

基于队列长度的自适应占空比 MAC 协议

张闪闪,甄雪

(曲阜师范大学 信息科学与工程学院,山东 日照 276826)

摘要:由于无线传感器网络中设备电池能量有限,实现能量高效性是其主要考虑的问题。众所周知优秀的数据传输协议对降低能耗,延长网络生命周期有重要意义。而在基于竞争的 MAC 协议中,空闲监听是主要能量消耗来源,所以通常设计较低占空比的 MAC 协议来减少空闲监听时间。但是如果占空比过小,使得网络中所有数据传输集中在较短时间内,会导致节点无法完成所需要的通信。所以选择合适的节点占空比,对于资源受限的网络来说是一种很重要的节能方法。文中提出了一种新型的异步 MAC 协议 QL-MAC,节点利用 CSMA 争用信道,发送一系列短的信标包唤醒目标节点,采用虚拟载波监听防止串听。不同于其他改变接收节点的方法,该协议从发送节点来考虑,可以根据缓冲区队列长度判断该网络负载,从而自适应调整节点的占空比,使节点及时发送数据,减少时延,并告知接收节点做出相应改变,进一步节省能量。在 OMNET++上实现了 QL-MAC 协议,仿真结果表明 QL-MAC 协议展现了良好的时延和能量有效性。

关键词:占空比;CSMA;MAC;无线传感器网络

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2020)01-0070-06

doi:10.3969/j.issn.1673-629X.2020.01.013

An Adaptive Duty-cycle MAC Protocol Based on Queue Length

ZHANG Shan-shan, ZHEN Xue

(School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China)

Abstract: Due to the limited battery energy in wireless sensor networks, the realization of high energy efficiency is the main problem to be considered. It is well known that excellent data transmission protocols are important for reducing energy consumption and prolonging the life cycle of networks. In MAC protocols based on competition, idle listening is the main source of energy consumption, so MAC protocols with lower duty-cycle are usually designed to reduce idle listening time. However, if the duty-cycle is too small, all data transmissions in the network will be concentrated in a relatively short time, which will result in the failure of the nodes to complete the required communication. Therefore, selecting the appropriate duty-cycle is an important energy saving method for resource-constrained networks. We put forward a new kind of asynchronous MAC protocol named QL-MAC. The protocol uses the CSMA mechanism to contend channel and sends some short beacon packets to wake up the target node, and uses virtual CSMA to reduce the overhearing. Unlike other receiver-initiated protocols, this protocol can judge the load of the network according to the length of the buffer queue focused on the sender, so as to adaptively adjust the duty-cycle of the node. Then receiver decides whether to make a corresponding change. We have implemented QL-MAC in the OMNeT++, and the simulation shows that QL-MAC can reduce the end to end delay and energy consumption significantly.

Key words: duty-cycle; CSMA; MAC; WSNs

0 引言

无线传感器网络(wireless sensor networks, WSNs)是大量小型或微型低功耗传感器节点协作地感知和采集监测区域的各种信息并处理后,通过自组织多跳的方式形成短距离的无线通信网络。目前,WSNs已经广泛应用于智能建筑^[1]、抢险救灾^[2]、健康监测^[3]、目标跟踪^[4]等重要领域^[5]。WSNs在用来提供更好服务

的同时又面临很多挑战。由于WSNs中的节点大都采用电池供电,能量非常有限,并且节点一旦部署很难回收,所以对WSNs的能量效率研究一直是热点和难点问题。像所有共享媒体的网络一样,MAC是实现网络成功操作的重要技术。但是现有的无线媒体访问控制协议,例如蓝牙和802.11MAC协议都不能被直接应用到WSNs中,因为它们都没有将节能作为主要目标。

收稿日期:2019-02-23

修回日期:2019-06-24

网络出版时间:2019-09-25

基金项目:国家自然科学基金(61771289)

作者简介:张闪闪(1994-),女,硕士研究生,研究方向为无线网络与通信。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190925.1523.046.html>

已有许多研究表明,空闲监听消耗接收所需能量的 50% ~ 100%。例如,Stemm 和 Katz 测量到空闲:接收:发送比率为 1 : 1.05 : 1.4^[6],而 Digita 2 Mbps 无线局域网模块 (IEEE 802. 11/2 Mbps) 规范中空闲:接收:发送比率为 1 : 2 : 2.5^[7]。因此,为了延长网络的生存周期,有很多协议采用占空比机制来减少空闲监听造成的能量消耗。在占空比机制中,每个节点周期性地活跃期和睡眠期之间切换。在活跃期,节点监听信道并且传输数据,在睡眠期,节点切换到低功率状态(例如关闭收发器)进入睡眠模式。过低的占空比虽然节省能量但会造成密集传输,所以设计一个好的 MAC 协议,选择合适的占空比数值来实现高吞吐量低时延是 WSNs 的设计难点。

采用占空比机制的 MAC 协议一般分为两类:同步竞争和异步竞争。它们都是通过减少空闲监听,增加睡眠时间来节约能量的。有许多同步占空比协议被提出,例如 S-MAC^[8]、T-MAC^[9]、CL-MAC^[10]、TAS-MAC^[11]。但是这类同步协议要求每个节点可能会维持一个或多个同步时间表,并且对节点之间的时钟同步要求严格,这会产生很大的控制开销以及能量消耗,而且当网络负载发生变化时,高负载冲突较多,低负载空闲侦听较多,所以同步协议并不能达到很好的效果。依靠前导符采样的异步协议,例如 B-MAC^[12]、Wise-MAC^[13]、X-MAC^[14]、L-MAC^[15],允许节点维持自己独立的工作周期,收发双方不需要同步,采用低功率监听(LPL)前导序列技术唤醒接收节点,消除了同步协议的限制,在吞吐量延迟以及能量高效性方面超过了许多同步协议。但是,此类协议也有自身的缺点,例如长的前导符造成的非目标节点的串音问题以及异步之间互相等待引起的较高端到端时延等。

文中提出了一种基于缓冲区队列长度的异步 MAC 协议,不同于其他考虑接收节点操作的算法,而是从发送节点出发,利用缓冲区队列长度判断网络负载改变占空比,当网络负载高时节点可以延长发送时间来降低节点的端到端时延,尽快降低网络负载,网络负载低时,在保证数据发送量的条件下增加节点的睡眠时间,节省能量。

1 QL-MAC 设计

由于固定的占空比在网络负载变化时不能表现更好的性能,因此希望设计一个能根据网络负载大小改变占空比的异步 MAC 协议。首先将每个节点的周期分为非活跃阶段和活跃阶段,非活跃阶段即睡眠期,活跃阶段包括监听和传输,节点在这两个周期之间进行转换。设每个节点开始的睡眠-唤醒周期长度为 T 。

活跃阶段:此阶段包括检测事件、竞争信道、信标

包和数据包的传输和确认。协议开始时设定的活跃时间为 T_{active} ,其中包括检测事件 T_d ,竞争时间为 T_{contend} ,每个信标包的传输时间为 t_{beacon} ,接收一个确认包的时间为 t_{ack} ,传输数据时间为 T_{data} ,包括对分组的确认时间。

睡眠阶段:此阶段节点转入睡眠状态来节约能量,设睡眠时间开始时为 T_{sleep} 。在睡眠状态下,节点将采集的数据放入自己的缓冲队列中。

从以下几个阶段来详细介绍设计的 MAC 协议在这两个阶段的通信过程:

1.1 异步唤醒竞争信道

众所周知,两个节点通信时双方必须都处于活跃阶段,所以源节点发送数据时要等待目的节点唤醒才可以。异步通信中普通的前导符采样采用以下原理:节点周期性地唤醒一小段时间来对信道进行采样。如果信道是空闲的,那么节点立刻回到睡眠状态。如果检测到信道忙,则节点保持监听状态直到所有的数据都被接收或者是发生超时。这个方法要求在第一个数据帧前加上一个前导符,这个前导符的长度至少是一个信道采样间隔的长度,以免接收节点睡眠过程错过前导序列。但是此种方法非目标节点在长前导符发完之前不能发现该包并不是发给自己的,这造成了很大的能量浪费并且与节点密度成正比。为了避免这种问题,采用分组的前导符采样来进行异步唤醒。使用文献[16]中提出的 STEM-B 前导符变形,利用一系列的信标包来作为前导符,每个信标包包含了发送端和接收端的 MAC 地址以及此信标包的序列号。因此节点能够判断它是不是即将进行数据传输的目标节点,与传统的前导符采集协议相比,非目标节点可以尽早丢弃分组从而更早进入睡眠。当源节点唤醒时,在 T_d 时间内检测是否有事件发生,如果有事件发生则发送信标包等待目标节点唤醒。因为每个节点周期性的唤醒和睡眠,当网络中节点密度大,网络负载高时,可能有多个节点同时唤醒想要发送数据,所以会产生冲突,因此利用 CSMA 机制进行信道争用。设信道争用时间为 T_{contend} ,在此时间内没有收到任何信息则获得信道发信标等待目的节点唤醒, b_n 就是信标包,如果接收到信标,则判断信标是否发给自己,如果是则回复 ACK 等待接收数据,如果不是则转入睡眠,等待下一周期重复上述过程。

1.2 数据传输

当节点获得信道与目的节点建立通信后开始传输数据,如果将一个长消息装入数据包后,一旦损坏就需要重传整个长消息,而无线信道中传输差错与消息的长度成正比例关系,因此长消息有着比短消息更低的成功传输概率,这会有很大的能量时间以及带宽消耗。

因此该协议不像 SW-MAC^[17] 和 DS-MAC^[18] 那样发送大分组转入睡眠,而是将长消息分段,在唤醒时间连续发送数据包,并等待确认,直到唤醒时间到期,如图 1 所示。

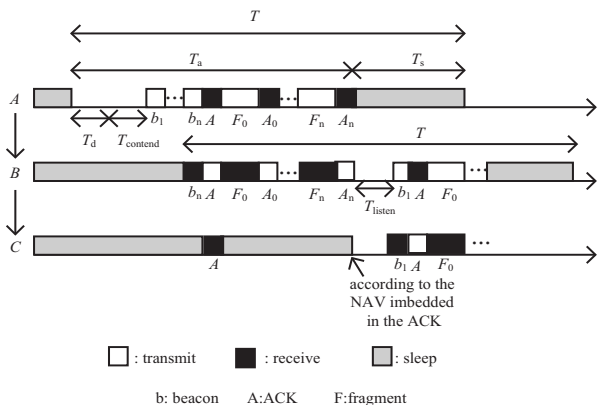


图 1 QL-MAC 协议数据传输

所以节点唤醒之后有以下几种情况发生:

(1) 节点有数据发送且获得信道,发送一系列含有目标地址和序列号的信标包,然后在信标包之间预留接收目的节点确认的时间 t_{ack} 。然后开始传输数据,所以此时数据传输总时间为:

$$T_{data}^1 = T_a - T_d - T_{contend} - T_b = T_a - T_d - T_{contend} - p(t_{beacon} + t_{ack}) \quad (1)$$

此处 p 为传输到第 p 个信标包时目的节点唤醒。若在 $T_{contend}$ 期间收到了信标,则判断此信标是否是发给自己的,如果是则回复 ACK 准备接收数据,如果不是则丢弃。

(2) 节点有数据发送但没有获得信道,为节省能量节点转入睡眠,所以活跃时间为:

$$T_a^1 = T_d + T_{contend} \quad (2)$$

(3) 节点唤醒但是在 T_d 时间没有数据要发送,此时设置监听时间为 $T_{listen} = t_{beacon} + t_{ack}$,在此时间内如果有信标包接收则接收信标包,回复 ACK 并转发数据,此时数据传输时间为:

$$T_{data}^2 = T_a - T_d - T_{listen} = T_a - T_d - t_{beacon} - t_{ack} \quad (3)$$

如果没有接收到信标包则睡眠,此时

$$T_a^2 = T_d + T_{listen} = T_d + t_{beacon} + t_{ack} \quad (4)$$

1.3 改变占空比

文中希望利用提出的协议根据网络负载的大小来自适应调整节点的睡眠时间。因此,就需要提出一种方法来度量网络负载情况。检测网络负载一般有以下几种方法:一是通过串听 RTS/CTS 等控制分组,二是通过估算传输分组之间的端到端时延,三是通过缓冲区队列长度。这是因为在采用 CSMA 协议的数据传输中,对于竞争失败的节点,数据会积压在此节点本身的缓冲区队列中等待下一次争用信道。所以当网络负

载处于较高情境时,缓冲区中积压的数据分组就会增加,可以传输的节点很难在较短的活跃期内传输完所有数据,并且其他没有获得信道从而处于退避阶段的节点也会有很多数据分组在缓冲区中积压。所以可以通过缓冲区队列长度反应网络负载。因此传感器节点通过本地缓冲区队列内待发数据分组的积压情况就可以真实迅速反映网络当前的负载情况,于是就利用这个特性来自适应调整节点在该周期的睡眠时间。

对于发送节点:

因为希望设计的协议能根据网络负载调整占空比,文中的重点在于找到一个缓冲区队列的阈值来判断网络负载大小。所以计算在整个睡眠-唤醒周期中能传输的分组数量为:

$$n = \frac{T_{data}}{t_{packet} + t_{ack}} \quad (5)$$

其中, t_{packet} 是一个分组传输的时间。

在最好的情况下信道空闲,节点仅传输一个信标就接收到来自接收端的 ACK,此时时间为 $t_{packet} + t_{ack}$,那么可求出最大传输分组个数 n_{max} 为:

$$n_{max} = \frac{T_a - T_d - T_{contend} - t_{beacon} - t_{ack}}{t_{packet} + t_{ack}} \quad (6)$$

实际情况中,由于可能产生信道争用,并且接收节点的唤醒时间可能不会那么及时,所以一次能发送的分组个数远小于 n_{max} 。

假设每个节点在 T_d 时间计算其缓冲区当前队列长度即缓冲区中分组数量,记为 L 。虽然一次能发送分组的个数远小于 n_{max} ,但是单纯的 n_{max} 作为阈值来和队列长度比较,当出现 L 和 n_{max} 很接近的时候,占空比调整会过于频繁,所以将 $2n_{max}$ 和 $n_{max}/2$ 作为阈值。 $L > 2n_{max}$,说明此时网络负载大, $L < n_{max}/2$,说明此时网络负载小。根据网络负载调整占空比,为了使协议实现起来更加简单,通过将占空比加倍和减半来实现。

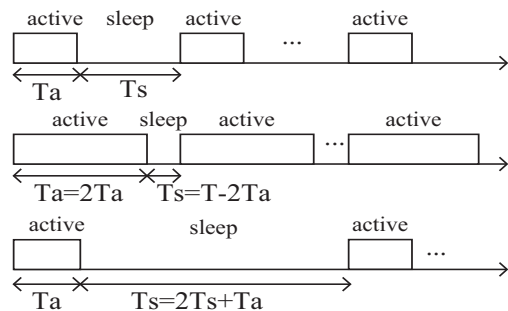


图 2 改变节点占空比

如图 2 所示,在加倍占空比操作中,保持周期 T 不变,通过延长活跃时间来实现。在减半占空比操作中,保持活跃时间不变通过延长睡眠时间增大周期 T 来实现。这样做的好处是:当网络负载大时,唤醒时间加倍,这样可以尽快发送完分组且降低节点的端到端时

延;当网络负载小时,唤醒时间不变,睡眠时间延长,这可以在保证节点每个周期发送数据量的情况下增加节点的睡眠时间,节省能量。

对于接收节点:
当发送节点的占空比改变时,接收节点也要做出相应改变,这种情况只需要在占空比加倍时候改变,否则就会出现当占空比加倍时发送端没有发送完但接收端的活跃时间已经到期的情况。为了避免此类情况发生,在信标包和确认包中加入一个 1 bit 的 Flag 字段,正常占空比用 0 来表示,如果占空比加倍则改为 1,接收节点根据 Flag 位对自己做出调整。而发送节点占空比减半时接收节点不需要改变,因为接收节点也会接收到其他节点的数据包,不能仅根据其中一个发送节点就延长自己的睡眠时间。

1.4 虚拟载波监听

在数据传输的过程中,当另一个节点在发送的时候节点唤醒,那么可能发生串听。为了解决这个问题,使用虚拟载波监听(virtual carrier sense, VCS)。VCS 采用网络分配向量 NAV 获得信道控制信息。在通信范围内的任何节点都能看到此值,它采用单位时间来维护并基于本地时钟递减。当 NAV 的值不为 0 的时候表明信道被占用。在睡眠阶段 NAV 计时器保持工作直到减为 0。当 NAV 的值为 0 的时候节点处在活跃状态,那么节点唤醒竞争信道,如果节点处于睡眠状态则继续睡眠等待下一个周期。

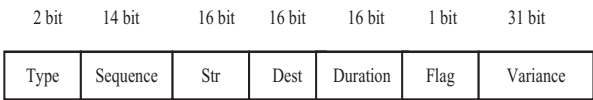


图 3 信标包和确认包格式

如图 3 所示,可以在信标包和确认包中的 duration 字段添加两个节点的通信时间来配置 NAV 值。这可以在发送端和接收端占用信道的时候,告知其他节点信道忙,无须争用信道,对于其他节点来说,可以减少不必要的争用,并且减少串听。占空比变化和 NAV 配置如下所示:

- (1)若 $L > 2n_{\max}$,则表明缓冲区队列长度远大于此周期内能传输的最大分组数量,说明节点此时的网络负载已经偏高,此时需要尽快发送分组,睡眠时间不变,通过延长唤醒时间将占空比加倍,此时 $T_a = 2T_a$, $NAV = 2T_a - T_d - T_{\text{contend}}$ 。
- (2)若 $L < n_{\max}/2$,则说明此时的网络负载偏小,需要更多的休眠时间,唤醒时间不变,通过增加睡眠时间将占空比减半,此时 $T_s = 2T_s + T_a$, $NAV = L * (t_{\text{packet}} + t_{\text{ack}})$ 。
- (3)其余情况执行原占空比, $NAV = T_a - T_d - T_{\text{contend}}$ 。

2 节点操作

具体节点操作的伪代码如下所示。State (NAV, channel state, node state) 分别代表 NAV 值,信道状态和节点状态。在 NAV 值中,0 代表 NAV = 0,1 代表 NAV ≠ 0。信道状态分为忙和空闲。节点状态包括活跃和睡眠。 φ 代表任意状态。

算法:节点数据传输。

```
1 if node A has data to transmit then
2 node A contends the channel;
3 node A detects the queue length L;
4 switch the state(NAV,channel state,node state) do
5 case(0, idle, active);
6 if  $L > 2n_{\max}$  then
7 double the dutycycle
8 set  $NAV = 2T_a - T_d - T_{\text{contend}}$ 
9 else if  $L < n_{\max}/2$  then
10 half the dutycycle
11 set  $NAV = L * (t_{\text{packet}} + t_{\text{ack}})$ 
12 end if
13 else
14 keep original dutycycle
15 set  $NAV = T_a - T_d - T_{\text{contend}}$ 
16 end if
17 node A send beacon packet to node B
18 case(0,  $\varphi$ , sleep)
19 node continues sleep and waits for next cycle
20 case(0,  $\varphi$ ,  $\varphi$ )
21 node countdowns NAV
22 else if node A received the beacon packet then
23 check flag
24 if flag = 1 then
25 set  $T_a = 2T_a$ 
26  $T_s = T - 2T_a$ 
27 node A replies ACK to sender
28 else
29 node A replies ACK to sender
30 end if
31 end if
32 else
33 node keeps active until the end of  $T_{\text{listen}}$ 
34 node turns to sleep and wakes up in the next cycle
35 end if
```

当节点唤醒之后,先监听信道,若检测到信道忙则进入睡眠状态倒计时 NAV,当 NAV = 0 时且节点处于活跃状态则再次接入信道,否则继续睡眠等待下一周期。若检测到信道空闲且 NAV = 0,那么根据当前缓冲区长度判断是否需要更改占空比,若负载过大,则将占空比加倍,过小则将占空比减半,并且设置 NAV 持续时间占用信道发送数据,若负载正常则占空比不变;

如果一个节点在最大监听时间内没有事件发生,即不发送数据也不接收数据,直接转入睡眠状态节省能量。

图 4 给出了节点的有限状态机转换图。

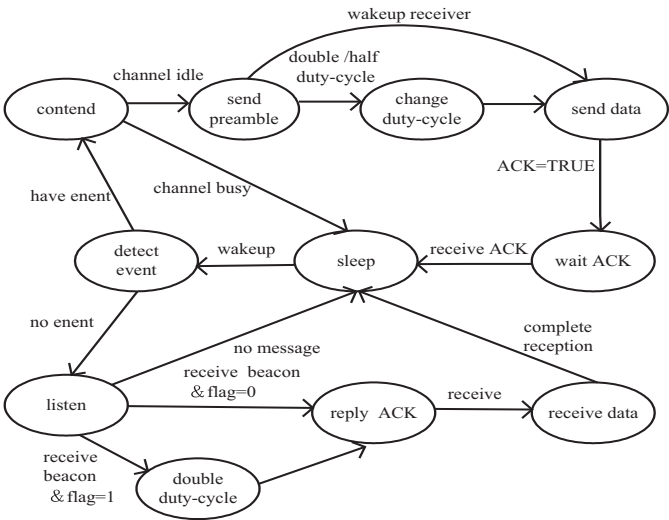


图 4 有限状态机

3 仿真与分析

为了验证 QL-MAC 的性能,在 OMNET++ (Version:5.2.1) 仿真器中对 QL-MAC 协议和 X-MAC,S-MAC 协议在端到端时延和能量有效性方面作性能比较。S-MAC 是无线传感器网络的典型同步 MAC 协议,而 X-MAC 解决了异步 MAC 协议中前导符过长的缺点,二者在无线传感器网络中都表现出了很好的性能。使用链式拓扑和星型拓扑。在仿真中,每个节点之间的距离是 200 m,通信范围是 250 m,确保节点仅能和一跳邻居通信。采用单信道进行通信,节点工作在半双工模式下,不考虑干扰,利用源节点产生数据分组的时间间隔来表示网络中的负载情况。设置此间隔从 1 s 变化到 10 s,当间隔为 1 s 的时候表示此时网络负载最高,10 s 时表示网络负载最低。在实验过程中,每个仿真运行 60 s,节点初始占空比为 10%,每个周期为 1 s。

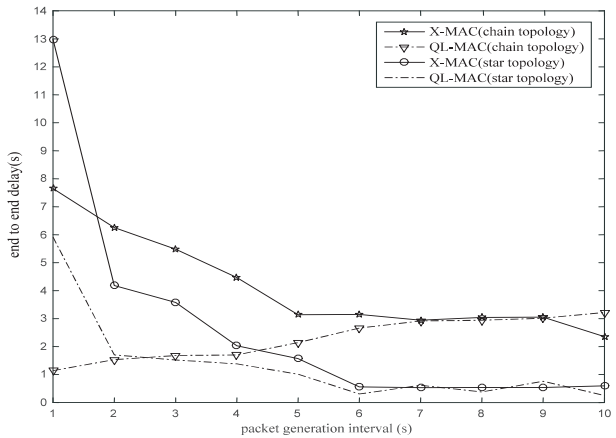


图 5 端到端时延对比

如图 5 所示,不论链式拓扑还是星型拓扑,随着包

间隔时间的增加,X-MAC 的网络延迟逐渐降低并趋于平稳,当包间隔时间为 10 s 时,网络延迟最小。在链式拓扑下,QL-MAC 的时延逐渐增加然后趋于平稳,这是因为 QL-MAC 在高负载时,增加活跃时间,减少睡眠可以获得比低负载下更好的时延,即使后来时延增加也是低于 X-MAC。在星型拓扑下,高负载时节点竞争激烈,X-MAC 的 preamble 包竞争变强,导致比链式更高的时延,但是当负载低时,由于跳数减少时延下降明显,此后趋于平稳。QL-MAC 在星型的高负载下也是竞争激烈,相比链式时延增加,但是仍然低于 X-MAC,随负载变低,QL-MAC 的优势逐渐降低,但是仍然和 X-MAC 不相上下。

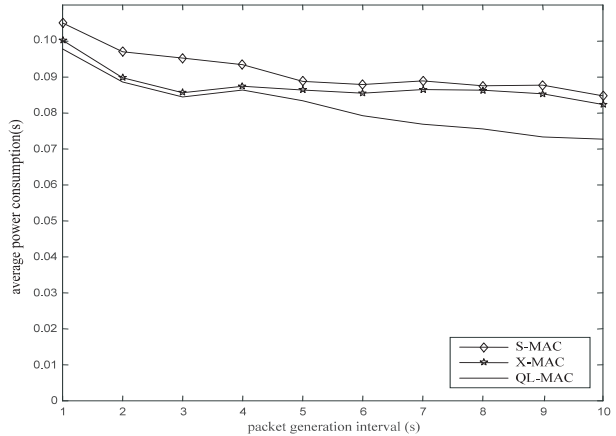


图 6 节点平均能量消耗对比

从图 6 可以看出,随着网络负载变低,系统消耗的平均功率在降低。当高负载时,X-MAC 数据传输效率低,节点需要唤醒睡眠多次才可传完,而 QL-MAC 协议会根据高负载延长活跃时间,减少多次睡眠唤醒之间消耗的能量,从而节约小部分能量,在低负载下 QL-MAC 优势更加明显。总之,QL-MAC 协议在能

量有效性方面比 S-MAC 和 X-MAC 要好。

4 结束语

提出了一种基于 CSMA 和缓冲区队列长度的异步 MAC 协议,主要目的是用来减少动态链路负载下的端到端时延。该协议使用 CSMA 争用信道,从发送节点出发根据缓冲区队列长度自适应改变节点的睡眠时间,当网络负载较大时,增大活跃时间,当网络负载较小时,延长睡眠时间,并通知接收节点做出相应改变。最后通过仿真表明,该协议获得了良好的时延和高能效。

参考文献:

- [1] CUI Kaiwen, KUMAR A, XAVIER N, et al. An intelligent home appliance control-based on WSN for smart buildings [C]//IEEE international conference on sustainable energy technologies. Hanoi; IEEE, 2017; 282-287.
- [2] LARA-CUEVA R A, GORDILLO R, VALENCIA L, et al. Determining the Main CSMA parameters for adequate performance of WSN for real-time volcano monitoring system applications[J]. IEEE Sensors Journal, 2017, 17(5): 1493-1502.
- [3] DEY N, ASHOUR A S, SHI F, et al. Developing residential wireless sensor networks for ECG healthcare monitoring[J]. IEEE Transactions on Consumer Electronics, 2017, 63(4): 442-449.
- [4] DEMIGHA O, HIDOUICI W, AHMED T. On energy efficiency in collaborative target tracking in wireless sensor network; a review[J]. IEEE Communications Surveys & Tutorials, 2013, 15(3): 1210-1222.
- [5] 崔司千. STDMA 多跳无线网络分布式时隙共享策略研究[D]. 哈尔滨: 哈尔滨工业大学, 2016.
- [6] HAN J A, KIM Y H, PARK S J, et al. Energy efficient MAC protocol with reducing redundant packet mechanism in wireless sensor networks[M]//Systems modeling and simulation; theory and applications. Berlin; Springer, 2006.
- [7] LEE B, PARK K, ELMASRI R. Balancing overhearing energy and latency in wireless sensor networks[C]//Proceedings of the 2008 IFIP conference on wireless sensor and actor networks (WSAN 08). Ottawa, Ontario, Canada; Springer, 2008; 25-36.
- [8] WEI Y, HEIDEMANN J, ESTRIN D. An energy-efficient MAC protocol for wireless sensor networks [C]//IEEE global telecommunications conference. [s. l.]; IEEE, 2002.
- [9] VAN DAM T. An adaptive energy efficient MAC protocol for wireless sensor networks [C]//Proceedings of the 2nd ACM conference on embedded networked sensor systems (Sensys 2003). [s. l.]; ACM, 2003.
- [10] HEFEIDA M S, CANLI T, KHOKHAR A. CL-MAC: a cross-layer MAC protocol for heterogeneous wireless sensor networks[J]. Ad Hoc Networks, 2013, 11(1): 213-225.
- [11] HUANG P, LIU C J, XIAO L. TAS-MAC: a traffic-adaptive synchronous MAC protocol for wireless sensor networks [C]//2013 IEEE international conference on sensing, communications and networking (SECON). New Orleans, LA; IEEE, 2013; 113-121.
- [12] POLASTRE J, HILL J, CULLER D. Versatile low power media access for wireless sensor networks[C]//2nd international conference on embedded networked sensor systems. Baltimore, MD, USA; ACM, 2004; 95-107.
- [13] ELHOIYDI A, DECOTIGNIE J D. WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks [C]//International symposium on algorithms and experiments for sensor systems, wireless networks and distributed robotics. Turku, Finland; Springer, 2004; 18-31.
- [14] BUETTNER M, YEE G V, ANDERSON E, et al. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks[C]//4th international conference on embedded networked sensor systems. Boulder, Colorado, USA; ACM, 2006; 307-320.
- [15] DINH T, KIM Y, GU T, et al. L-MAC: a wake-up time self-learning MAC protocol for wireless sensor networks [J]. Computer Networks, 2016, 105; 33-46.
- [16] SCHURGERS C, TSIATSIS V, GANERIWAL S, et al. Optimizing sensor networks in the energy-latency-density design space[J]. IEEE Transactions on Mobile Computing, 2002, 1(1): 70-80.
- [17] LIANG Lulu, LIU Xiaonan, WANG Yongtao, et al. SW-MAC: a low-latency MAC protocol with adaptive sleeping for wireless sensor networks[J]. Wireless Personal Communications, 2014, 77(2): 1191-1211.
- [18] WANG G, YU J, YU D, et al. DS-MAC[J]. Journal of Network and Computer Applications, 2015, 58(C): 155-164.