

增强超宽带无线网络 QoS 的调度算法

张 盈,王洪江,韦 岗

ZHANG Ying,WANG Hong-jiang,WEI Gang

华南理工大学 电子与信息学院,广州 510640

School of Electronic and Information Engineering,South China University of Technology,Guangzhou 510640,China

E-mail:zhangyingmml@163.com

ZHANG Ying,WANG Hong-jiang,WEI Gang.Scheduling algorithm for QoS enhancement in Ultra-Wide Band(UWB) wireless networks.Computer Engineering and Applications,2009,45(8):125-128.

Abstract: Based on distributed UWB networks specified by ECMA-368 protocol,an advanced scheduling algorithm is proposed to efficiently enhance traffic QoS in the paper.The scheduling algorithm runs in MAC layer without a central scheduler.It uses a borrowing-lending scheme to achieve dynamic band width allocation for variation of wireless channel status and traffic transmission status.Simulation results show that the scheduling algorithm can efficiently enhance the traffic QoS and improve the performance of the whole network.

Key words: Ultra-Wide Band(UWB);scheduling;Quality of Service(QoS);band width allocation

摘 要:基于对 UWB 网络中已有调度算法性能的分析,提出了一种用于增强 UWB 无线网络 QoS 的调度算法。该算法采用跨层设计的思想,能根据无线物理层信道状态和应用层业务传输速率的变化进行动态带宽分配,分配过程主要采用带宽借贷的思想,即在具有不同优先级的业务流之间进行带宽调度,以尽最大可能满足具有不同优先级的业务流 QoS 要求。仿真结果表明,该算法能有效地增强网络的 QoS,提高整个网络的性能。

关键词:超宽带;调度;服务质量;带宽分配

DOI:10.3778/j.issn.1002-8331.2009.08.038 **文章编号:**1002-8331(2009)08-0125-04 **文献标识码:**A **中图分类号:**TP393

1 引言

超宽带(Ultra-Wide Band,UWB)无线网络是以超宽带通信技术作为物理层的网络系统,是下一代短距离无线通信网络系统的重要组成部分。欧洲 ECMA 组织制定了超宽带网络的媒体接入控制(Medium Access Control,MAC)层协议及其与物理层的接口协议,即 ECMA-368^[1]和 ECMA-369^[2]。当前,这两个标准协议已经通过国际标准化组织的审议,即将成为 Wimedia 网络系统的标准协议体系。按照该标准协议,人们可以搭建最基本的网络平台系统。然而,针对网络中业务的 QoS 问题,该协议并未提出一种有效的解决方法,因此,如何保证网络中业务的 QoS,已经成为当前 UWB 网络研究的重要内容。其中 MAC 层的资源调度算法是保证业务 QoS 要求的基础,是 UWB 网络研究的关键技术之一。实验表明,有效的资源调度算法能大幅度地提升无线网络的整体性能,在满足各种业务流 QoS 要求的基础上,也使得网络的带宽资源得到充分的利用。因此,研究 UWB 网络中的资源调度算法具有极其重要的实际意义。

到目前为止,已有许多文献对 UWB 网络的资源调度算法进行了深入的研究。文献[3]基于 IEEE 802.15.3 标准的 UWB

网络提出了一种满足公平原则的包调度算法,该算法利用了多个节点之间能进行并发传输数据的特点,来提高网络的吞吐量。但是,该调度算法并没有考虑业务流的 QoS 要求,所以,这种算法所具有的资源利用效率并不高。文献[4]针对具有中心调度器的蜂窝网络,提出了考虑多种业务流 QoS 要求和信道状态变化的动态资源调度算法。然而,这种算法并不适用于分布式的 UWB 网络。文献[5]提出了一种动态资源分配的调度算法,该算法基于反馈的收发数据包错误率,通过重传控制和前向纠错的方法来控制数据包的发送,但是该算法并没有给出当信道状态改变时该如何通过增加或释放节点带宽来实现最优资源调度的过程。文献[6]提出了一种基于速率的带宽借用思想,在带宽资源损失容限的范围内,放弃部分带宽资源提供给其他节点,以提高整个系统的吞吐率。可见,现有的各种资源调度算法要么只适用于非分布式的网络系统,要么未能反映当信道状态和业务流速率同时发生变化的状况,对于基于分布式结构的 UWB 网络的调度算法,还有待进一步的研究。

为此,本文提出了一种基于 ECMA-368 标准的分布式 UWB 网络系统的动态资源调度算法,主要通过所有节点的资源利用

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60672065);国家杰出青年科学基金(the National Science Foundation of China for Distinguished Young Scholars under Grant No.60625101)。

作者简介:张盈(1982-),女,硕士研究生,主要研究方向为无线通信;王洪江(1977-),男,博士后,主要研究方向为无线通信;韦岗(1963-),男,教授,博士生导师,主要研究方向为无线通信、多媒体信号处理。

收稿日期:2008-01-28 **修回日期:**2008-04-11

状态、应用层状态和物理层传输信道的状态信息,实现整个网络中带宽资源的动态分配,以满足 UWB 网络中不断变化的业务流 QoS 要求,从而有效地提高带宽资源的利用率。因此,本算法利用了跨层设计的思想,通过结合物理层信道状态和应用层业务流状态的信息,在整个网络所有节点之间,进行带宽资源的借和贷操作,从而实现带宽资源的动态分配。在 ECMA-368 UWB 的 NS2 仿真平台上的运行结果表明,本文的算法在不同程度的网络负载下,都能实现带宽资源的最优动态分配。

2 UWB 网络模型

基于 ECMA-368 协议的 UWB 网络采用全分布式结构,网络中没有中心协调器,所有节点之间的信息交互是通过每个节点依次广播一种信标帧 (beacon 帧) 来实现的。其 MAC 层协议采用时分复用 (TDMA) 的接入模式。每个节点均采用资源预留协议 (Distributed Reservation Protocol, DRP) 的方式接入,即在数据传输之前,节点先预留带宽资源,在得到足够的带宽之后,才能进行数据的传输。ECMA-368 协议采用如图 1 所示的超帧结构:

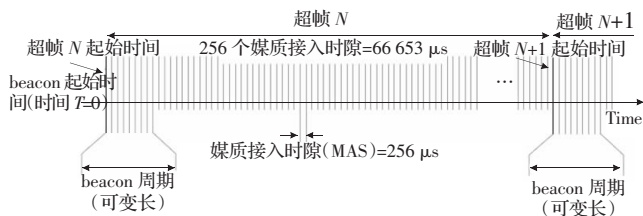


图 1 MAC 超帧结构

一个超帧持续时间为 65 535 μs ,可划分为 256 个媒质接入时隙 (MAS)。其中,最前面的 32 个 MAS 用于发送 beacon 帧,每个 beacon 帧由多个信息单元 (Information Element, IE) 组成,这些 IE 主要用于实现控制信息的交互。其余 224 个 MAS 主要用于传输数据帧信息。所以在 UWB 网络中,带宽资源的分配过程就是对 MAS 的分配过程。

与其他无线网络类似,UWB 网络的应用层业务流包括 CBR、VBR 和 UBR 三种类型。在日常生活中,CBR 对应于语音传输业务,VBR 对应于视频传输业务,UBR 对应于普通的文件传输业务。在日常应用中,CBR 和 VBR 业务的 QoS 要求的优先级往往高于 UBR 业务。由于 UWB 网络拥有较高的速率,低速率 CBR 业务的 QoS 要求通常都能够得到满足。因此,在高速的 UWB 网络中,下文中调度算法的设计主要考虑 VBR 业务,特别是高速率的 VBR 业务,比如高清电视的 QoS 要求。此外,ECMA-368 协议规定 UWB 网络的物理层采用 MB-OFDM 通信技术,其传输频段为 3.1~10.6 GHz,支持的数据传输速率为 53.3 Mb/s, 80 Mb/s, 106.7 Mb/s, 160 Mb/s, 200 Mb/s, 320 Mb/s, 400 Mb/s 和 480 Mb/s。

3 增强业务流 QoS 的调度算法

计算机通信网络的跨层设计思想,是当前比较热门的研究课题,其主要目的是通过收集多个层中的数据信息,来优化某个层的协议算法。本文算法就是基于应用层和物理层的跨层信息的结合。根据用户占用的带宽 (MAS 数) 与应用层业务流传输速率和物理层传输速率之间的关系,可以得到满足单个业务流的 QoS 所需占用的 MAS 数表达式为:

$$N_{\text{mas}} = \frac{V_{\text{expected}} \times 256}{V_{\text{channel}}} \quad (1)$$

其中, N_{mas} 为业务流在每一个超帧时间长度中满足其 QoS 所要求占用的时隙数, V_{expected} 为应用层业务流的速率, V_{channel} 为无线信道的传输速率。

在 ECMA-368 定义的 UWB 网络中,各个节点在发送 beacon 帧广播自身信息的同时,监听邻居节点的 beacon 帧来获取信息。同时,协议规定 beacon 帧可以携带多个面向应用的信息单元 (Application-Specific Information Element, ASIE)。这里设计了两种 ASIE 机制,分别为节点信息单元 (Node Information Element, NIE) 和时隙变化信息单元 (MAS Changed Information Element, MCIE)。调度算法运行之前,各个节点广播自身的 NIE,同时收集邻居节点的 NIE 组成一个全局信息表 (Global Information Table, GIT),作为运行调度算法的数据基础。调度算法运行之后,各个节点根据自身带宽分配结果的变化广播 MCIE,同时根据收集邻居节点的 MCIE 更新带宽资源的利用状况,为下一次调度算法的运行提供保证。

GIT、NIE 和 MCIE 的结构如图 2 所示。其中,设备号域用来区分不同的设备;业务类型域用来区分业务流的类型;业务速率域用来标记当前业务流的传输速率;占用 MAS 数域用来记录设备运行调度算法后占用的 MAS 数;信道速率域用来标记当前信道传输速率;调度标志域标记设备经过调度算法后能否满足 QoS 要求,满足则标记为 0,否则标记为 1;改变类型域用来标记 MAS 数是增加还是释放操作;MAS 改变位置域用来标记在一个超帧中 MAS 改变的位置。

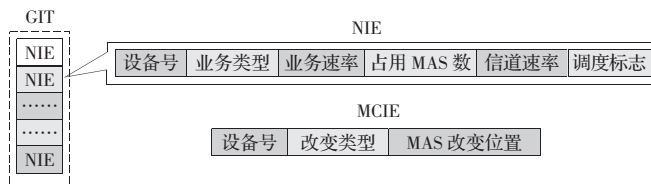


图 2 GIT、NIE 和 MCIE 结构示意图

可见,通过 NIE 和 MCIE 的发送和收集,每个节点在同一时刻包含的 GIT 信息,即全局业务流状态信息是一致的,所以每个节点根据 GIT 信息所做出的资源调度结果也将一致。因此,可以根据 GIT 提供的全局状态信息,采用借贷算法来动态分配每个业务的带宽。首先,定义业务流的优先级别,从高到低依次为 CBR、VBR、UBR。高优先级的业务可借用低优先级业务的带宽;低优先级的业务在保证不断开连接的情况下,可以贷出带宽。ECMA-368 UWB 环境下低速率 CBR 的 QoS 要求能优先得到满足,因此,本算法只需考虑 VBR 业务的 QoS 要求。每个 VBR 和 UBR 业务均按照图 3 的流程进行借贷。对每个 VBR 业务,考察其调度标志域,其值为 1 的 VBR 业务最先进入如图 3(a)所示的带宽动态借贷过程。

图 3 中, N_{need} 为 VBR 业务达到 QoS 要求所需要改变的 MAS 数,其表达式为:

$$N_{\text{need}} = N_v - N'_v \quad (2)$$

N_v 和 N'_v 分别为 VBR 业务根据式 (1) 计算出的当前帧 QoS 要求的和上一帧占用的 MAS 数。 N_{sum} 为所有 UBR 节点所能提供的 MAS 数和系统空闲的 MAS 数总和,即:

$$N_{\text{sum}} = \sum_{i=1}^n N_{u_i} + N_{\text{sys}} \quad (3)$$

其中 N_{u_i} , $i=1, \dots, n$ 为第 i 个 UBR 能提供的 MAS 数, N_{sys} 为当前

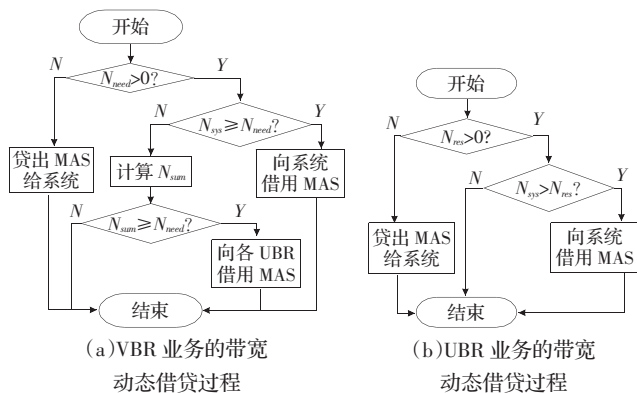


图3 业务流借贷算法流程图

系统空闲的 MAS 数。当系统空闲 MAS 不足时,根据每个 UBR 能够贷出 MAS 数目大小的比例向多个 UBR 借用 MAS,向第 i 个 UBR 借用的 MAS 数 N_{b_i} 可表示为:

$$N_{b_i} = \text{round} \left(\frac{N_{u_i} N_{\text{need}}}{\sum_{i=1}^n N_{u_i}} \right), \quad i=1, \dots, n \quad (4)$$

因为有取整操作,所以最后所借出的 MAS 不一定与需求的 MAS 数相等,还要计算两者的差值 N_{balance} :

$$N_{\text{balance}} = N_{\text{need}} - \sum_{i=1}^n N_{b_i} \quad (5)$$

若 N_{balance} 大于零,则将多余的 MAS 还给 N_{b_i} 最小的 UBR 业务;若其 N_{balance} 小于零,则再向 N_{b_i} 最大的 UBR 用户再借用 N_{balance} 个 MAS,使借用的 MAS 数与 N_{need} 相等。

图 3(b) 为 UBR 业务带宽动态借贷过程,其中, N_{res} 为 UBR 业务达到 QoS 要求所需要改变的 MAS 数与借出 MAS 数之差,其表达式为:

$$N_{\text{res}} = (N_{u_i} - N'_{u_i}) - N_{b_i} \quad (6)$$

其中, N_{u_i} 和 N'_{u_i} 分别为 UBR 业务根据式(1)计算得到当前帧 QoS 要求和上一帧占用的 MAS 数。 N_{b_i} 为 UBR 业务借出的 MAS 数且 N_{b_i} 与 N_{b_i} 相等。

上述过程表明,基于借贷机制的调度算法尽力满足了高优先级业务流 VBR 的 QoS 要求。在 VBR 业务之间, MAS 的借用首先发生在调度标志域为 1 的业务,也就是说,上一次没有成功满足 QoS 要求的业务本次可优先满足其 QoS 要求。在 UBR 业务之间,当 VBR 业务需要借用 MAS 的时候,所有的 UBR 都参与了借用过程,而且借出 MAS 能力大的 UBR 提供的 MAS 数多。可见,基于借贷机制的调度算法在同一优先级级别的业务之间是考虑公平性的。

4 实验结果与分析

这里给出本文算法在基于 ECMA-368 UWB 的 NS2 平台上进行仿真的结果。仿真过程中,采用了 17 对节点,每对节点之间传送一种类型(CBR、VBR 或 UBR)的业务流。其中, CBR 业务速率恒定为 0.064 Mb/s, VBR 业务速率在 18~22 Mb/s 之间随机变化, UBR 业务速率小于等于 1 Mb/s, 设备之间的最大传输距离设为 3 m。信道传输速率在 ECMA 协议规定的几种 UWB 速率之间随机变化, VBR 业务流传输速率也随之变化。

本文算法在系统轻负载(初始每帧有 49 个 MAS 空闲)和重负载(初始每帧只有 2 个 MAS 空闲)的情况下均进行了仿真

实验。首先随机截取了 VBR 业务在 157~207 帧之间的运行结果,绘制了如图 4 所示的带宽分配与时间曲线关系图。

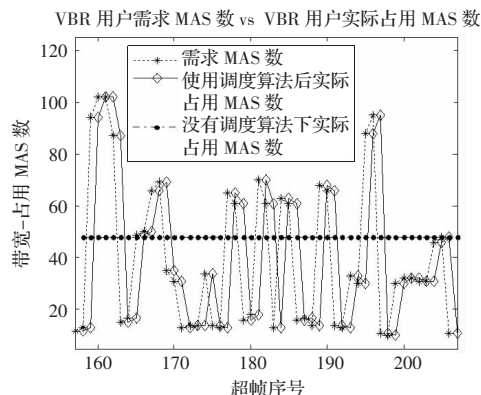


图4 带宽分配-时间曲线

从图 4 中可以看出,每一时刻的业务 QoS 要求(即 MAS 数)会随着业务流传输速率与信道速率的变化而动态改变。没有使用调度算法前, VBR 用户实际占用的 MAS 数固定不变,不能反映出 QoS 要求 MAS 数的改变,比如,在 QoS 需求减少时,却占用了多余的 MAS 数,极大地浪费了带宽资源;而在 QoS 需求增多时,又不能得到满足。使用本文调度算法后,业务占用的带宽随着 QoS 要求动态地改变,能及时地满足带宽的 QoS 要求。另外,图中显示,本算法的调度结果有一个超帧(65 535 μs)长度的时延,这是因为获取无线信道的信息需要花费一个超帧的时间,但是这不会影响到本算法的运行效果。

为了反映不同信道状态下带宽的分配情况和本算法的成功率,统计了 1000 s 内, VBR 业务流在不同信道速率下实际占用 MAS 数的平均值,得到了如图 5 所示的带宽分配-信道速率的分布图。

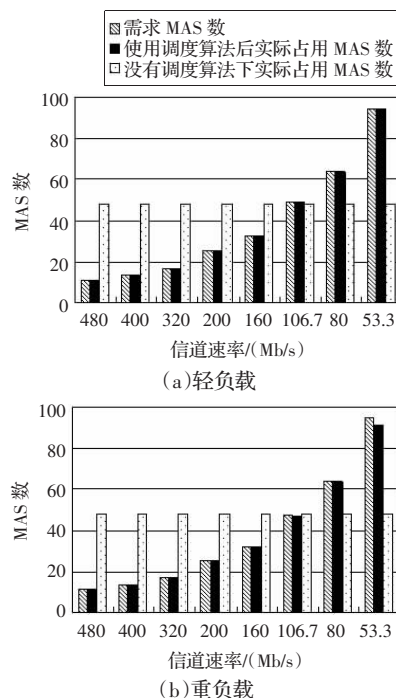


图5 带宽分配-信道速率的分布图

由图 5 可见,在业务流传输速率随机变化的情况下,当信道状况变差,即信道速率越来越低时, QoS 需求的 MAS 数越来越多。显然,没有调度算法的情况下, VBR 用户实际占用的 MAS 数固定不变,不能反映出 QoS 需求 MAS 数的改变。图 5(a) 给

出了轻负载情况下的对比结果。可以看出, VBR 用户实际占用的 MAS 数与 QoS 需求相同, 即在信道状态变化时, 本算法能很好地满足业务的 QoS 要求。图 5(b) 给出了重负载情况下的对比结果。可以看出, 信道速率较高时, QoS 需求的 MAS 数较少, 本调度算法仍能很好地满足业务的 QoS 要求; 当信道状况变差, 即信道速率较低时, 由于 QoS 需求的 MAS 数大大增多, 而重负载环境下系统可提供的带宽很少, 即使运用调度算法抢占低优先级用户的带宽, 在个别时刻也难以达到其 QoS 要求。因此, 使用调度算法后实际占有的 MAS 数均值比 QoS 要求的 MAS 数均值偏低。这里统计了轻重负载下 VBR 业务运行调度算法后能达到其 QoS 要求的成功率如表 1 所示。

表 1 业务流达到 QoS 要求的成功率

信道速率/(Mb/s)	480	400	320	200	160	106.7	80	53.3
轻负载成功率/(%)	100	100	100	100	100	100	100	100
重负载成功率/(%)	100	100	100	100	100	99.03	98.88	95.43

可见, 在轻负载环境下本算法都能成功满足 QoS 要求的带宽。即使在重负载环境下, 160 Mb/s 速率以上也都能达到 100% 的成功率。只有在信道速率下降到 106.7 Mb/s 后, 成功率才稍有下降, 但是仍然达到 95% 以上, 非常逼近于业务 QoS 的要求。

5 结论

本文提出了一种用于分布式 UWB 无线网络的动态资源调度算法, 主要采用了跨层的设计思想, 充分利用了应用层和物理层的状态变化信息, 通过所有节点的带宽资源借贷的方式,

来实现带宽资源的动态分配, 从而增强整个 UWB 无线网络的 QoS。仿真结果表明, 在网络轻负载和重负载的情况下, 该算法都能很好地满足不断变化的业务流的带宽资源需求, 从而增强整个网络系统的 QoS 性能。

参考文献:

- [1] ECMA. Standard ECMA-368 1st-December 2005 High rate ultra wideband PHY and MAC standard[S]. Ecma International, December 2005.
- [2] ECMA. Standard ECMA-369 1st-December 2005 MAC-PHY interface for ECMA-368[S]. Ecma International, December 2005.
- [3] Jiang H, Zhuang W. Effective packet scheduling with fairness adaptation in ultra-wideband wireless networks[J]. IEEE Transactions on Wireless Communications, 2007, 6(2): 680-690.
- [4] Young-June Choi, Saewoong Bahk. Wireless-adaptive fair scheduling for multimedia stream in time division multiplexed packet cellular systems[C]//Eighth IEEE International Symposium on Computer and Communication, 2003, 2: 1085-1090.
- [5] Giancola G, De Nardis L, Di Benedetto M G, et al. Dynamic resource allocation in time-varying ultra wide band channels[C]// IEEE International Conference on Communications, June 2004, 6: 3581-3585.
- [6] Ei-Kadi M, Olariu S, Abdel-Wahab H.A rate-based borrowing scheme for QoS provisioning in multimedia wireless networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(2): 156-166.

(上接 108 页)

复杂度的估计, 具体的复杂度要根据具体情况而定。

如果初始指针 i 落在 a 这段已赋值的中间某个位置, 可以先让指针往前走, 然后再往后走来分析, 对结果没有任何影响, 所以假设初始指针恰好落在第一个位置。如果攻击者已经得到了 j_i 的值, 那么就可以按照上面描述的方法进行。如果 j_i 是未知的, 能够计算出 j_i 的概率为 $(a/2^n)^2$, 因为此时 $S_{i-1}[i]$ 是已知的, 必须保证 Z_i 和 $S_{i-1}[j_i]$ 同时落在这 a 个已知的值之中, j_i 才能够计算出来, 概率恰为 $(a/2^n)^2$, 只有具有了 j_i 的正确值, 才能保证在这最初的 a 个过程中所有 j_i 都是正确的, 这样分析过程才能够顺利进行下去。但是, 如果初始指针 i 出现在没有赋值过的地方, 即未知的位置, 或者不能够正确计算出 j_i , 只能完全按照猜测赋值方法, 逐个进行赋值, 否则, 因为 j_i 完全依赖于 j_{i-1} 和 $S_{i-1}[i]$, 该分析方法是无法进行下去的。本文默认已知 j_i 的初始值。

4 结束语

值得一提的是, 当已知的取值为非连续的情形时, 复杂度比较接近于随机分布下的复杂度, 因此, 需要直接使用 Knudsen 等人的方法进行分析。

当已知的取值为连续的情形时, 复杂度要比同样数量的已知值在随机分布的情形下低得多, 而且, 这部分连续值的数量知道得越多, 复杂度降低得会越快, 当有连续 100 个以上的

取值泄漏时, 普通计算机就可以在很短时间内计算出整个完整的初始状态, 并能够预测以后的所有输出值, 从而破译 RC4 算法。

参考文献:

- [1] Roos A.A class of weak keys in the RC4 stream cipher. Two Posts in Sci Crypt, message-id: 43u1eh\$1j3@hermes.is.co.za and 44ebge\$llf@hermes.is.co.za, 1995.
- [2] Fluhrer, Mantin, Shamir. Weaknesses in the key scheduling algorithm of RC4[C]//Selected Areas in Cryptography: 8th Annual International Workshop, SAC 2001, Canada, August, 2001. New York: Springer, 2001, 2259: 1-24.
- [3] Golić. Linear statistical weakness of alleged RC4 keystream generator[C]//Advances in Cryptology-EUROCRYPT'97, EUROCRYPT'97, Germany, May, 1997. Berlin: Springer-Verlag, 1998, 1233: 226-238.
- [4] Fluhrer, McGrew. Statistical analysis of the alleged RC4 keystream generator[C]//Proceedings of the 7th International Workshop on Fast Software Encryption, FSE 2000, USA, November 6-10, 2000. New York: Springer, 2000, 1978: 19-30.
- [5] Knudsen, Meier, Preneel, et al. Analysis methods for (alleged) RC4[C]//Advances in Cryptology-ASIACRYPT'98, ASIACRYPT'98, Beijing, October, 1998. Beijing: Springer-Verlag, 1998, 1514: 327-341.
- [6] Mantin I. The security of the stream cipher RC4[D]. Rehovot Israel: The Weizmann Institute of Science, 2001.