;
      dh = HDR_MAC802_11(pktTx_);
      dh->dh_fc.fc_retry = 1;
      sendRTS(ETHER_ADDR(mh->dh_ra));
      inc_cw();
      mhBackoff_.start(cw_,is_idle());
      t_free+=((Random::random() % cw_) * phymib_.getSlot-
Time());
    }
}
```

3.3 网络负载更新及竞争窗口调节

笔者新开发了 Refresh_load()和 Adjust_cw()函数，以实现网络负载更新和初始竞争窗口调节。当站点成功发送一个数据帧后，将接收到目的站点反馈的确认帧，此时源站点将调用 recvACK()函数对确认帧进行处理，以结束此次成功的数据帧发送过程。因此，可以在 recvACK()函数中，调用 Refresh_load()和 Adjust_cw() 函数来完成对网络负载状况的更新和站点初始竞争窗口的调节。Refresh_load()和 Adjust_cw() 函数的实现代码如下：

```
double Refresh_load()
{
  t_coll_avg=b_sr*t_coll_avg+(1-b_sr)*t_coll;
  t_free_avg=b_sr*t_free_avg+(1-b_sr)*t_free;
  if(t_free_avg<=0) l_now=l_opt;
  else l_now=t_coll_avg/t_free_avg;
  return l_now;
}
void Adjust_cw()
{
  if (l_now>l_opt+a_tv)
  {
    cw_=(cw_+1)*2-1;
  }
  else if (l_now<l_opt-a_tv)
  {
    cw_=(cw_+1)/2-1;
  }
  else cw_=cw_;
  if (cw_>(phymib_.getCWMax())
    cw_=(phymib_.getCWMax());
  if (cw_<phymib_.getCWMin())
    cw_=phymib_.getCWMin();
}
```

4 仿真及性能分析

笔者在完成网络负载自适应 DCF 机制仿真模块开发的基础上，编写 Tcl 脚本，进行了大量仿真实验，以便对网络负载自适应 DCF 机制和标准 DCF 机制的性能进行分析和比较。

4.1 仿真环境设置

仿真的网络拓扑结构如图 2 所示，其中偶数站点为发送方，奇数站点为接收方。传输层代理采用 TCP 协议，应用层

绑定 FTP 数据流。表 2 给出了网络仿真的主要参数设置。

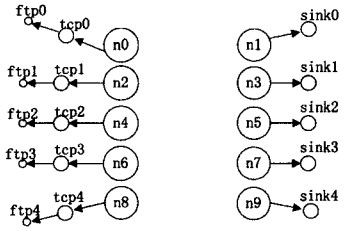

图 2　仿真网络拓扑结构

表 2　仿真参数

信道带宽	1Mbps
站点数量	50,70,100,130,160
竞争窗口最小值 CWmin	31
竞争窗口最大值 CWmax	1023
最优负载 l_opt	0.85
门限值 σ	0.3

4.2　性能分析

吞吐量分析。各种网络规模下的网络平均吞吐量如图 3 所示。随着网络中站点数量不断增加，标准 DCF 机制和网络负载自适应 DCF 机制的网络平均吞吐量均呈下降趋势，但网络负载自适应 DCF 机制的网络平均吞吐量提高了约 6%。

时延分析。各种网络规模下的平均时延如图 4 所示。随着网络中的站点数量不断增加，标准 DCF 机制和网络负载自适应 DCF 机制的平均时延都呈增大趋势，但网络负载自适应 DCF 机制的平均时延始终小于标准 DCF 机制；当网络规模超过为 100 个站点后，标准 DCF 机制的平均时延迅速增大，网络负载自适应 DCF 机制的平均时延始仍然呈缓慢上升趋势。

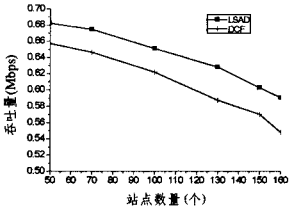

图 3　网络平均吞吐量

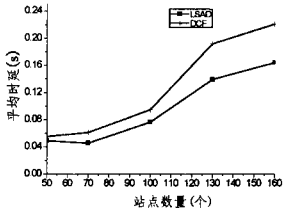

图 4　网络平均时延

时延抖动分析。标准 DCF 机制和网络负载自适应 DCF 机制的时延抖动如图 5 所示。

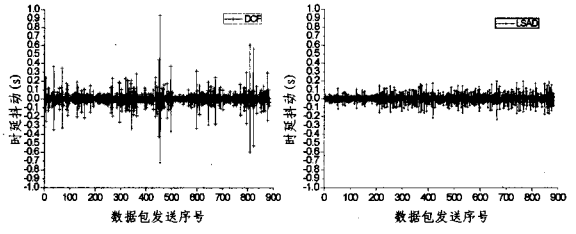

图 5　时延抖动比较

标准 DCF 机制的时延抖动很大，最大幅度达到了－0.718s 至＋0.935s；此外，标准 DCF 机制的时延抖动幅度出现剧烈变化的频率较高，经常超出±0.1s 范围。网络负载自适应 DCF 机制的时延抖动较小，最大幅度在－0.211s 至＋0.241s 之间，比标准 DCF 机制的时延抖动幅度小得多；同时，网络负载自适应 DCF 机制的时延抖动幅度通常维持在±0.1s 之间，不会出现频繁的剧烈变化。

公平性分析。网络中各个站点的平均吞吐量如图 6 所示。标准 DCF 机制中各站点的平均吞吐量相差很大，信道带宽的分配极不公平，例如，18 号站点的平均吞吐量为 0.025 Mbps，而 28 号站点的平均吞吐量为 0.079Mbps，是 18 号站点的 2.16 倍；网络负载自适应 DCF 机制中各个站点的平均吞吐量基本上都维持在 0.048Mbps 左右，各站点占用的信道带宽相差很小，网络信道带宽的分配比较公平。因此，网络负载自适应 DCF 机制显著改善信道带宽分配的公平性。

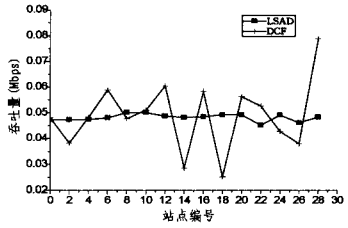

图 6　站点平均吞吐量

结束语　本文结合 IEEE802.11 无线局域网网络负载自适应 DCF 机制，基于 NS2 开发了相应的仿真模块，实现了 NS2 中网络负载自适应 DCF 机制的仿真。仿真结果表明，与标准 DCF 机制相比，网络负载自适应 DCF 机制对 IEEE802.11 无线局域网的网络吞吐量和平均时延有一定的提高，但改善并不明显，而对时延抖动和公平性有较大改善。

参 考 文 献

[1] Fall K, Varadhan K. The ns Manual [EB/OL]. http://www.isi.edu/nsnam/ns/doc. 2003

[2] IEEE Std. 802.11, 1999 Edition, PartII: IEEE 802.11 Wireless LAN medium access control (MAC) and physical (PHY) layer specifications[S], 1999

[3] Peng Y, Wu H T, Long K P, et al. Simulation analysis of TCP performance on IEEE 802.11 Wireless LAN[J]// Proc. of the ITIN2001. 2001:520-525

[4] Cali F, Conti M, Gregori E. IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism[J]. IEEE JSAC, 2000, 18(9)

[5] Bianchi G. Performance analysis of the IEEE802.11 distributed coordination function. IEEE JSAC, 2000, 18(3)

[6] Ozugur T, Naghshineh M, et al. Fair media access for wireless LANs[J]// Proc. of the GLOBECOM'99. 1999:570-579

[7] Aad I, Castelluccia C. Differentiation mechanisms for IEEE 802.11[J]// Proc. of the ICCS^{20th}. 2001:209-218

[8] Younggoo K, Yugang F, Haniph L. A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs[J]// Proc. of IEEE InfoCom'2003. 2003

[9] 彭泳，程时端. 一种自适应无线局域网协议[J]. 软件学报，2004(4):604-615

[10] 刘岩，舒炎泰，张亮，等. 无线网络中分布式协调功能的改进[J]. 计算机应用，2004(7):162-166