

DOI: 10.13382/j.jemi.B2104387

智能手机能耗多目标优化机制研究*

薛普俊 朱正伟 诸燕平 朱晨阳
(常州大学计算机与人工智能学院 常州 213164)

摘要:智能手机的网络请求导致其续航能力下降,合并转发技术可有效降低能耗,但设置最优的合并转发时间仍是技术发展的关键,通过大量人工实验解决该问题耗时耗力。因此,基于统计模型检验,使用工具UPPAAL-SMC,以概率时间自动机对安卓设备中用户请求以及WiFi模块进行仿真建模,量化能耗、延迟、用户满意度等属性,进而利用统计模型检验,对不同请求频率的场景进行蒙特卡洛模拟,获得延迟对能耗以及用户满意度的影响。最终进行多目标优化,求得通用最优合并转发延迟时间为22 s,在满足用户体验的前提下平均降低了20%的能耗。该方法可在不同的使用场景下模拟得到通用最优合并转发延迟时间,为开发者提供参考。

关键词:合并转发;WiFi模块;统计模型检验;多目标优化;延迟时间

中图分类号:TP31;TN91 **文献标识码:**A **国家标准学科分类代码:**520

Research on multi-objective optimization mechanism of smartphone energy consumption

Xue Pujun Zhu Zhengwei Zhu Yanping Zhu Chenyang

(School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213164, China)

Abstract: The network request of smart phone leads to the decline of its endurance. The combined forwarding technology can effectively reduce energy consumption. But setting the optimal combined forwarding time is still the key to technological development. And it is time-consuming and labor-intensive to solve this problem through a large number of manual experiments. Therefore, this paper based on statistical model checking and used the tool UPPAAL-SMC to conduct simulation modeling of user requests and WiFi modules in Android devices with probabilistic time automata. Quantifying energy consumption, delay, user satisfaction and other attributes. And then used statistical model checking to conduct Monte Carlo simulation of scenarios with different request frequencies to obtain the effect of acquisition delay on energy consumption and user satisfaction. Finally, multi-objective optimization was carried out to obtain the general optimal combined forwarding delay time of 22 s, which reduced energy consumption by 20% on average on the premise of satisfying user experience. This method can simulate the general optimal merge forwarding delay time in different application scenarios, which can provide reference for developers.

Keywords: the combined forwarding; WiFi module; statistical model checking; multi-objective optimization; delay time

0 引言

随着科技的进步,智能手机快速发展,但物理尺寸以及电池技术限制了其续航能力,锂电池的寿命^[1-2]更是一大困扰。智能手机的各种APP进行网络请求更新数据的操作更加快了智能手机的耗能^[3]。深入研究并提高智

能手机的节能技术将会大大提升智能手机的续航能力^[4]。事实上,智能手机的网络请求通过移动网络3G/4G^[5]或是WiFi模块^[6]实现,为了更好地节省能源,无线装置会在一定条件下转入睡眠等低功率状态,待有网络请求时再恢复高功率状态。但一些零散的网络请求使得无线装置处于高功率状态的时间延长,导致其功耗增加。此情况通常通过阻塞部分网络请求合并转发的方式解

决。智能手机中不同的网络请求需要的响应速度不同,对无需及时响应的请求,可以适当进行延迟,与后续请求一起合并转发,从而减少 WiFi 模块处于高功耗状态的时间,达到降低能耗的效果。

关于智能手机能耗,有关学者进行了大量研究。Li 等^[7]发现,在 HTTP 请求中,发送 1 Byte 和 1 000 Byte 消耗的能量大致相同,由此提出将多个 HTTP 请求捆绑发送^[8],从而形成一种自动检测请求绑定并进行自动绑定的方法^[9]。但捆绑 HTTP 请求可能会造成延迟,对用户体验产生一定影响。Huang 等^[10-11]通过对定期的网络请求进行研究,发现其请求的内容通常是一些推送消息以及用户行为追踪等,这类请求可以进行合并转发。此外,研究手机屏幕开启以及关闭时网络请求的特点发现,屏幕开启时,所发出的请求应当立即响应;屏幕关闭时,有些请求则不必立即响应,后者可以进行合并转发。Huang 等的优化策略对屏幕开启时的请求施加了严格的约束,即不允许有任何延迟。但从节能的角度出发,实际应用中屏幕开启时仍有大量的请求是可以被延迟的。并且, Huang 等的这种严苛的优化策略不具有普适性。文献[12]基于 Huang 的研究,提出 DelayDroid 转换系统,可以自动将非合并转发请求的应用转换为合并转发请求应用,转换后的应用支持多应用的网络请求调度,但仅可在屏幕关闭时进行调度,使用场景较为单一。文献[13]基于用户的使用习惯对应用程序的网络活动进行了重要性标记,通过多背包算法建立了应用程序网络请求最优控制的数学模型。该方法虽然兼顾了用户体验,但并没有对用户体验进行量化,且需要大量的用户使用数据,比较耗费时间。

先前研究并没有权衡用户体验给出合理的延迟时间。通过实际测量能耗计算合理的延迟时间费时费力,而使用统计模型检验可以方便地得出结论。文献[14]使用概率时间自动机^[15]对多房间供暖问题进行建模,并利用统计模型检验模拟得出权衡能耗和用户不适感的最优温度设置阈值。文献[16-17]对 WiFi 模块进行了自动机的建模,但并没有对用户请求行为进行模拟。

本文参考安卓开发文档,使用概率时间自动机对安卓设备中网络请求对 WiFi 模块的能耗影响进行建模,量化网络请求过程中能耗、延迟以及用户满意度等属性。同时使用统计模型检验的蒙特卡洛模拟,对安卓设备中请求频率较低的场景,模拟合并转发网络请求的延迟对能耗以及用户满意度的影响。在此基础上权衡能耗和用户满意度,进行多目标优化,最终在多个场景的帕累托最优值中得到通用的最优延迟时间。

1 理论背景

1.1 安卓 WiFi 模块功率状态转换机制

安卓开发文档中详细描述了在 3G/4G 网络下安卓智能手机的能耗特点,并列举了一些优化网络下载降低能耗的方法。对于无线装置的一般原则和对应的优化网络下载的方法同样适用于安卓智能手机 WiFi 模块。基于此,将 WiFi 模块的功率状态转换机制以及降低能耗的方法介绍如下。

由图 1 所示, WiFi 模块共有深度睡眠、轻度睡眠、空闲监听、高功耗 4 个功率状态,当用户需要发出一个请求时,会转移至高功耗状态。若请求获得唤醒锁或短时间内仍有新的请求则转移至空闲监听状态,否则转至轻度睡眠状态。在空闲监听状态等待 5 s 后若唤醒锁已释放则转入轻度睡眠状态,再空闲 10 s 则会转入深度睡眠状态。在由深度睡眠和轻度睡眠转移至高功耗状态的过程中分别会有一定的延迟。由于不同设备使用的无线技术和 WiFi 模块的差异,延迟时间和空闲时间可能会不同。本文采用文献[18]所测量的功率值,依次为 0.01 W (深度睡眠)、0.12 W (轻度睡眠)、0.40 W (空闲监听)、0.60 W (高功耗)。

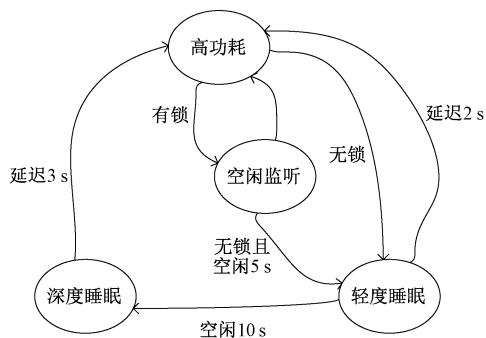


图 1 WiFi 状态转移图

Fig. 1 WiFi state transition diagram

若用户每发出一个请求都立即进行转发, WiFi 模块会立即转入高功耗状态, 经历一系列状态转换之后才会回到较低功耗的深度睡眠状态。若将用户请求阻塞一段时间, 进行推迟请求并将多个请求合并转发, 则会降低转入高功耗状态的次数, 从而降低能耗。但是合并转发以及状态转移带来的延迟会对用户体验造成影响, 所以需要权衡推迟请求的时间以及对用户产生的不适感, 找到一个最优的延迟时间。

1.2 概率时间自动机

概率时间自动机^[15]作为一种表现出概率、不确定性和实时特征的系统建模形式, 相对于普通自动机, 拓展了

状态停留的时间以及位置转移的概率。广泛应用于无线通信协议、汽车网络协议、随机安全协议等领域。

$\mathbb{R}_{\geq 0}$ 表示非负实数集, $\mathbb{Q}_{\geq 0}$ 为非负有理数集, $\mathbb{N}$ 为自然数集。可数集合 Q 上的离散概率分布是函数 μ , 即 $Q \rightarrow [0, 1]$ 使 $\sum_{q \in Q} \mu(q) = 1$ 。对于函数 $\mu(Q \rightarrow \mathbb{R}_{\geq 0})$, 定义 $\text{Support}(\mu) = \{q \in Q \mid \mu(q) > 0\}$, 其中 $\text{Support}(\mu)$ 是一个可数集合, μ 是一个分布。对于一个任意集合 Q , 存在 $\text{Dist}(Q)$ 为函数 $\mu(Q \rightarrow [0, 1])$ 的集合, $q \in Q$ 时, μ_q 表示 q 处的点分布, q 的概率为 1。AP 为一组原子命题。

时钟是一个在非负实数范围内变化的变量, 以与时间相同的速率增加。对于一个有穷时钟集 X , 函数 $v: X \rightarrow \mathbb{R}_{\geq 0}$ 则称为一个时钟解释^[15], 所有时钟解释的集合即 $\mathbb{R}_{\geq 0}^X$ 。对于任意 $v \in \mathbb{R}_{\geq 0}^X$, $v + \delta$ 表示 v 中所有时钟值增加 δ 的时钟解释, 其中 $\delta \in \mathbb{R}_{\geq 0}$ 和 $x \in X$ 。 $v[x := 0]$ 表示将 x 中的时钟重置为 0 的时钟解释。

定义 1(时钟约束^[15]) 对于一个时钟集 X , 时钟约束 φ 的集合 $\Phi(X)$ 定义为:

$$\varphi := \text{true} \mid x \leq d \mid c \leq x \mid x + c \leq y + d \mid \neg \varphi \mid \varphi \wedge \varphi \quad (1)$$

其中, $x, y \in X$; $c, d \in \mathbb{N}$ 。 $v \models \varphi$ 表示时钟解释 v 满足时钟约束 φ , 当用 $v(x)$ 替换每次出现的时钟 x 时, φ 解析为真。

定义 2(概率时间自动机语义^[15]) 概率时间自动机可以用八元组表示, 即 $PTA = \langle L, L^0, X, \text{Act}, \text{inv}, \text{enab}, \text{prob}, \omega \rangle$ 。其中, L 是一个有穷位置集合; $L^0 \in L$ 是一个初始位置; X 是一个有穷时钟集合; Act 是一个有穷动作集合; $\text{inv}: L \rightarrow \Phi(X)$ 是不变量条件; $\text{enab}: L \times \text{Act} \rightarrow \Phi(x)$ 是允许条件; $\text{prob}: L \times \text{Act} \rightarrow \text{Dist}(2^X \times L)$ 是概率转移函数(部分); $\omega: L \rightarrow 2^{AP}$ 是一个标记函数, 将每个位置映射到一组原子命题。

两个不同的概率时间自动机通过 CCS^[19] (calculus of communicating systems) 同步管道进行同步。CCS 作为一种并发系统模型, 是进程代数领域的开拓性工作, 属于一个函数式语言, 可以对并发的交互式系统进行模型化。

图 2 展示了 UPPAAL^[20] 中两个概率时间自动机的同步。 $p1, p2, p3, p4$ 分别为两个进程从 A 节点到 B 或 C 节点的概率。在这两个进程中, 通过 $a!$ 发布消息, $a?$ 接收消息, 从而使两个进程同时进行, 根据各自的概率到达下一个节点。

1.3 统计模型检验

统计模型检验是对系统进行有限数量的模拟运行, 并使用假设检验来推断样本是否为满足或违反规范提供了统计证据^[20]。它可以利用统计、仿真等概率学方法, 进行系统模型的随机仿真运行, 产生样本验证其系统性质是否满足。使用统计模型检验的蒙特卡洛模拟, 可以

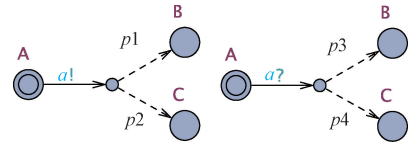


图 2 概率时间自动机同步

Fig. 2 Probabilistic time automaton synchronization

使计算机自动随机地进行模拟, 并且利用工具 UPPAAL-SMC^[20] 收集数据, 很大程度上节约了时间和精力, 并且可以计算无误差区间内信任值以提供数据支持。

2 基于概率时间自动机仿真建模

2.1 WiFi 模块模型

图 3 为安卓智能手机 WiFi 模块的模型。此模型主要负责 WiFi 模块不同功率状态间的状态转移, 以及与用户请求模型、中间控制模型进行通信, 接收转发请求的指令, 进行状态转移。在模型中有高功耗 (HighPower)、空闲监听 (IdleListen)、深度睡眠 (DeepSleep) 以及轻度睡眠 (LightSleep) 状态。采用 xDS 、 xLS 、 xIL 、 xHP 作为时钟变量, 以秒为单位分别记录 WiFi 模块在 4 个功率状态下所经过的时间。

在每个节点的不变量中, 时钟变量的导数为 1, 表示模型运行到此节点时, 此时钟变量计时; 时钟变量的导数为 0, 时钟变量则不计。如此可以在不同的节点分别记录不同功率状态所处的时间, 以便于计算 WiFi 模块所消耗的能耗。类似于式 (2) 的表示:

$$xDS' = 1 \quad (2)$$

依据 WiFi 模块每个功率状态所消耗的时间, 采用 1.1 节所提到的各个功率状态的功率, 最终计算出整个 WiFi 模块消耗的能量 $energy$, 单位为焦耳 (J)。用式 (3) 表示:

$$energy = 0.01 \times xDS + 0.12 \times xLS + 0.4 \times xIL + 0.6 \times xHP \quad (3)$$

WiFi 模块通过接收用户请求指令进行转发请求, 并返回至 HighPower 节点。同步动作 beacon? (与用户模板中的 beacon! 相对应), 表示接收到用户请求指令并进行状态转移。由于 DeepSleep 转移至 HighPower 节点和 LightSleep 转移至 HighPower 节点时会存在一定延迟, 所以分别设定一个过度状态进行 3 s 和 2 s 的延迟。由节点 Interim1 和 Interim2 用不变量 $\text{delayT} \leq 3$ 和 $\text{delayT} \leq 2$ 来完成延迟。使用方法 Transpond() 清空请求缓存队列表示完成发送请求。当前无请求需要转发时, 系统会根据不同的条件转入相应的节点。若当前缓存中有新的新请求等待转发, 转入 IdleListen 节点。在 IdleListen 节点时, 空闲 5 s 后, 若唤醒锁已释放, 转入 LightSleep 节

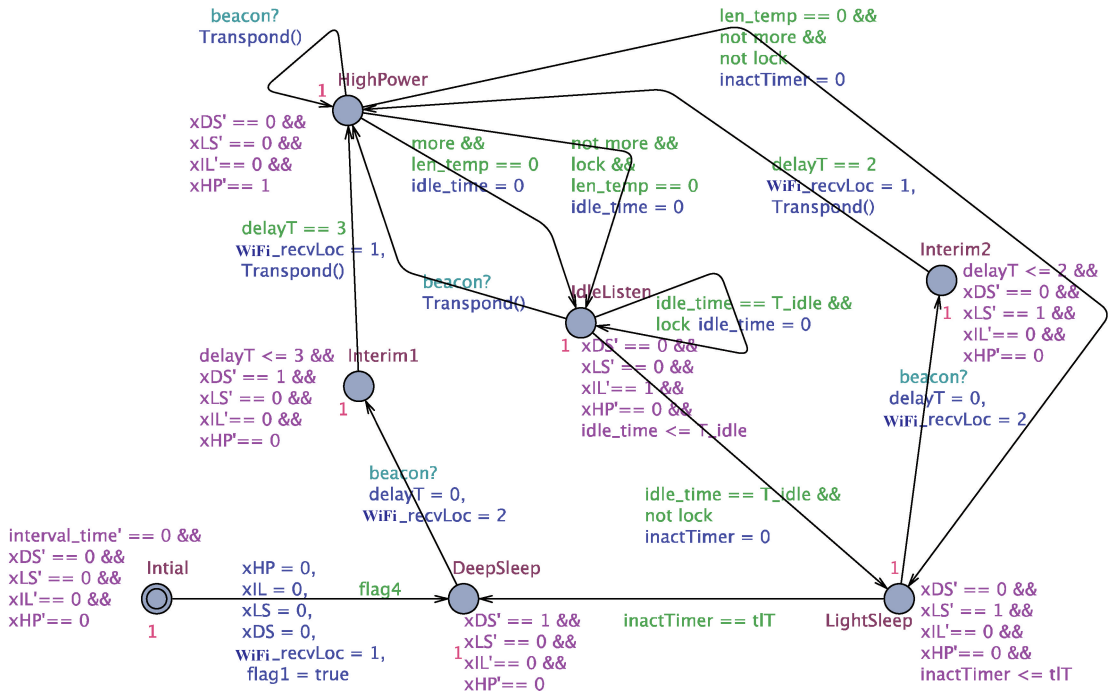


图 3 WiFi 模块模型

Fig. 3 WiFi module model

点,否则重新计时。在 LightSleep 节点时,若 10 s 内没有用户请求指令,会转入 DeepSleep 节点。

2.2 用户请求模型与阻塞用户请求模型

在实际使用安卓智能手机应用时,根据重要程度可将用户发出的请求大致分为两类:即时请求(QM),如访问网站等需要立即获得消息的操作;可以推迟发送的请求(PM),如天气、新闻、购物及娱乐软件等后台推送的消息。

鉴于以上两种情况,所以触发请求的情况有两种。

当前请求为 QM 时直接请求,同时将缓存中的请求一起发出;对于 PM ,当达到预定的延迟时间 T 时,再将缓存中的请求一起发出。关于缓存,模型中设置一个相应的循环队列存储用户的请求信息,循环队列可以不断进行循环的写入和删除。

基于以上描述,建立对用户请求的模拟,如图 4 所示,时钟变量 t 保证了模型每隔 $N(N \in [1, 10])$ s 运行一次,以此得到一个用户随机请求的模型。图中从 Interim 节点到 Transfer 节点采用设置的概率分布,分别进行 QM 、 PM 以及不进行请求(NM)3 种操作。节点中含有‘C’的状态为 committed 状态,此状态下没有时延。虚线为 UPPAAL 的概率表示,可以根据设定的概率随机选择路径,使用方法 Request(int m)来将用户请求存入缓存队列,并根据参数 m 判断请求类型($m = 1$ 为 PM , $m = 2$ 为 QM),对 QM 随机获得唤醒锁。图 4 中的概率为:

$$QM : NM : PM = 1 : 8 : 1 \quad (4)$$

图 4 中 QR 是一个 bool 值,为 QM 的标识,! QR 为 PM , NO 表示 NM ,用以说明请求类型。模型通过 beacon! 与 WiFi 模块模型进行通信,使 WiFi 模块进行状态转移。但由于 WiFi 模块在处于 Interim1、Interim2 两个节点时($WiFi_rcvLoc = 2$),正在进行延迟,无法完成同步,所以仅通过方法 updateQueueLength()更新缓存队列,待转入高功耗节点时转发请求即可。当 WiFi 模块模型处于 4 个功率状态节点时($WiFi_rcvLoc = 1$),可直接进行同步,进行状态转移。

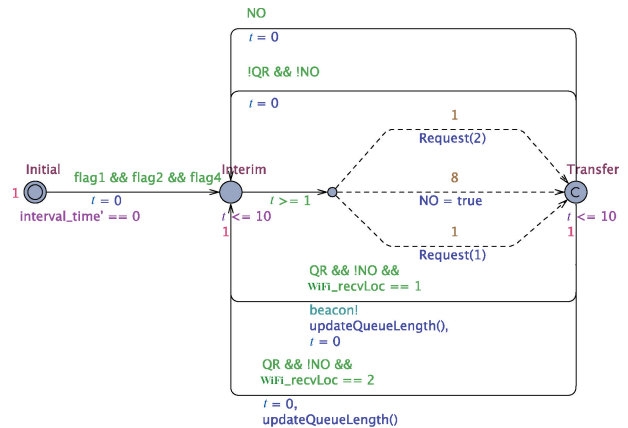


图 4 用户请求模型

Fig. 4 User request model

当请求类型为 PM 时,用户请求模型将请求存入缓存队列,待阻塞用户请求模型中间隔时间 $interval_time$ 达到 T 时,将请求发出。阻塞控制模型如图 5 所示。

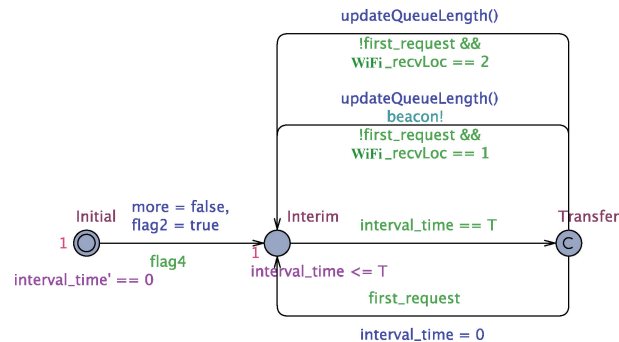


图 5 阻塞控制模型

Fig. 5 Blocking control model

2.3 其他模型

在 *QM* 中, 会有一部分请求需要获得唤醒锁, 在用户请求模型中, 用户发出需要即时响应的请求后, 系统以 1/2 的概率随机地使 *QM* 获得唤醒锁, 以此模拟现实中的状况。同样, 建立一个释放唤醒锁的模型, 每 1~10 s 随机进行一次运行, 以 1/2 的概率决定唤醒锁是否释放, 并通过 `updateLock()` 进行唤醒锁的释放(唤醒锁数目减 1), 当唤醒锁全部释放完成后, 布尔变量 `lock` 置为 `false`。唤醒锁释放模型如图 6 所示。

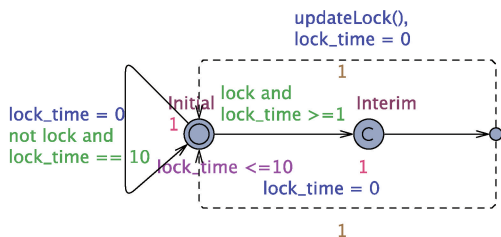


图 6 唤醒锁释放模型

Fig. 6 Wake up lock release model

全局模型如图 7 所示。通过语句 $t:delayRange$ 在定义的范围随机选择出延迟时间 T , 使用时钟变量 `discomfort_time` 记录不舒适感, 时钟变量根据导数值的大小会有不同的变化速率, 值越大, 随着时间变化越快。由 QM 带来的延迟一般为状态转移造成, 而 PM 带来的延迟主要是延迟时间 T 造成, 通常情况下 PM 相较于 QM 对用户体验造成的影响小。所以当请求类型为 QM 时, `discomfort_value = 1`; PM 时, `discomfort_value = 0.1`。

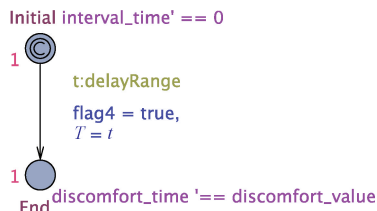


图7 全局模型

Fig. 7 Global model

行蒙特卡洛模拟并得到因为延迟时间 T 而产生的能耗和用户不适感数据。在众多延迟时间 T 中找出帕累托最优值(不唯一),从而找出最优延迟时间 T 。不同使用场景得到的最优延迟时间 T 不同,通过选取多个场景,分别进行模拟,最终找出选取场景中通用的延迟时间 T 。具体描述如下。

1) 分别对每个场景中的每个情况(如表1共24种情况)用UPPAAL-SMC生成数据,并使用方差分析(ANOVA)检验延迟时间 T 对能耗和用户不适感的显著性($P < 0.05$ 则显著,若不显著,返回生成更多数据),计算各种情况下每个延迟时间 T 对应的平均能耗和平均不适感,并找出各个情况的帕累托最优值

2) 为得到不同的偏好结果,将每种情况下得到的帕累托最优值的能耗和不适感进行加权相加。若能耗权值较大,则偏向于降低能耗。

3) 实验的目的是降低能耗和不适感, 所以将每种情况下的帕累托最优值中能耗和不适感加权和进行升序排序, 选取较低的 5 个加权和对应的延迟时间进行统计, 出现次数最多的即为通用延迟时间。

3.1 场景选择

采用 3 种智能手机的使用场景,分别如下:

1) 智能手机基本处于待机状态

大部分为天气类、新闻类、购物类等应用的一些后台请求信息,对响应速度要求较低,可以选择推迟。 NM 和 PM 的概率较高。

2) 用户在线看视频

一般情况下用户仅播放视频时很少有其他操作,但也不排除一些软件在后台有请求。播放视频时的网络请求为一段时间请求一次进行缓存。基于上述描述,此场景为在场景1的环境下,附加一个固定时间的请求机制。

3) 用户在线听音乐

此场景与场景 2 的区别在于用户有可能仅听音乐,或听音乐时也有一些其他操作(聊天、微博等)。因此,此场景中各类请求都有,并有固定时间的请求机制缓存音乐。

在每种场景中,不同类型请求的概率是变化的,所以在每个场景中选择 8 种不同的请求概率比值.在 24 种情

3 基于统计模型检验的数据分析

参考文献[14],使用UPPAAL-SMC作为查询引擎进

况下选出一个通用的最优延迟时间 T 。表 1 展示了不同场景对应的不同请求种类的概率比值。从表中可以看出,场景 1 和场景 2 中的概率比值相同,原因是场景 2 作为用户看视频的场景,相比于场景 1,多一个固定时间请求(固定 30 s 一次,模拟看视频时一段时间请求一次)的机制,其他用户请求情况与场景 1 类似。场景 3 中也存在一个固定时间请求机制。

表 1 不同场景的不同请求种类的概率比例

Table 1 Probability ratios of different types of requests for different scenarios

场景 1 (QM : NM : PM)	场景 2 (QM : NM : PM)	场景 3 (QM : NM : PM)
1 : 8 : 1	1 : 8 : 1	2 : 6 : 2
1 : 7 : 2	1 : 7 : 2	2 : 4 : 4
1 : 6 : 3	1 : 6 : 3	2 : 2 : 6
1 : 5 : 4	1 : 5 : 4	3 : 3 : 4
1 : 4 : 5	1 : 4 : 5	3 : 5 : 2
1 : 3 : 6	1 : 3 : 6	3 : 2 : 5
1 : 2 : 7	1 : 2 : 7	4 : 3 : 3
1 : 1 : 8	1 : 1 : 8	4 : 1 : 5

3.2 实验结果

延迟时间 T 取 11 ~ 60 s, 每次模拟都在 UPPAAL-SMC 中随机选择一个延迟时间来进行。统计参数概率不确定度 ε 为 0.05, 比率下限 u_0 为 0.9, 比率上限 u_1 为 1.1。在 UPPAAL-SMC 中模拟的命令如下:

```
simulate 6 400 [<=3 600] { T, 0.6 * Wi-Fi. xHP + 0.4 *  
Wi-Fi. xIL + 0.01 * Wi-Fi. xDS + 0.12 * Wi-Fi. xLS, discomfort_  
time } : 1 : false
```

模拟 3 600 s 内, 即 1 h 内产生的 WiFi 能耗和不适感, 每个场景的每种情况都进行 6 400 次的模拟, Wi-Fi. xHP、Wi-Fi. xIL、Wi-Fi. xDS、Wi-Fi. xLS 为 Wi-Fi 模块各个功率状态 1 h 内所经历的时间。在 UPPAAL-SMC 中延迟时间 $T=30$ s (场景 1 中 $QM : NM : PM=1 : 8 : 1$) 的能耗和不适感的模拟图如图 8 所示。

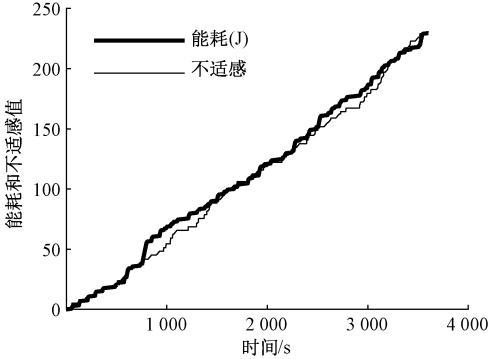


图 8 UPPAAL-SMC 一次模拟变化图

Fig. 8 One-time simulation change diagram of UPPAAL-SMC

在 6 400 次独立实验中, 每次随机生成一个范围为 [11, 60] 的延迟时间 T , 最终生成的数据以 $(T, energy, discomfort_time)$ 的形式来存储, 以 T 为分组, 共 50 组。在 6 400 个数据中平衡数据, 使每个 T 包含的数据量相同。

将表 1 中 24 个情况模拟 6 400 次, 平衡数据(保证每个 T 所对应的样本数相同)之后, 使用 R 工具进行方差分析(ANOVA)检验数据, 每个情况在运行 6 400 次之后数据都具有显著性。其中一个方差分析的结果(场景 1 中 $QM : NM : PM=1 : 8 : 1$)如表 2 所示。

表 2 R 工具方差分析结果

Table 2 R tool variance analysis results

分析对象	F 值	P 值	显著性
discomfort_time	8 407. 2	$<2.2 \times 10^{-16}$	***
energy	1 433. 7	$<2.2 \times 10^{-16}$	***

因为只有延迟时间 T 一个因素, 所以使用的是单因素方差分析的方法, 表 2 展示了因素 T 对不适感 discomfort_time 和能耗 energy 的方差分析, 第 3 列为 P 值, 最后一列为显著性的标记, 当 $0 < P < 0.001$ 时, 标记为 ‘***’; $0.001 < P < 0.01$ 时, 标记为 ‘**’; $0.01 < P < 0.05$ 时, 标记为 ‘*’; $0.05 < P < 0.1$ 时, 标记为 ‘.’; $P > 0.1$ 则没有标记, 显著性依次降低。当 P 值小于 0.05 时, 具有显著性。在表 2 中, 可以看出延迟时间 T 对不适感和能耗的影响 P 值都是远小于 0.05 的, 所以具有显著性。

在确定数据具有显著性之后, 求得每个延迟时间下能耗和不适感的均值。图 9 展示了其中一个场景下能耗和不适感随延迟时间 T 的变化图(场景 1 中 $QM : NM : PM=1 : 8 : 1$), 能耗和不适感均为均值。图中能耗的单位为焦耳(J), 不适感即对用户体验造成影响的程度用数值反映。从图中可以看出, 随着延迟时间的增加能耗减少, 而不适感不断上升。该结果符合预期, 即找出能耗和不适感的一个折衷。

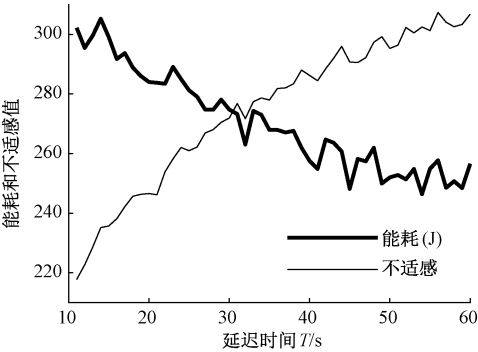


图 9 能耗和不适感随延迟时间 T 的变化图

Fig. 9 Change of energy consumption and discomfort with delay time T

通过均值求得表 1 中 24 种情况下的帕累托最优值。从中挑选出能耗和不适感加权和最低的 5 个延迟时间,用计数的方法进行统计得到图 10~12 中 3 个场景下(每个场景 8 个不同请求概率的情况)选择出来的最优延迟时间出现次数的结果。其中不同场景下得到的最优延迟时间统计次数的分布是不同的,通过将这 3 个场景的结果合并,最终选择出现次数最多的延迟时间作为通用的延迟时间。图 13 则为权重(能耗:不适感=0.5:0.5)下的统计结果。

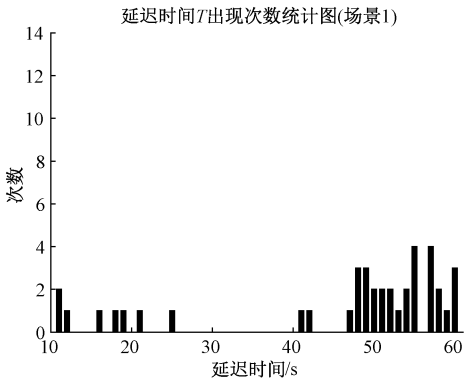


图 10 场景 1 次数统计结果

Fig. 10 Statistical results of scene 1

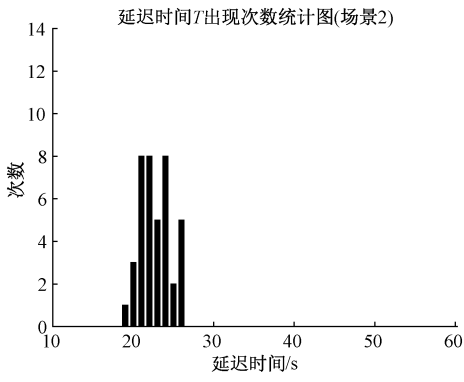


图 11 场景 2 次数统计结果

Fig. 11 Statistical results of scene 2

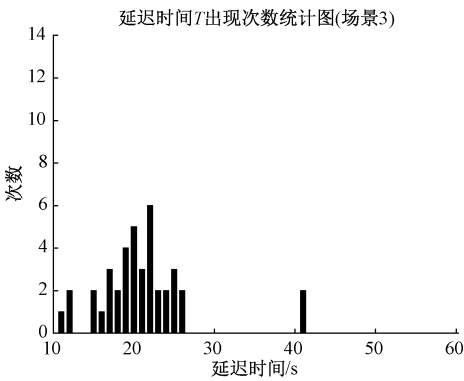


图 12 场景 3 次数统计结果

Fig. 12 Statistical results of scene 3

图 13 可以看出,次数最多可达到 14,对应的延迟时间为 22 s。所以,在权重比为 0.5:0.5 的情况下,得到的通用最优延迟时间为 22 s。

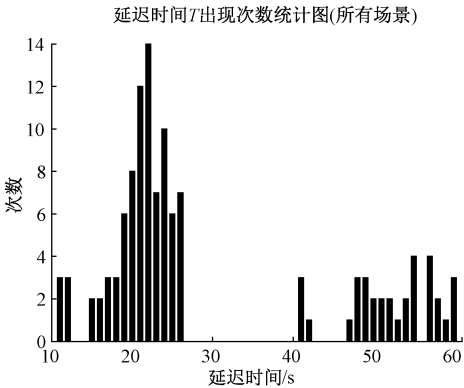


图 13 加权场景次数的统计结果(0.5:0.5)

Fig. 13 Statistical results of weighted scene times (0.5:0.5)

带有偏好的结果可以通过改变不同的权重比获得。图 14、15 分别为偏向降低能耗(能耗:不适感=0.6:0.4)和偏向降低不适感(能耗:不适感=0.4:0.6)的两种情况的结果图。

图 14 表示偏向降低能耗,从图中可以看出延迟时间 T 出现的次数分布相比图 13 往右偏移,延迟时间较长的部分出现次数相对较多。这是因为随着延迟时间的增加能耗必然降低。此时出现次数最多的延迟时间仍为 22 s,这意味着在图 14 的情况下,通用的最优延迟时间为 22 s。

图 15 表示偏向降低不适感,从图中可以发现出现次数偏向于延迟时间较少的部分。其中,延迟时间为 11 s 的次数最多,达到了 12 次,表明在图 15 的情况下,延迟时间 11 s 是通用的最优延迟时间。

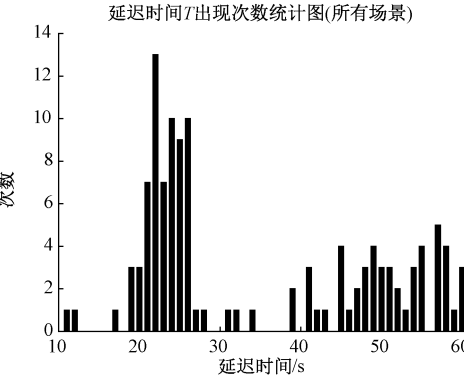


图 14 加权场景次数的统计结果(0.6:0.4)

Fig. 14 Statistical results of weighted scene times (0.6:0.4)

将延迟时间设为 22 s,即得出的通用最优延迟时间。在 3 个不同场景下将合并转发模型与普通模型(无延迟合并转发)一个小时内产生的平均能耗进行对比,结果如图 16 所示。不难看出,在满足用户体验的前提下,能耗分别降低了 25%、23%、12%,平均降低了 20%。

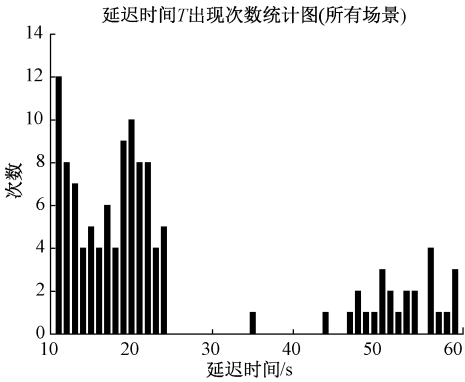


图 15 加权场景次数的统计结果(0.4 : 0.6)

Fig. 15 Statistical results of weighted scene times (0.4 : 0.6)

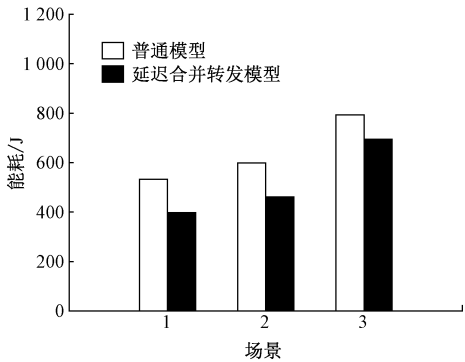


图 16 与普通模型不同场景下的能耗对比

Fig. 16 Energy consumption comparison with classical model in different scenarios

在合并转发模型中,将通用最优延迟时间 22 s 与其他延迟时间所产生的能耗和不适感进行对比,分别进行 1 h 的模拟,得出 3 个场景的平均能耗和平均不适感。数据如表 3 所示,第 3 列和第 5 列分别为当前延迟时间与通用最优延迟时间 22 s 相比能耗和不适感的变化。例如,22 s 相对于 20 s,虽然不适感增加 2.2%,但能耗降低 3.1%;相对于 24 s,能耗增加 1.0%,但不适感降低 1.4%。综合来看,22 s 的延迟时间最优。

表 3 通用最优延迟时间与其他延迟时间对比

Table 3 General optimal delay time compared with other delay times

时间/s	能耗/J	能耗变化/%	不适感	不适感变化/%
14	588.3	-12.0	440.9	+7.4
16	567.5	-9.6	450.1	+5.2
18	546.3	-6.1	456.4	+3.8
20	529.3	-3.1	463.6	+2.2
22	512.9	0	473.7	0
24	507.8	+1.0	480.2	-1.4
26	495.7	+3.5	492.5	-3.8
28	494.8	+3.7	503.0	-5.8
30	489.7	+4.7	508.1	-6.8

4 结 论

本文使用概率时间自动机对安卓智能手机 WiFi 模块、用户、阻塞控制器等进行了建模,同时使用统计模型检验的蒙特卡洛模拟,模拟合并转发网络请求的延迟对能耗以及用户满意度的影响。由于采用阻塞用户请求降低能耗的方案,即时请求并不受其影响,所以仅选取 3 种即时请求频率较低的场景。通过找出不同场景的帕累托最优值,最终计算出 3 个场景中通用最优延迟时间,并给出不同偏好的结果,实验结果可以给开发人员提供参考。在接下来的工作中,将会重点研究手机其他模块能耗和用户体验的相关优化问题。

参考文献

[1] 刘大同, 宋宇晨, 武巍, 等. 锂离子电池组健康状态估计综述[J]. 仪器仪表学报, 2020,41(11):1-18.
LIU D T, SONG Y CH, WU W, et al. Review of state of health estimation for lithium-ion battery pack [J]. Chinese Journal of Scientific Instrument, 2020, 41(11): 1-18.

[2] 丁阳征, 贾建芳. 改进 PSO 优化 ELM 预测锂离子电池剩余寿命[J]. 电子测量与仪器学报, 2019,33(2): 72-79.
DING Y ZH, JIA J F. Improved PSO optimized extreme learning machine predicts remaining useful life of lithium-ion battery [J]. Journal of Electronic Measurement and Instrumentation, 2019,33(2):72-79.

[3] LI D, HAO S, GUI J P, et al. An empirical study of the energy consumption of android applications [C]. 2014 IEEE International Conference on Software Maintenance and Evolution, 2014:121-130.

[4] GAO S, PENG Z, XIAO B, et al. SCoP: Smartphone energy saving by merging push services in Fog computing[C]. 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), 2017:1-10.

[5] BOLLA R, KHAN R, PARRA X, et al. Improving smartphones battery life by reducing energy waste of background applications [C]. Eighth International Conference on Next Generation Mobile Apps. IEEE, 2014:123-130.

[6] 程茹洁, 陆建, 蒋厚明, 等. 智能移动终端的能耗模型研究[J]. 计算机技术与发展, 2017,27(12):128-132,138.
CHENG R J, LU J, JIANG H M, et al. Research on energy consumption model on smartphones[J]. Computer Technology and Development, 2017, 27 (12): 128-132,138.

[7] LI D, HALFOND W G J. An investigation into energy-

- saving programming practices for Android smartphone app development [C]. International Workshop on Green and Sustainable Software, 2014:46-53.
- [8] LI D, HALFOND W G J. Optimizing energy of HTTP requests in Android applications [C]. International Workshop on Software Development Lifecycle for Mobile, 2015:25-28.
- [9] LI D, LYU Y J, GUI J P, et al. Automated energy optimization of http requests for mobile applications [C]. 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), 2016: 249-260.
- [10] HUANG J X, QIAN F, GERBER A, et al. A close examination of performance and power characteristics of 4G LTE networks [C]. International Conference on Mobile Systems, Applications, and Services, 2012: 225-238.
- [11] HUANG J, QIAN F, MAO Z M, et al. Screen-off traffic characterization and optimization in 3G/4G networks [C]. Internet Measurement Conference, 2012: 357-364.
- [12] 蔡华谦, 张颖, 黄罡, 等. 一种优化安卓应用 3G/4G 网络请求能耗的方法 [J]. 软件学报, 2017, 28(12): 3367-3384.
- CAI H Q, ZHANG Y, HUANG G, et al. Approach to scheduling network requests in Android apps [J]. Journal of Software, 2017, 28(12): 3367-3384.
- [13] 朱正伟, 刘晨, 黄晓竹, 等. 基于用户行为的智能手机能耗优化方法 [J]. 计算机工程, 2018, 44(5): 286-290.
- ZHU ZH W, LIU CH, HUANG X ZH, et al. Smartphone energy consumption optimization method based on user behavior [J]. Computer Engineering, 2018, 44(5): 286-290.
- [14] DAVID A, DU D, LARSEN K G, et al. Optimizing control strategy using statistical model checking [C]. NASA Formal Methods Symposium, 2013:352-367.
- [15] NORMAN G, PARKER D, SPROSTON J. Model checking for probabilistic timed automata [J]. Formal Methods in System Design, 2013, 43(2):164-190.
- [16] NAKAJIMA S, UEDA Y. Power consumption analysis of smartphone applications using UPPAAL [C]. IEEE International Conference on Cyber-Physical Systems, Networks, and Applications, 2013: 33-38.
- [17] 张贵玲, 朱正伟, 蒋威, 等. 基于 UPPAAL 的手机功耗流量建模方法研究 [J]. 电子测量与仪器学报, 2020, 34(12): 144-150.
- ZHANG G L, ZHU ZH W, JIANG W, et al. Power consumption and traffic modeling of smartphone based on UPPAAL [J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(12): 144-150.
- [18] MANWEILER J, CHOUDHURY R R. Avoiding the rush hours; WiFi energy management via traffic isolation [J]. IEEE Transactions on Mobile Computing, 2012, 11(5): 739-752.
- [19] 祁方民, 鱼滨, 牟力科, 等. 基于 CCS 的信号量形式化建模与验证 [J]. 计算机应用与软件, 2011, 28(8): 230-233.
- QI F M, YU B, MOU L K, et al. Formal modeling and validation of semaphore based on CSS [J]. Computer Application and Software, 2011, 28(8): 230-233.
- [20] BULYCHEV P, DAVID A, LARSEN K G, et al. UPPAAL-SMC: Statistical model checking for priced timed automata [J]. Electronic Proceedings in Theoretical Computer Science, 2012:1-16.

作者简介



薛普俊, 2019 年于山西农业大学获得学士学位, 现为常州大学硕士研究生, 主要研究方向为智能手机能耗分析和形式化方法。

E-mail: junm21@outlook.com

Xue Pujun received his B. Sc. degree from Shanxi Agricultural University in 2019. Now he is a M. Sc. candidate at Changzhou University. He main research interests include smartphone energy consumption analysis and formal methods.



朱正伟, 1984 年于东南大学获得学士学位, 1997 年于华东理工大学获得硕士学位, 2006 年于南京理工大学获得博士学位, 现为常州大学教授, 主要研究方向为无线传感器网络、智能检测技术及应用。

E-mail: zhuzw@cczu.edu.cn

Zhu Zhengwei received his B. Sc. degree from Southeast University in 1984, M. Sc. degree from East China University of Science and Technology in 1997, and Ph. D. degree from Nanjing University of Science and Technology in 2006, respectively. Now he is a professor at Changzhou University. His main research interests include wireless sensor network, intelligent detection technology and application.

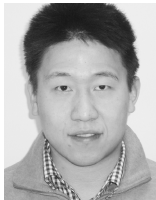


诸燕平, 2001 年于南京航空航天大学获得学士学位, 2004 年于南京航空航天大学获得硕士学位, 2010 年于南京航空航天大学获得博士学位。现为常州大学副教授, 主要研究方向为数值优化、信号处理、无线传感器网络。

E-mail: zhuyanping@cczu.edu.cn

Zhu Yanping received her B. Sc. degree from Nanjing University of Aeronautics and Astronautics in 2001, M. Sc. degree from Nanjing University of Aeronautics and Astronautics in 2004, and Ph. D. degree from Nanjing University of Aeronautics

and Astronautics in 2010, respectively. Now she is an associate professor at Changzhou University. Her main research interests include mathematical optimization, signal processing, and wireless sensor networks.



朱晨阳(通信作者),2012 年于华中科技大学获得学士学位,2014 年于美国宾夕法尼亚大学获得硕士学位,2020 年于英国南安普顿大学获得博士学位。现为常州大学讲师,主要研究方向为机器学习的集成方法、形式化方法、集成形式化方法、数据可视

化和时间建模。

E-mail: zcy@cczu.edu.cn

Zhu Chenyang (Corresponding author) received he B. Sc. degree from Huazhong University of Science and Technology in 2012, M. Sc. degree from the University of Pennsylvania in 2014, and Ph. D. degree from University of Southampton in 2020, respectively. Now he is a lecturer at Changzhou University. His main research interests include ensemble methods of machine learning, formal methods, integrated formal methods, data visualization, time modeling.