

文章编号:2095-6134(2015)04-0562-09

# 移动智能终端下 3G 网络低功耗延迟唤醒策略<sup>\*</sup>

陈 博<sup>1†</sup>, 李 曦<sup>1,2</sup>, 周学海<sup>1,2</sup>, 席 菁<sup>1</sup>, 朱宗卫<sup>2</sup>  
(1 中国科学技术大学软件学院, 江苏 苏州 215123; 2 中国科学技术大学计算机科学与技术学院, 合肥 230027)  
(2014 年 8 月 13 日收稿; 2014 年 10 月 28 日收修改稿)

Chen B, Li X, Zhou X H, et al. Low-power strategy based on delayed awakening 3G network for smartphone[J].  
Journal of University of Chinese Academy of Sciences, 2015,32(4):562-570.

**摘 要** 突发性数据传输易导致 3G 网络工作模式从低功耗状态向高功耗状态迁移,状态切换加剧能耗开销. 本文对移动终端网络通信行为进行分析,提出移动终端下 3G 网络低功耗延迟唤醒策略. 该策略对网络数据进行延时敏感性划分,并据此进行不同时间的延时传输,延迟 3G 网络唤醒时间达到降耗目的. 仿真实验表明,策略有效降低系统功耗,具有较强的可行性.  
**关键词** 3G 网络; 低功耗; 网络通信行为  
**中图分类号**:TP338.6      **文献标志码**:A      **doi**:10. 7523/j. issn. 2095-6134. 2015. 04. 020

## Low-power strategy based on delayed awakening 3G network for smartphone

CHEN Bo<sup>1</sup>, LI Xi<sup>1,2</sup>, ZHOU Xuehai<sup>1,2</sup>, XI Jing<sup>1</sup>, ZHU Zongwei<sup>2</sup>  
(1 School of Software Engineering, University of Science and Technology of China, Suzhou 215123, Jiangsu, China;  
2 School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

**Abstract** The burst data transfer will lead to 3G migration from the low-power state to high-power state, and frequent switching exacerbates the energy consumption. We analyze 3G network communication behavior and give a low-power strategy based on delayed awakening 3G network for smartphone. The strategy classifies packages according to the delay-sensitivity, and delays sending packet for different intervals so as to reduce the energy consumption. Simulation results show that the proposed mechanism effectively reduces the power consumption of the system and has the strong feasibility.  
**Key words** 3G network; low power; communication behavior

近年,移动互联网的快速发展和多种通信网络(3G、4G、逐步覆盖城域的 WIFI 网络等)规模的迅速扩张,促使移动终端下通信应用、电商应用、移动支付等都呈现极速上升趋势. 而在移动应用快速增长的同时,智能终端功耗问题日益凸显. 据统计数据<sup>[1-2]</sup>,信息与通信业所占能耗为国民生产总能耗 3%,其中 57% 来自移动通信网络和移动终端设备,随着 4G 网络<sup>[3]</sup>、大数据中心的日渐兴起该比例仍在逐年增加. 究其原因,一方面由于通信业快速发展,促使移动设备使用变得多样

<sup>\*</sup> 国家自然科学基金(61272131,61379040)资助  
<sup>†</sup> 通信作者, E-mail: chenbo2008@ustc.edu.cn

化、日常化. 另一方面则由于在移动通信网络中, 相应网络架构<sup>[4]</sup>、通信协议等在最初设计时更多追求性能指标(如吞吐量、QoS、可靠性等), 并未考虑或较少考虑功耗因素.

移动网络通信行为受限于使用方式影响, 数据传输常呈现随机性、突发性等特点<sup>[5]</sup>, 而该行为易导致网络部件频繁切换工作模式, 消耗大量能耗. 如音乐类应用在后台周期性与服务端进行数据交换用于获取听众喜好、更新目录等, 该类小流量数据通信却导致 3G 模块瞬时能耗占据较大比重的系统总能耗. 原因在于突发数据导致 3G 网络从低功耗 IDLE 状态跃迁至高能耗 FACH 状

态或 DCH 状态. 而对于其他类应用程序的一些基本操作, 如用户点击屏幕(加载文章)、滑动屏幕(载入标题和图片)等交互都会导致突发网络数据传输, 同样导致 3G 网络工作状态切换. 典型移动智能终端通信行为如图 1 所示, 其数据传输间歇式进行, 其中短间隔描述通常为临时通信、握手协议等通信行为, 长间隔通常为周期性与服务端进行数据同步或消息推送等通信行为. 其中数据包簇①与数据包簇②的发生前段均需进行工作模式切换, 将直接导致网络部件由低功耗睡眠状态切换至高功耗工作状态.

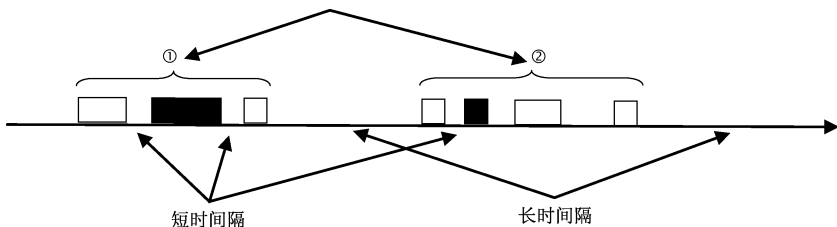


图 1 移动终端典型网络通信行为

Fig. 1 Typical network communication behavior of smartphone

本文提出移动智能终端下针对 3G 网络的低功耗延迟唤醒策略, 对网络数据包进行延时传输敏感性划分, 将延迟可容忍数据包缓存处理并进行延时传输, 以此延长 3G 网络的睡眠状态时间, 如将突发数据①与突发数据②进行缓存并延时传输, 减少 3G 网络唤醒次数达到降耗目的.

## 1 研究背景

移动网络的能耗问题研究主要集中在骨干网及数据中心能耗、基站能耗、智能终端能耗、移动应用及服务能耗等几个方面. 其中针对智能终端能耗、移动应用能耗的问题很早被业界关注, 典型如早期 ARM 公司<sup>[6]</sup>使用 Profier 工具对 Symbian 手机通过编译跟踪、指令覆盖等方式进行跟踪分析, 并指导设计低功耗应用程序. 苹果公司通过聚合技术将低层级操作进行组合, 形成短时间空闲间隔, 以使 CPU 更长时间处于非工作的低功耗状态. 同样, 苹果公司在其 2013 年发布的 OS X 10.9 Mavericks<sup>[7]</sup>操作系统中对架构进行设计, 增加 App Nap 功能用于监测后台运行的进程, 当某进程没有进行任何操作时, App Nap 会将该进程进行减速处理. 同时, App Nap 确定某进程应得到的系统资源(处理器、网络、磁盘等)等级, 并积极地

对不太重要的任务进行资源节流, 保持 CPU、硬盘及网络等尽可能在低功耗模式运行.

针对移动智能终端的 3G 网络低功耗设计也已经相继提出一些有效的方法<sup>[8-11]</sup>. Hossein 等<sup>[12]</sup>通过收集大量手机终端用户发送和接收的网络数据, 并对网络数据流量的组成、传输量的大小、TCP 传输质量及与电源管理关系等进行分析, 得出结论如下: 1) 浏览网页的流量占总流量的一半; 2) 小数据量的发送会造成更大的能耗; 3) 数据包丢失是造成高能耗的主要原因; 4) 状态转换时的尾端切换会造成不必要的能耗. 该文对移动终端网络通信行为进行研究, 给出较明确的数据统计及理论分析, 但针对问题却没有给出有效的解决方法. 普度大学 Abhinav Pathak 与微软联合开发 Eprof 能耗分析工具<sup>[13]</sup>, 以进程为粒度进行能耗的监测, 该小组发现 3G 网络中尾部能耗<sup>[14]</sup>带来了极大的电量消耗, 并在后续工作中给出针对尾部能耗的解决方法. Christian 等<sup>[15]</sup>通过分析网络部件能耗与 Web 体验质量(QoE, Quality of Experience)的协调关系并进行判断, 得出网络数据包的合理发送时间, 文中针对游戏 Angry Bird 进行验证, 该方法平衡了用户体验与移动智能终端系统能耗关系, 在一定程度上可降低能耗, 但在

某些场景下较难计算网络数据包的合理发送时间,存在一定局限性. Ekhiotz Jon 等<sup>[16-17]</sup>分析移动终端的后台数据通信行为,并给出网络流量控制框架及关键算法的描述,给出较为完整的功耗优化流程,其算法在实现过程中对网络数据包进行了大、小流量的划分,并给出大、小流量的最迟发送时间,并在最迟发送时间之前对数据进行传输处理. 但对于每一个数据包的最迟发送时间,仍然没有给出较为合理的推算方法.

本文在上述文献基础上,通过大量的实验数据进行分析,给出针对移动智能终端 3G 网络进行低功耗优化策略,并通过实验环境进行测试,本文优化策略对当前典型移动应用程序可有效降低 3G 网络部件的能耗.

## 2 3G 网络工作模式回顾

随着移动通信技术和互联网技术的快速发展,使得移动网络呈现出异构化的发展趋势,如 3G、WiMax 等网络系统被竞相提出. 本节首先介绍 3G 网络组成及其工作模式<sup>[18-20]</sup>,3G 网络可以在不同的工作模式下,完成不同速率的网络传输. 同时,本文提出的功耗优化策略通过 3G 网络的工作模式转换进行实现,在本节同时对网络工作状态的识别方法进行介绍,即分别通过 2 种方式对当前 3G 网络所处的工作状态进行判断.

### 2.1 3G 网络工作模式

3G 网络主要由 Core Network (CN, 核心网)、Universal Terrestrial Radio Access Network (UTRAN, 无线接入网)、User Equipment (UE, 用户设备) 3 部分组成. UE 端主要有 2 种网络连接模式,分别为空闲模式及连接模式. 其中,空闲模式下 3G 网络处于 IDLE 状态. 而连接模式下,具体又区分为 4 种工作状态: CELL-DCH (Dedicated Channel)、CELL-FACH (Forward Access Channel)、CELL-PCH (Paging Channel) 和 URA-PCH (Paging Channel) 状态. 每一种工作状态下具有不同的传输速率及功耗. 如在 IDLE 模式下,UE 端不进行数据收发处理,即不与 UTRAN 接入网建立连接. 而在连接模式下,3G 网络连接状态又根据传输数据量大小、上下行的速率分配等关键因素进行进一步工作状态设定. 如当 UE 在 CELL-DCH 状态工作时,UE 处于激活状态,且与接入网络建立有效连接,并利用上下行专属信道进行通信传输.

3G 网络中,不同工作模式之间的转换由无线资源管理协议 RRC (Radio Resource Control) 控制,如图 2 所示,典型有两态转换 (IDLE、CELL-DCH) 与三态转换 (IDLE、CELL-FACH 及 CELL-DCH). 在 DCH 模式下 UE 处于连接模式,该模式提供较高的、专属的上下行信道进行通信. CELL-FACH 状态下 UE 同样处于连接模式,但上下行只有少量数据通过公共信道进行传输. 因此,当处于 DCH 状态时,如未有数据包进行传输或传输的数据包较小,则经过一段时间 ( $t_1$ ) 网络连接状态会切换至 FACH,在 FACH 状态如一段时间没有数据包传输,则经过一段时间 ( $t_2$ ) 切换至 IDLE 状态. 如有数据包传输则再次迁移至 DCH 状态. 在两态模式切换下,网络从 CELL-DCH 直接转换到 IDLE 状态,可以理解为是一种深度睡眠机制,该机制在一定程度上可以降低转换时带来的能耗开销. 而三态模式切换下,是一种缓慢睡眠机制,该机制避免了深度睡眠后,突发数据的传输导致能耗的激增. 综上,两态模型与三态模型有各自适合场景及优势. 本文以两态场景为基础实现优化机制.

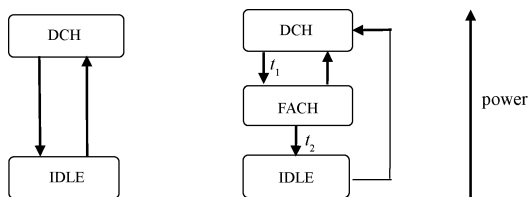


图 2 3G 网络工作模式

Fig. 2 Radio resource control state for 3G

### 2.2 3G 网络工作状态识别方法

3G 网络不同的工作模式具有不同的数据传输率,且对应不同的功率. 针对该特性,本文通过 2 种方式对当前终端所处 3G 传输信道 (工作模式) 进行验证,分别通过测量 RTT (Round-Trip Time) 值及测量功耗对 3G 网络所处工作模式进行识别. 第 1 种方式中,RTT 时间值为数据包发送端从发送该数据包开始,到接收到服务器端发送的确认帧所用的往返时间. RTT 值具体由 3 个部分决定: 3G 网络链路传播时间、服务器端的处理时间以及路由器的缓存中的排队和处理时间. 其中前 2 部分的值作为同一个 TCP 连接相对固定,而路由器中缓存的排队和处理时间会随着整个网络拥塞程度的变化而改变. 因此,一段时间的平均

RTT 变化在一定程度上反映了数据传输速率的变化,为了降低网络通信中的随机抖动,本文通过加权平均方法对测试中的 RTT 样本值进行了平滑处理.第 2 种方式中,3G 模块部件的瞬态功耗值同样反映 3G 网络所处的工作模式,对于处于不同传输信道的设备其功率不同,可以通过功耗仪测试其瞬态功率进行工作状态识别.

本文对 2 种方式分别进行了测试,从而分析 3G 模块所处的工作模式区别.首先,通过发送不同大小数据测试其 RTT 值,结果如图 5 所示,其

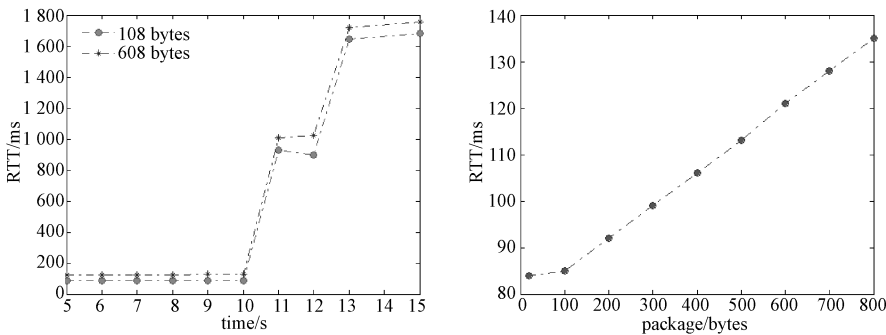


图 3 3G 网络状态切换分析  
Fig.3 State switching of 3G network

其次,本文通过测量通信模块的瞬态功耗对其所处的 3G 网络工作模式进行识别,分别测量大数据包与小包数据在不同时间间隔下稳定时的功耗情况,测量数据如表一所示,其中,当有数据传输时,1)所测试设备中 3G 通信模块满足两态模型,DCH 对应功率为  $5.9 - 4.8 = 1.1\text{ W}$ ,而 IDLE 对应功率可近似为  $0\text{ W}$ .2)两态模型下,发送数据包的大小对通信模块切换状态的改变没有影响.因此,当有数据包发送时,3G 模块即从 IDLE 状态切换至 DCH 状态.3)由 DCH 切换到 IDLE 的 Inactivity timer 的  $T_{\text{DCH}}$  值为 13 s.(注:通信模块无数据传输时功耗稳定在  $4.8\text{ W}$ )

表 1 不同大小数据包的功耗测量  
Table 1 Power consumption measurement  
for packages with different siges

发送时间 间隔/s	小数据包/108 bytes	大数据包/608 bytes
5	5.9	5.9
10	5.8~5.9	5.9
12	5.8~5.9	5.8~5.9
13	5.7->5.4->4.8->上升	5.7->5.0->4.8->上升

中 11 s 与 12 s 之间为从 IDLE 到 DCH 状态切换过程,接近 13 s 传输信道已完全迁移为 IDLE 状态.本测试环境为两态模型,RTT 时间呈现 3 种状态,即为 IDLE 状态、IDLE 转换至 DCH 的跃迁状态(11 s 与 12 s 左右)及 DCH 状态.同时测试了数据包的大小对于 RTT 值的影响,由于区分数据包大小的界限为 500 bytes,本文将测试数据包的大小设计为 608 bytes 及 108 bytes,从图 3(a)可得,2 种数据包发送时 RTT 值相似,即 3G 网络并未对数据包的大小进行区分处理.

### 3 非延迟敏感数据的 3G 网络功耗优化

3G 网络通过切换工作状态达到功耗和性能之间平衡.如 UE 端当前处于 CELL-DCH 状态,如在一定时间内无数据进行传输,则 RRC 协议将 UE 端网络的工作状态切换至 CELL-FACH 状态或 IDLE 状态.这种方式在一定程度上控制通信模块长期处于高能耗状态.但是,当有数据包发送时,即使小容量数据包也会导致 3G 网络的工作模式由低能耗 IDLE 状态切换至高能耗 DCH 的状态,类似频繁切换导致了大量额外能耗的产生.本节将介绍 3G 网络下基于数据重要性的 3G 网络延迟唤醒策略.重点介绍了实现策略中对数据包重要性的划分、延迟发送时间(等价于 3G 网络延迟唤醒时间)度量、优化策略的具体流程及数据缓冲实现等.

#### 3.1 网络数据特点及重要性划分

移动终端应用程序对网络传输的实时性需求在一定程度上是不同的.如新闻浏览、社交工具等前台应用直接涉及用户体验,通常对实时性要求相对较高.如游戏应用、音乐应用等所包含的后台



广告信息的数据交互,通常对时效性要求较低.而某些应用在移动终端处于睡眠状态同样会周期性地传输数据(广告、信息推送、同步数据等),该类后台数据通信行为通常具有小容量数据、周期性通信等特点,该类应用对数据传输的实时性需求可进一步降低.因此,对网络数据可按实时性需求具体分为 3 种,分别为前台数据、后台数据及睡眠状态数据.本文在优化策略实现过程中,分别对网络数据包进行传输重要性划分,并根据不同类型进行相应的延时传输处理.该方法可在一定程度上减少 3G 网络工作状态切换次数,降低 3G 部件处于高能耗状态的时间,从而降低功耗.

### 3.2 延迟时间度量

文献[15]和[17]中,分别给出了相应的延迟时间计算方法,但在某种程度上仍然难以估计具

体的延时发送时间.如图 4 所示,本文针对大量应用程序(如新浪微博、腾讯 QQ、酷狗音乐等)分析其睡眠状态下后台数据的网络通信行为,得出典型的睡眠状态下网络数据发送特点,并据此设定数据延时传输时间.如新浪微博客户端在后台运行时发送少量数据包,且超过 85% 的数据包小于 100 bytes,通过 ethernal 抓包工具获取数据包后,分析为保持 Keep-Alive 连接信息的数据包.如 Kugou Music 应用程序在睡眠状态下执行过程中,后台数据会以 600 s 左右的时间间隔进行数据的传输处理,通过分析可得,该数据为获取用户喜好、更新列表等操作.另如,腾讯 QQ 客户端在开启阶段经过大量数据通信后,后续以 500 s 左右时间规律与服务器进行通信.

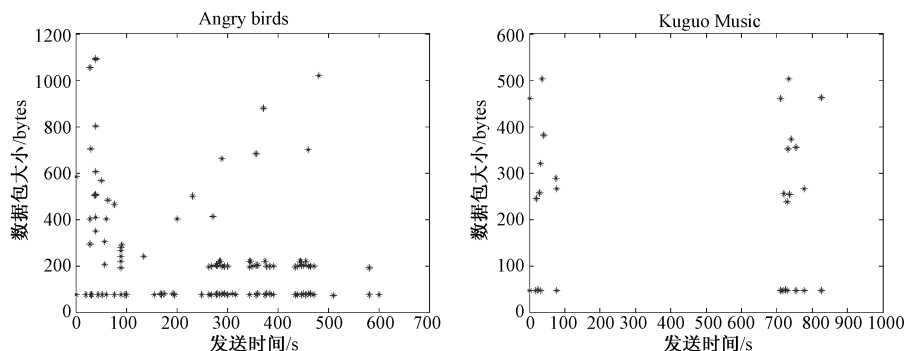


图 4 Angry Bird 与 Kuguo 音乐应用程序通信行为

Fig. 4 Communication behavior of Angry Bird and Kuguo Music

由于应用程序与服务器进行周期性数据通信,同时各应用程序在睡眠状态下通信周期性时间不同.通过长期测试发现,当某些程序数据缓存时间过长,程序会存在数据重发现象,相同数据包数量会随着缓冲时间增长而增加.另外,发送至客户端的数据在 NAT 端的缓存时间具有时间限制,因此随着 UE 端数据包缓存时间的延长,导致某些服务器端发送的推送消息也易导致错过接收.因此,通过大量实验数据分析,并考虑不同应用程序的通信机制及程序自身特点,本文对缓存时间根据不同应用设置为 2~10 s(3G 网络延迟唤醒)时间长度,根据不同应用进行动态调整.

### 3.3 3G 功耗优化实现流程

针对睡眠状态下数据周期性交互的特点,本文提出针对后台数据 3G 网络可延迟唤醒的低功耗策略,具体步骤如下:

1) 应用层通过套接字函数向底层发送网络数据,在此过程中获得数据包进程对应的进程号,将该信息保存在套接字结构体中,并向下传递至 sk\_buff 结构体,最后可在 Netfilter<sup>[21]</sup> 钩子函数中获得该进程号,可在此具体区分该数据报所属进程类型(前台进程、后台进程、睡眠状态进程);

2) 前台、后台进程数据对实时性要求较高,将其直接传输.睡眠状态下网络数据依次加入到缓冲队列并记录时间,当缓冲队列加入第一个数据包时开启定时器 T1,如 3.2 节描述,根据实验数据并综合考虑丢包率及服务器端推送消息的实时性对定时器值进行设置,同时,定时器值设置需根据后台应用程序的类型进行动态调整.该定时器用于控制被缓冲的数据何时发送.

3) 定时器 T1 时间到达后进行数据集中发送,此时 3G 网络迁移至 DCH 状态,由于数据包

集中发送后,可能有小量网络数据发送(如图 5 中 P1),该小量数据易导致网络处于 DCH 状态,引起尾部能耗,此时设置定时器  $t_2$ ,在  $t_2$  时间内接收到的的小容量量数据包立即发送,如  $t_2$  时间后接

收到的网络数据包(如图 5 中 P2),则进行缓存处理,在后续周期进行聚集发送.

4) 聚集发送完成,网络进入 IDLE 状态,则再次开启定时器 T1,再次进行缓冲操作.

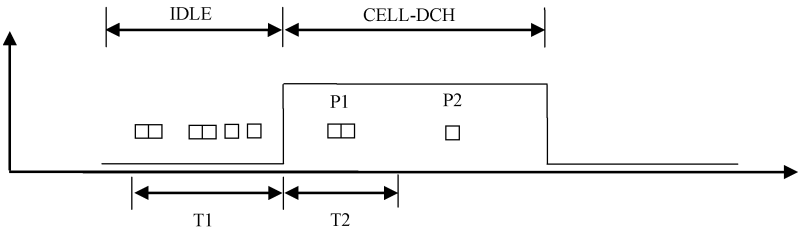


图 5 数据延时发送过程

Fig. 5 Process of package delayed sending

### 3.4 数据缓冲队列的实现

缓冲队列实现是通过 NetFilter 框架<sup>[21]</sup>的 NF\_QUEUE 机制,且带有 list\_head 的自定义结构体实现. 功耗管理模块的架构的主要部分是集中在对缓冲队列的管理上. 包括数据包入缓冲队列管理,出缓冲队列管理,及传输信道状态记录的定时器.

缓冲队列实现的伪代码如下

```
当一个数据包( skb )经过 netfilter 框架的
NF_IP_POST_ROUTING 钩子处理函数
1 IP_Header ← Get_IP_Header( skb )
2 if protocol( IP_Header ) = TCP do
3     geted_oom ← Get_oomadj_byPid( skb. sk. PID )
4     if geted_oom > -20 do //Get the available oomadj
5         return NF_QUEUE
6     else
7         return NF_ACCEPT
8     end if
9 end if
10 return NF_ACCEPT

数据包返回 NF_QUEUE 重新排队
Input: struct nf_queue_entry *entry, unsigned int queueenum
1 add_to_list( entry )
2 TotalNumberInList ← TotalNumberInList + 1
3 if should_sending = 1 and PassedTimeSinceStartSend
  < EXPIRE_TIME do
4     WakeUpSendThread( )
5 end if
6 if should_sending = 1 and PassedTimeSinceStartSend >=
  EXPIRE_TIME do
7     StopSendingTime ← GetStopSendingTime( )
8     should_send = 0
9 end if
```

据包,有 2 个时机可将排队中数据包重新发送出去:1)数据开始发送进行计时,在 EXPIRE\_TIME 时间内将所有数据发送出去. 2)数据缓存一段时间再发送,该过程体现在定时器伪代码中.

```
发送线程
1 while TURE do
2     TotalNumberInList ← GetTotalNumberInList( PacketList )
3     SetTotalNumberInList( PacketList, 0 )
4     while TotalNumberInList < > 0 do
5         Send the packet in PacketList
6         TotalNumberInList ← TotalNumberInList-1
7     end while
8     SuspendThread( )
9 end while
```

在发送线程中,得到队列中数据包的大小 TotalNumberInList,然后将该数量数据包一次性发送. 最后,该线程进入休眠态,等待唤醒.

```
定时器所设定时间长度决定了数据包在缓存队列中的最长时间
1 if TotalNumberInList > 0 and shoud_send = 0 do
2     if PassedTimeSinceStopSend > = DEADLINE do
3         SetStartSendTime( )
4         should_send ← 1
5         WakeUpSendThread( )
6     end if
7 end if
8 ResetTimer( )
//Timer will become unavailable if it runs, so reset it.
```

### 4 实验验证

为了验证基于后台周期性数据的 3G 网络功耗管理对功耗和性能的影响,本文在 Pandaboard 实验平台上实现了本文的优化策略,并对典型的

struct NF\_queue\_entry 封装已在排队中的数

应用程序(新闻类、视频类、音乐类等)进行了验证测试,实验结果显示针对前台数据、后台数据、睡眠状态数据划分为指导的数据包缓冲延时发送

机制,能够在不影响用户体验的情况下,达到一定的降耗目的,以下为具体实验数据及分析.

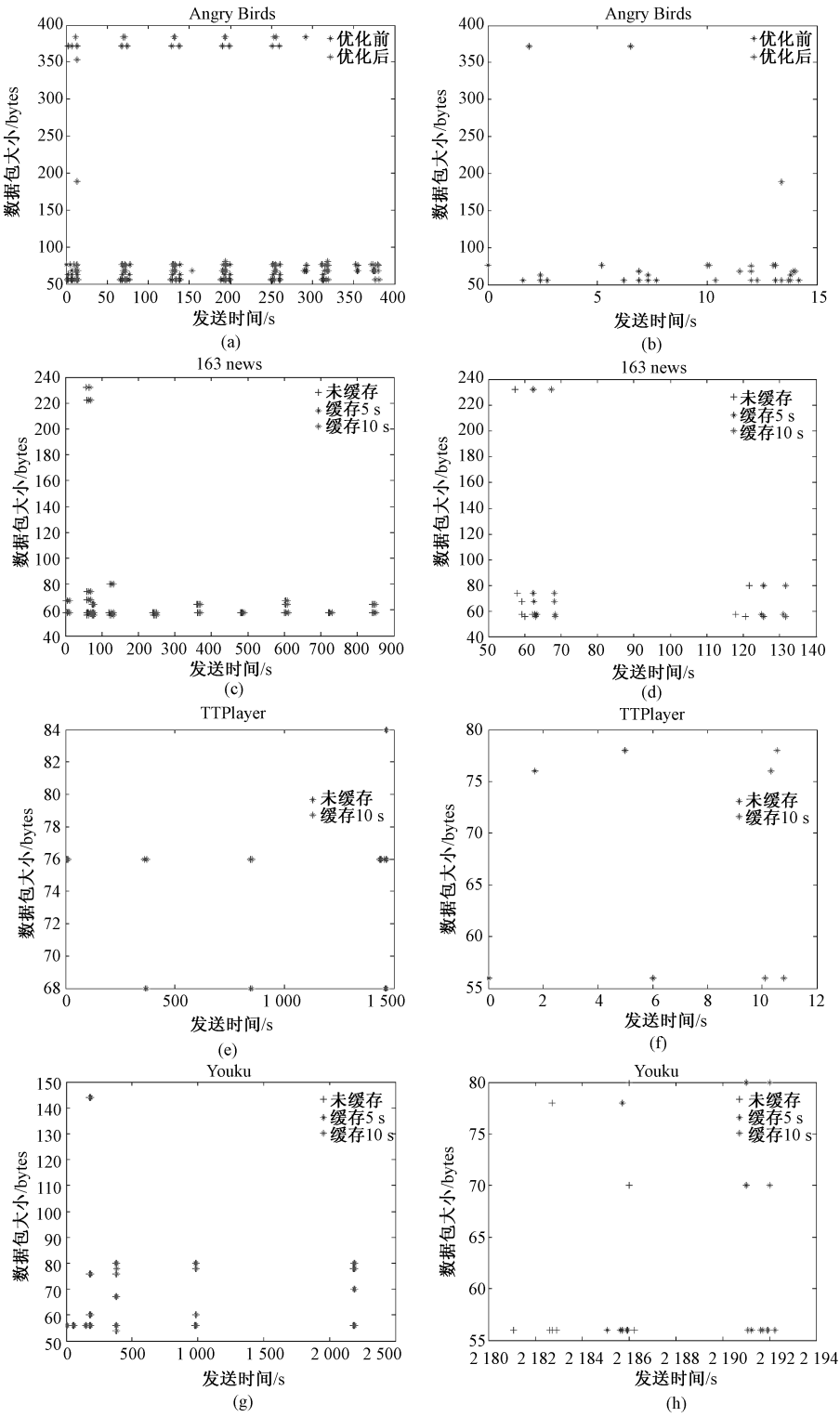


图 6 应用程序优化后效果  
Fig. 6 Optimization effects of different applications

为验证本文提出的优化策略,本文分别对几种典型应用程序(游戏类、新闻类、视频类等)进行了测试验证.实验过程分别针对不同应用程序进行不同时间段的 5 次测试,每组测试时间长度 4 h 的程序运行,同时进行数据监测.通过对不同应用程序进行后台数据分析,可见游戏 Angry Bird 2 次突发数据间隔约为 50 s,而每次突发数据的持续时间约为 10 s.163 news 应用程序中 2 次突发数据间隔约为 100 s,每次突发数据的持续时间约为 5 s.分别对游戏 Angry Bird、新闻客户端 163 news、音乐类 TTPlayer 及 Youku 客户端分别进行数据缓冲,设定缓冲时间分别为 5 s,考虑不同应用在睡眠状态下数据发送周期不同,本文同时对新闻客户端 163 news 及 YouKu 客户端进行 10 s 的延时处理,分别对不同的数据序列进行对比.

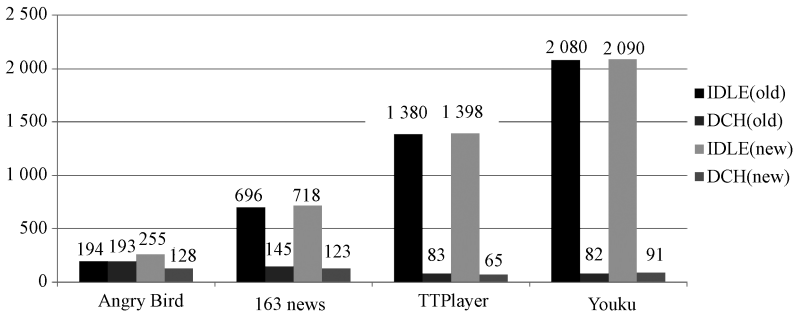


图 7 优化前后各状态时间对比  
Fig. 7 Total time in each mode

如图 7 所示,本文比较优化前后应用程序处于不同传输信道的时间.如 Angry Bird 应用程序,优化前 IDLE 状态持续时间为 194 s,DCH 状态持续时间为 193 s,优化后 IDLE 时间为 255 s,DCH

优化效果如图 6 所示,其中左侧一列(a)(c)(e)(g)为游戏 Angry Bird、新闻客户端 163 news、音乐类 TTPlayer 及 Youku 客户端 4 种应用程序的数据包发送序列情况,右侧(b)、(d)、(f)、(h)分别为对左侧结果数据进行局部放大,右侧效果图可见明显的数据聚集发送过程.如 Angry Birds 其睡眠状态下后台数据间隔 50 s 进行数据发送,数据发送持续时间为 10 s,本文对其进行延迟 10 s 处理,可得其初始功耗大概为  $(10 + 13) \times \text{DCH}$  状态的功耗,进行优化后功耗大约为  $(1 + 13) \times \text{DCH}$  状态的功耗,理论功耗降低 46%,而实际测得数据发送时间长度为 5 s,即优化后所得功耗为  $(5 + 13) \times \text{DCH}$  状态的功耗,实际功耗降低 30.8%.通过分析后台数据,数据包的缓存带来一定量数据重发现象,影响整体数据包发送规律,导致实际功耗与理论功耗出现一定偏差.

持续时间为 128 s, IDLE 状态时间从优化前 50.12% 提高至优化后 66.57%,而 DCH 状态时间从原优化的 49.48% 降低至 33.33%,有效降低了 3G 网络处于高能耗 DCH 状态的时间(表 3).

表 2 优化前后 DCH 和 IDLE 状态所占时间比例  
Table 2 Total time in DCH state and IDLE state

应用程序	DCH 状态(新)	DCH 状态(原)	IDLE 状态(新)	IDLE 状态(原)
Angry Bird	49.88	33.43	50.12	66.57
163 news	17.25	14.63	82.75	85.37
TTPlayer	5.67	4.44	94.33	95.56
YouKu	4.17	4.67	96.21	95.83

通过对不同类型应用程序睡眠状态下后台数据包缓存效果对比分析,针对不同应用其功耗降低效果也会不同,原因在于不同应用后台数据发送数据包呈现不同规律,当应用程序发送数据较频繁,数据缓存对降低功耗的效果较为明显,缓存

数据减少通信模块处于高功耗状态时间.对于后台数据较少的应用,缓存数据包对功耗的影响有所降低.如通过本文优化策略 Angry Bird 使得网络处于 3G 下 IDLE 状态下的时间提高 16.15%,而 YouKu 视频客户端应用程序使得 3G 网络下



IDLE 状态时间提高 0.5%.

5 总结

本文通过分析应用程序后台数据的行为规律<sup>[23]</sup>,提出基于后台周期性数据延迟发送的功耗优化策略.通过区分网络数据为前台数据、后台数据及移动终端睡眠状态下的后台数据,根据数据类型进行一定时间的缓存、延时传输处理.通过降低 3G 模块处于高能耗 CELL-DCH 状态的时间,达到降低功耗的目的.通过实验验证及数据分析,可见本文提出策略有效提高 3G 网络模块处于低功耗的 IDLE 状态时间,对于功耗优化有明显效果.目前本文工作主要集中在睡眠状态下的后台数据传输优化,下一步工作将重点分析前台应用程序的网络数据通信行为,统计数据传输时间及规律,以便进行动态的缓冲调优及算法实现.

参考文献

[ 1 ] Kim S, Kim H K, Kim H J. Climate change and ICTs [ C ] // Proceeding of Telecommunications Energy Conference, 2009:1-4.

[ 2 ] Wang X F, Vasilakos A V, Chen M, et al. A survey of green mobile networks: opportunities and challenges [ J ] Mobile Networks and Applications, 2011, 17 ( 1 ) : 4-20

[ 3 ] Hepburn A. Infographic: 2013 mobile growth statistics [ EB/OL ] (2013-10-01) [ 2014-09-25 ]. <http://www.digitalbuzzblog.com/infographic-2013-mobile-growth-statistics>.

[ 4 ] Google Company. Android SDK [ EB/OL ]. ( 2008-07-09 ) [ 2014-09-25 ] <http://www.android-doc.com/reference/packages.html>.

[ 5 ] Balasubramanian A, Balasubramanian N J, Venkataramani A. Energy consumption in mobile phones: a measurement study and implications for network applications [ C ] // Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. 2009: 280-293.

[ 6 ] ARM Company. ARM Technical Support Knowledge Articles [ EB/OL ]. (2011-05-01) [ 2014-09-25 ]. <http://infocenter.arm.com/>.

[ 7 ] Apple Company. OS X Mavericks is more than powerful. [ EB/OL ]. ( 2013-07-01 ) [ 2014-09-25 ]. <http://www.apple.com/osx/advanced-technologies/>.

[ 8 ] Perrucci G P, Fitzek F, Sasso G, et al. On the impact of 2G and 3G network usage for mobile phones'battery life [ C ] // Proceedings of the European Wireless Conference ( EW ), 2009:255-259.

[ 9 ] Armstrong T, Trescases O, Amza C, et al. Efficient and transparent dynamic content updates for mobile clients [ C ] // Proceedings of the 4th International Conference on Mobile Systems, Applications and Services ( MobiSys'06 ), 2006.

[ 10 ] Qian F, Wang Z, Gerber A, et al. Profiling resource usage for mobile applications: a cross-layer approach [ C ] // Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services ( MobiSys'11 ), 2011: 321-334.

[ 11 ] Palit R, Naik K, Singh A, et al. Impact of packet aggregation on energy consumption in smartphones [ C ] // Proceedings of the 7th Inter-national Wireless Communications and Mobile Computing Conference ( IWCMC ), 2011: 589-594.

[ 12 ] Hossein F, Dimitrios L, Ratul M, et al. A first look at traffic on smartphones [ C ] // Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement Melbourne, Australia ( SIGCOMM ), 2010.

[ 13 ] Pathak A, Hu Y C, Zhang M, et al. Where is the energy spent inside my app? fine grained energy accounting on smartphones with eprof [ C ] // Proceedings of the ACM European Conference on Computer Systems ( EuroSys ), 2012: 29-42.

[ 14 ] Liu H, Zhang Y, Zhou Y, et al. TailTheft: leveraging the wasted time for saving energy in cellular communications [ C ] // Proceedings of the 6th International Workshop on MobiArch, ser. ( MobiArch ), 2011 : 31-36.

[ 15 ] Schwartz C, Hossfeld T, Lehrieder F, et al. Angry apps: the impact of network timer selection on power consumption, signalling load, and Web QoE [ J ]. Journal of Computer Networks and Communications, 2013.

[ 16 ] Vergara E J, Tehrani S N . Energy-aware cross-layer burst buffering for wireless communication [ C ] // Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, 2012:1-10.

[ 17 ] Vergara E J, Sanjuan J. Kernel level energy-efficient 3G background traffic shaper for android smartphones [ C ] // Proceedings of Wireless Communications and Mobile Computing Conference ( IWCMC ), 2013.

[ 18 ] Garcia J S. 3G Energy-efficient packet handling kernel module for android [ D ]. Link Oping University, 2012.

[ 19 ] Zhao B, Tak B C, Cao G. Reducing the delay and power consumption of web browsing on smartphones in 3G networks [ C ] // Proceeding of the 31st International Conference on Distributed Computing Systems ( ICDCS ), 2011:413-422.

[ 20 ] Schulman A, Navda V, Ramachandran, et al. Bartendr: a practical approach to energy-aware cellular data scheduling [ C ] // Proceedings of the sixteenth annual international conference on Mobile computing and networking ( MobiCom ), 2010.

[ 21 ] Netfilter.org. Linux netfilter Hacking HOWTO [ EB/OL ]. (2002 - 07 - 02 ) [ 2014 - 09 - 25 ]. <http://www.netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO.html>.

[ 22 ] Qian F, Wang Z G, Gerber A, et al. Top: tail optimization protocol for cellular radio resource allocation [ C ] // Proceeding of the 18th IEEE International Conference on Network Protocols ( ICNP ), 2010:285-294.