

文章编号:1674-2869(2017)05-0488-08

软件定义无线传感器网络中低控制负载的睡眠调度

赵 腾¹,王海晖^{1,2*},彭 云¹,赵茂阳¹,张水平^{1,2}

1. 武汉工程大学计算机科学与工程学院,湖北 武汉 430205;
2. 智能机器人湖北省重点实验室(武汉工程大学),湖北 武汉 430205

摘 要:为了克服软件定义无线传感器网络(SDWSN)中控制流量的限制,依据基于能量消耗的连通 k 邻域睡眠调度算法和软件定义网络(SDN)的特性,提出了一种低控制负载的睡眠调度方案.首先,利用SDN的网络模型重新设计无线传感器网络的模型;然后,依据新的网络模型,对传统的睡眠调度方案进行改进.在此基础上,设计出能够降低网络中控制流量的流表,从而降低SDWSN中的控制负载.仿真结果表明,本文所提出的控制流设计方案在控制流量和数据平面中更新节点状态的平均响应时间要优于SDWSN中的睡眠调度,该方案能够最小化整个网络中的控制流量,在延长网络生命周期的同时降低控制负载.

关键词:无线传感器网络;软件定义网络;睡眠调度

中图分类号:TP391 文献标识码:A doi:10.3969/j.issn.1674-2869.2017.05.015

Low Control Overhead-Based Sleep Scheduling for Software-Defined Wireless Sensor Networks

ZHAO Teng¹, WANG Haihui^{1,2,*}, PENG Yun¹, ZHAO Maoyang¹, ZHANG Shuiping^{1,2}

1. School of Computer Science and Technology, Wuhan Institute of Technology, Wuhan 430205, China
2. Hubei Key Laboratory of Intelligent Robot(Wuhan Institute of Technology), Wuhan 430205, China

Abstract: To overcome the limitation of the control flow in the software-defined wireless sensor network (SDWSN), a low control overhead sleep scheduling program was proposed based on the energy consumption of the connected k neighborhood sleep scheduling algorithm and the characteristics of software-defined network (SDN). Firstly, the model of the wireless sensor network was redesigned by using the SDN network model. Then, the traditional sleep scheduling scheme was improved according to the new network model. On the basis, the flow table can be designed to reduce the size of the control flow in the network and the control flow overhead can be reduced in the SDWSN. The simulation results show that the control flow size and the average response time of updating the node status in the data plane are better than those in the SDWSN. The proposed scheme can minimize the total control flow size in the whole network and reduce the control overhead while extending the network life cycle.

Keywords: wireless sensor network; software-defined networking; sleep scheduling

收稿日期:2017-05-08

基金项目:国家自然科学基金项目(61401107);武汉工程大学研究生教育创新基金项目(CX2016068)支持

作者简介:赵 腾,硕士研究生. E-mail: zhaoteng93@outlook.com

*通讯作者:王海晖,博士,教授. E-mail: wanghaihui69@sina.com

引文格式:赵腾,王海晖,彭云,等. 软件定义无线传感器网络中低控制负载的睡眠调度[J]. 武汉工程大学学报, 2017, 39(5):488-495.

ZHAO T, WANG H H, PENG Y, et al. Low control overhead-based sleep scheduling for software-defined wireless sensor networks[J]. Journal of Wuhan Institute of Technology, 2017, 39(5):488-495.

近年来,over-the-air 软件更新和 re-tasking 策略被用于增强不同厂商产品的兼容性和增强典型的无线传感器网络(wireless sensor network ,WSN)中传感器节点硬件重循环方面的灵活性. 软件定义网络(software defined network ,SDN)^[1-4]是一种新兴的解决方案,可以在 WSN 中动态地重新配置传感器节点的感测、路由及数据转发任务. SDN 的主要思想是在物理层面上解耦控制平面和数据平面,传感器节点不具有任何任务决策能力,而是由来自集中式控制器的指令控制. 因此,传感器节点易于控制并且重新指定任务或者重新编程非常简单,而已经部署的传感器节点的硬件不需要进行很大的改变.

OpenFlow^[3]是一种开放协议,它是控制平面和数据平面之间的标准接口,允许在流表中重新编程和更新. Sensor OpenFlow^[2]是将软件定义网络整合到无线传感网络中的早期尝试之一. 在此之后,针对 WSN 的 SDN 解决方案^[4],基于状态的 OpenFlow 解决方案由于其减少了数据平面和控制平面之间的业务交换而受到极大关注. Gante 等人^[5]提出了一种提供带内和带外控制流管理的方法. 此方法提出常规节点受益于集中式架构,它处理路由,移动性和服务质量管理. 但是,在流表中替换和添加流表项需要内存和流量负载,一种称为 FLIP^[6]的快速权重策略保留 SDN 更新,基于流表项的替换和添加之间的二元性降低了负载.

软件定义无线传感器网络(software-defined wireless sensor network ,SDWSN)的传感器激活和感测调度在文献^[7]中进行了分析,特别提出了具有低重新调度时间和低控制负载的高效算法,以设计用于具有不同感测目标的多任务的基于能效的传感器调度. 几个软件定义无线传感器网络原型^[8]已经用基于服务质量(QoS)且要求动态调整感测参数的传感器节点进行了实现. 例如, TinySDN^[9], 一个硬件独立的基于 TinyOS 的 SDN 框架在多控制器上进行了实现. 与基于 TinyOS 的应用中使用集合树协议(CTP)相比,该方法在低延迟和存储器负载中受益. 该架构主要包括启用 SDN 的传感器节点和启用 SDN 的控制器节点. 启用 SDN 的传感器节点像 SDN 交换机一样执行数据包传感任务和转发,而启用 SDN 的控制器节点主要保存拓扑信息并管理控制流.

传统的无线传感器网络通常由于传感器尺寸过小、难以接近以及能量有限而遭受寿命问题. 睡眠调度是延长网络寿命的有效方法之一,其允许

唤醒节点的子集,而其他节点进入休眠一定量的时间以节省能量. 在对无线传感器网络的睡眠调度研究中,基于连接 k 邻域(connected k neighborhood ,CKN)^[10]的算法有效地延长了网络寿命. Wang 等人在软件定义无线传感器网络中提出了基于能耗的 CKN(EC-CKN)算法^[12],这是一种众所周知的基于能耗的睡眠调度算法,并提出了一种称为 SDEC-CKN 的新方案^[11].

然而,控制平面和数据平面之间的控制业务是软件定义无线传感器网络中的主要关注点之一,SDWSN 没有任何单独的控制信道. 此外,在大多数情况下,流表中的流表项仅使用分组头部来进行规则匹配. 最近, Galluccio^[4]提出了一种根据节点当前状态来设计流表项的方法,但是对于 WSN 中能量平衡睡眠调度缺乏详细的讨论. 虽然在 SD EC-CKN^[11]中研究了基于能耗的睡眠调度算法,但每个节点需要在每个周期内向控制器发送信标消息,这将会导致高控制负载.

本文将 SDN 架构引入传统 WSN 架构之中,并通过设计合理的数据包和流表,实现基于 SDN 的睡眠调度算法,并依据 SDN 架构^[13-16]的特性,降低网络中控制负载,在控制流量和数据平面中更新节点状态的平均响应时间要优于 SDWSN 中的睡眠调度.

1 系统模型

1.1 网络模型

相较于 WSN,SDWSN 具有统一和随机部署无线传感器节点的能力. 如图 1 所示,控制平面由单个控制器组成以确定通过网络的路由和数据转发决定,而数据平面包含传感器节点和交换机. 交换机将控制器产生的数据包转发到网络. 每个传感器节点由用于发送和接收数据包的通信单元、感测单元和流表组成. 任何节点更改其状态(睡眠或唤醒)、转发或丢弃数据包都要根据流表中的流表项来进行.

1.2 SDWSN 中的主要组件

SDWSN 中最重要的三大组件是控制器、流表和交换机.

1) 控制器:SDWSN 中的控制器拥有由传感器节点生成的全局拓扑信息. 在初始阶段,控制器广播监视消息(类似于拓扑发现)以获取每个节点的能量状态、到节点的距离、邻居节点 ID 和延迟. 在获取这些信息作为输入之后,映射函数生成诸如网络互连、信号强度和链路可靠性的映射信息. 控

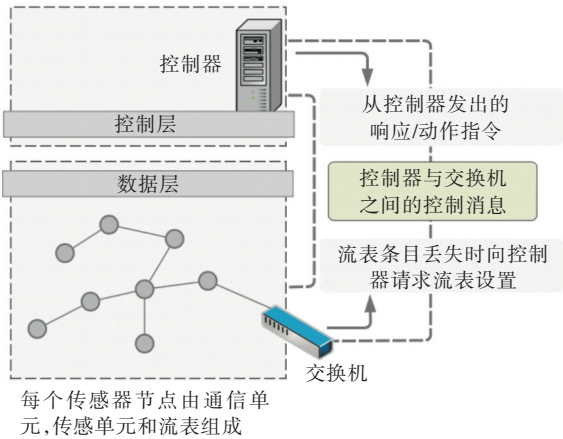


图 1 SDWSN 网络模型
Fig. 1 SDWSN network model

制器可以直接访问这些映射信息，并通过映射功能随时更新这些信息。

2)流表:SDN 中的流大致分为数据流和控制流.数据流中包含感测数据,而控制流主要是指流建立请求、流动作请求、拓扑发现消息、1 跳邻居节点的能量和状态(睡眠或唤醒状态)消息、路由信息等.典型的流表包含以下字段:匹配、操作和统计.图 2 示出了流表中的流表项的示例.此流表项转发源节点 ID = X 的数据包,统计字段生存时间(time to live, TTL) = 80 指定流表中此项的剩余时间,计数器表示此流动操作执行 16 次.匹配规则字段将流表头与其自己的流表项进行比对,并允许执行响应动作.操作字段通常为指定转发或丢弃数据包、将数据包上传更新到控制器、状态改变和修改数据包的字段等动作.统计字段用于指定流表中流表项的 TTL,并计算满足以下规则的数据包数.此外,流表项可以包含多于一个的匹配规则字段.控制器可以基于应用层需求更新、修改、添加和删除流表中的任何项。

匹配			操作	统计	
规则 1	规则 2	规则 N		生存时间	计数器
源节点 ID=X			转发	80	16

图 2 流表项示例
Fig. 2 An example of flow entry

3)SDWSN 交换机:数据平面中的交换机不再具有如在传统 WSN 交换机中的决策能力.这个交换机简单地将动作指令从控制器转发到数据平面中的传感器节点,此外当流表中发生项缺失的时候将通过该交换机来向控制器请求响应。

1.3 睡眠调度算法

在传统的 WSN 中的睡眠调度算法,在有效的关闭能量相对较低的节点同时保证网络的连通性.在 CKN 算法中,每个节点如果有 k 个处于唤醒状态的邻居节点,这个节点将会关闭自己;如果处于唤醒状态的邻居节点数量低于 k ,节点保持唤醒状态.这种分布式的 CKN 算法延长了节点的存活时间和整个网络的生命周期.但是,CKN 算法中的能量不能保证被均衡消耗.和 CKN 算法不同的是,EC-CKN 算法在 CKN 算法的基础上考虑节点剩余能量,平衡整个网络的能量消耗,同时保证网络依旧是 k 邻接. EC-CKN 的详细过程如下所示:

- 1)获取当前节点剩余能量信息 Er_{ank_u} ;
- 2)广播 Er_{ank_u} ,接收当前处于唤醒状态的邻居节点 N_u 的能量等级.定义 R_u 为这些能量等级;
- 3)广播 R_u ,从每个 $S_v \in N_u$ 中获取 N_u ;
- 4)如果 $S_v \in N_u$ 中的任何一个节点的 $|N_u| < k$ 或者 $|N_v| < k$,则节点保持唤醒状态.返回;
- 5)计算 $E_u = \{S_v | S_v \in N_u \text{ 而且 } Er_{ank_v} > Er_{ank_u}\}$;
- 6)当满足以下条件的时候节点进入睡眠状态,不满足则保持唤醒状态; E_u 中的任何两个节点直接自身或间接地通过具有大于 Er_{ank_u} 的 Er_{ank_v} 的 S_u 的 2 跳邻居中的节点连接;
- 7) N_u 中的任何节点与 E_u 至少有 k 个邻居;
- 8)返回.

在 SDWSN 中,选择 EC-CKN 作为基于 SDN 的基本算法,因为它的判断标准与节点的剩余能量相关,可以直接反映整个网络的能量消耗.在 SD-ECCKN 算法中,采用基于 SDN 的架构来代替传统的 WSN,在新架构下去除节点进行广播的过程,从而减少总传输时间并且降低能耗.从步骤 2 到步骤 3,每个节点广播两次以获得其 1 跳和 2 跳节点的状态,用于稍后判断其自身状态.在每个间隔中,每个节点的这两个广播过程花费大量的通信能量.而在 SDWSN 中,控制器根据节点发送的数据,决定网络中每个节点的状态,并根据 ECCKN 算法对网络拓扑进行更新,然后控制器将这些决定发送到所有状态需要改变的节点。

而在 SDWSN 中,由表 1 可以看出,根据这些差异,所提出的 SD-ECCKN 算法在每个周期内消除了广播过程.每个节点通过已经最初预先定义的特定路由向控制器发送信标数据.在网络的整个生命周期内,节点之间没有数据交换(广播).因此,网络寿命下降的总传输时间节省了能源,延长了整个网络的使用寿命。

表1 SD-ECCKN与EC-CKN的区别

Tab. 1 Differences between SD-ECCKN and EC-CKN

SD-ECCKN	EC-CKN
1.每个节点保持唤醒,以在初始网络拓扑中的路由之后向控制器发送信标数据	1. 每个节点广播信标数据两次以获得其当前1跳和2跳邻居的能量等级
2. 控制器根据节点发送的信标数据,决定网络中每个节点的睡眠状态,并根据ECCKN算法对网络拓扑进行更新。然后控制器将这些决定发送到所有受影响的节点	2. 每个节点基于通过广播信标数据获得的能量等级来确定其睡眠状态两次
3.每个节点执行决定:进入睡眠状态或保持唤醒以在每个间隔内发送数据	3.每个节点执行决定:进入睡眠状态或保持唤醒以在每个间隔内发送数据

2 SDWSN中的控制流

控制流分为以下类型:a)控制请求消息,<类型=0>,b)控制动作消息,c)更新消息<类型=1>,以及d)广播消息<类型=2>.

a. 控制请求消息<控制请求头,节点能量,邻居节点ID>:发送到控制器以获得响应指令,其中<节点能量>和<邻居节点ID>表示节点的剩余能量

和1跳邻居节点ID分别生成控制请求消息.

b. 控制动作消息<控制动作头,状态>:来自控制器的相应响应/动作指令,其中<状态>定义节点的睡眠或唤醒状态的决策.

图3(a)示出了控制动作和控制请求消息的报头格式.<动作/请求>指定控制消息的类型,即控制消息从控制器发送(=1)或者需要发送到控制器(=0).<分组长度>用于包括有效载荷的数据包分组的长度.<源节点ID,目标节点ID,下一跳节点ID>分别指定生成分组的节点的地址,目标节点和当前节点的下一跳ID.<唤醒状态邻居节点数>表示生成控制请求消息的节点处于唤醒状态的1跳邻居节点数目.<标志位>用于指定节点是否需要将控制消息转发到控制器,<生存时间>指定删除此消息的剩余时间.在更新消息报头中.<状态>定义节点当前状态(睡眠=0或唤醒=1).<转发>指定是否需要将消息转发到其下一跳ID.

报头格式如图3(b)所示.另外,图3(c)示出了1跳广播消息报头.

节点的邻居表:节点保存邻居表,并用网络拓扑信息更新字段.典型的邻居表包含<邻居节点ID,能量,状态,RSSI>,其中RSSI表示来自1跳邻居的接收信号指示符的链路质量.

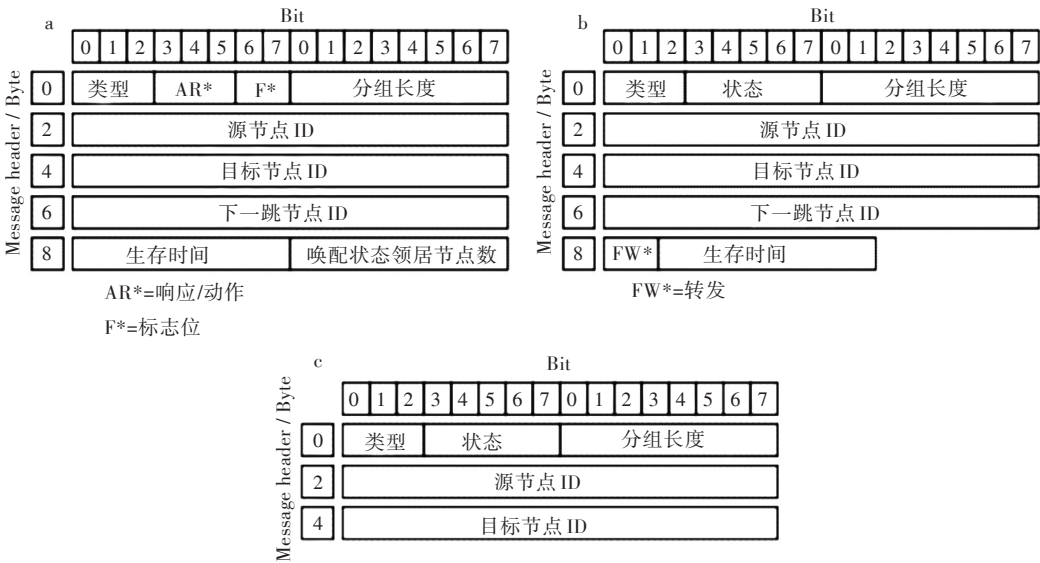


图3 消息报头格式

Fig. 3 Message headers format

3 基于睡眠调度的流表

在所提出的SDWSN中引入基于EC-CKN的睡眠调度方案^[12].

1)当1跳唤醒邻居的数目小于或等于k时,节点向其所有1跳邻居发送具有<标志位=0>和<动

作/请求=0>的唤醒请求消息.

2)当1跳唤醒邻居的数目大于k时,节点使用<动作/请求=0>和<标志位=1>向控制器发送控制请求消息.

图4示出了流表项,用于SDWSN中的实现基于EC-CKN的睡眠调度.匹配规则1和2检查消息

类型,并分别比较节点处于唤醒状态时的邻居节点. 相应的<标志位>值在第一个字节的第 7 位更新. 此外,<动作/请求>值在第一个字节的第 4 位更新.

1)当处于唤醒状态的邻居节点数量小于等于 k

匹配规则 1				匹配规则 2				动作规则 1				动作规则 2				统计	
运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	生存时间	计数器
=	0	3	0	<=	8	8	k	更新	(0+6)	2	0	更新	(0+3)	3	0	××	××
检查(类型)				比较 1 跳内处于唤醒状态的邻居节点数				更新<标志位>为 0				更新<动作/响应>为 0					

2)当处于唤醒状态的邻居节点数量大于 k

匹配规则 1				匹配规则 2				动作规则 1				动作规则 2				统计	
运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	生存时间	计数器
=	0	3	0	>	8	8	k	更新	(0+6)	2	1	更新	(0+3)	3	0	××	××
检查(类型)				比较 1 跳内处于唤醒状态的邻居节点数				更新<标志位>为 1				更新<标志位>为 0					

图 4 流表项

Fig. 4 Control-flow entries

3.1 当节点接收控制动作和请求消息时

- 1)如果<类型=0>,<动作/请求=0>,<标志位=0>:节点需要被唤醒. 因此,其将当前状态设置为唤醒状态,此后,向控制器发送具有<类型=1>和<状态=1>的更新消息. 此外,该节点广播消息以更新其所有 1 跳邻居的当前状态.
- 2)如果<类型=0>,<动作/请求=0>,和<标志位

- =1>:节点基于路由表信息将控制请求消息转发到其下一跳 ID.
- 3)如果<类型=0>和<动作/请求=1>:节点将目标地址与其自己的 ID 进行比较. 如果相等,则节点如根据<状态>位更新其状态;否则,节点将该控制动作消息转发到下一跳 ID. 图 5 示出了当节点接收控制动作和控制请求消息时的流表项.

1)当一个节点收到标志位为 0 的控制请求信息

匹配规则 1				匹配规则 2				匹配规则 3				统计	
运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值		
=	0	3	0	=	(0+3)	3	0	=	(0+6)	2	0		
动作规则 1				动作规则 2				动作规则 3					
类型	当前状态		数值	类型	偏移量	大小	数值	类型	偏移量	大小	数值	生存时间	计数器
更新			唤醒	更新	6	16	控制器 ID	更新	(0+3)	5	1	××	××

2)当一个节点收到标志位为 1 的控制请求信息

匹配规则 1				匹配规则 2				匹配规则 3			
运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值
=	0	3	0	=	(0+3)	3	0	=	(0+6)	2	1
动作规则 1				统计							
类型	偏移量	大小	数值	生存时间		计数器					
更新	6	16	下一跳 ID	××		××					

3)a. 当一个节点收到控制动作消息(目标节点 ID 与当前节点 ID 相同)

匹配规则 1				匹配规则 2				匹配规则 3			
运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值
=	0	3	0	=	(0+3)	3	0	=	4	16	ID
动作规则 1				动作规则 2				统计			
类型	当前状态		数值	类型	偏移量	大小	数值	生存时间		计数器	
更新			唤醒	更新	(0+3)	5	1	××		××	

b. 当一个节点收到控制动作消息(目标 ID ≠ 当前节点 ID)

匹配规则 1				匹配规则 2				匹配规则 3			
运算符	偏移量	大小	数值	运算符	偏移量	大小	数值	运算符	偏移量	大小	数值
=	0	3	0	=	(0+3)	3	0	≠	4	16	ID
动作规则 1				统计							
类型	偏移量	大小	数值	生存时间		计数器					
更新	6	16	下一跳 ID	××		××					

图 5 流表设计

Fig. 5 Flow table design

3.2 当控制器接收控制请求消息时

如果<类型=0>,<动作/请求=0>和<标志位=1>,则控制器获得<源节点 ID>. 基于<源节点 ID>的1和2跳邻居信息,控制器使用基于EC-KCN的睡眠调度算法决定唤醒或睡眠状态,并返回此动作消息与更新的<状态>到特定节点.

3.3 当节点接收到更新消息时

如果<类型=1>和<转发=1>,则节点将该消息转发到其下一跳. 然而,如果<目标节点 ID>与其自己的ID相同,则节点丢弃消息并更新<转发=0>.

4 性能评估

该部分通过实验在以下方面评估所提出的SDWSN的性能:

- 1)平均响应时间:在睡眠调度中的每个历元开始时更新SDWSN中的每个节点的状态所需的时间.
- 2)控制流量:在网络中循环的控制动作,控制请求,更新和广播消息方面的控制流的总大小(以字节为单位).

仿真设置:使用WSN模拟器NetTopo进行仿真. 结果在100个不同的拓扑结构上取平均值. 仿真参数如表2所示. 仿真结果如图6所示,图6(a)、(b)分别为本文提出的SDWSN和传统的WSN,图6(a)中处于网络中心的空心节点为控制器,浅黑色节点为唤醒状态节点,深黑色节点为睡眠状态节点,黑色实线为控制流的路径,灰色实线为真实链路. 图6(a)中所有计算通过控制器完成并生成数据包向下转发,节点通过控制流收到数据包后与自身流表进行校验,判断执行转发或者睡眠/唤醒动作. 图6(b)中没有控制器,每个节点通过接收两跳以内邻居节点的信息,计算决定自身是否进入睡眠状态.

表2 仿真参数

Tab. 2 Simulation parameters

参数 parameter	数值 value
网络大小 / m ²	400×300
节点数量	100
控制器坐标	(184, 196)
睡眠调度中k值	1~5
传输范围 / m	60
时间历元 / min	1
数据包大小 / byte	12
数据包数量	1 000

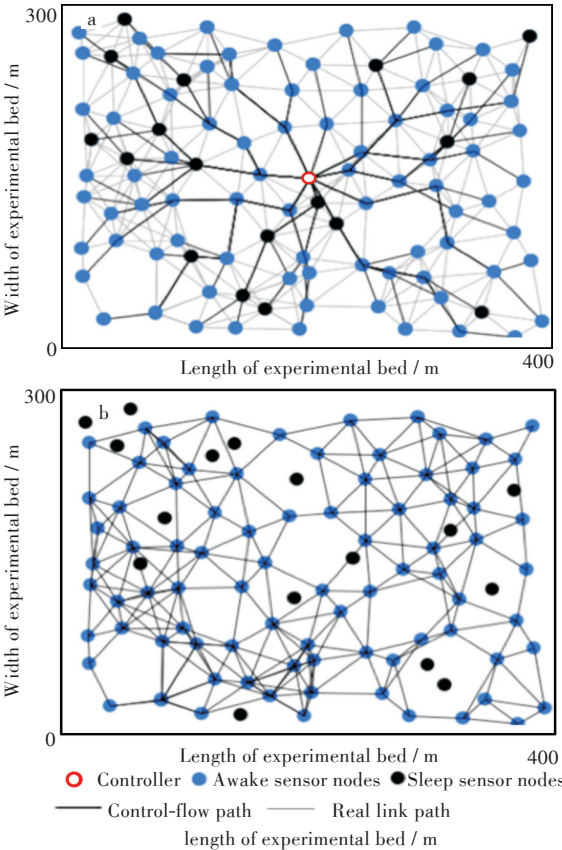


图6 仿真对比图 (a)软件定义无线传感网络;
(b)传统无线传感网络

Fig. 6 Simulating contrast chart
(a)Software-defined wireless sensor network;
(b)Traditional wireless sensor network

4.1 平均响应时间对SDWSN性能的影响

图7反应了25个传感器节点、单个控制器情况下,与本文提出的SDWSN与SD EC-KCN相比,平均响应时间的优势. 在每个周期开始时从每个传感器节点向控制器发送控制请求,流表项的设计将控制动作和控制请求最小化,以决定传感器节点的睡眠或唤醒状态.

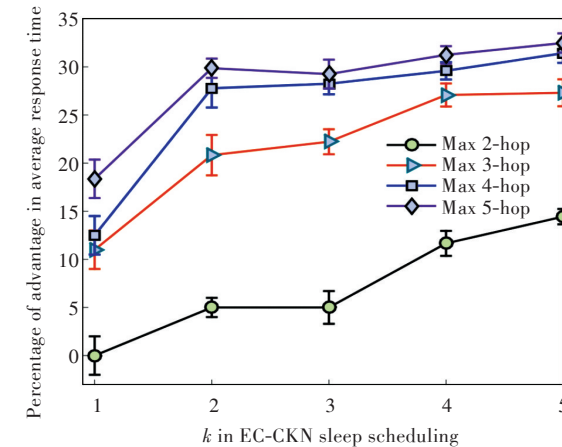
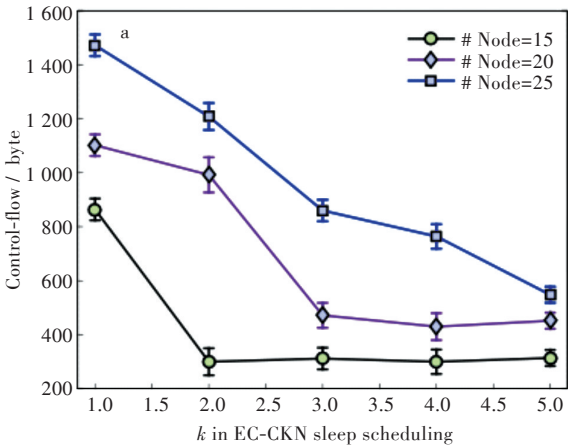


图7 平均响应时间的优势

Fig. 7 Advantages of average response time

由图 7 可知,在基于 EC-CKN 的睡眠调度中随着 k 值的增加观察到显著的优点.原因是在 SD EC-CKN 的睡眠调度中,所有节点必须向控制器发送请求.而本文中的 SDWSN,随着 k 值增加,传感器节点控制请求的数量将会减少.然而,广播和更新消息的数量增加.由于与控制动作和控制请求消息相比,更新或广播消息需要较少的跳数,因此用于整体控制流的负载降低.结果,平均响应时间也随着 k 值的增加而减小.当所有节点到控制器最大跳跃距离增加时,所提出的方法相对于在数据平面中具有相同数目节点的 SD EC-CKN 具有显著的优势.



4.2 控制流量对SDWSN性能的影响

图 8 显示出了节点距离控制器最大 3 跳和最大 5 跳情况下具有不同 k 值的控制流量.显然,部署的传感器节点数量较多处,控制流量也增加.如第四节所述,控制请求和控制动作消息随着 EC-CKN 中 k 值的增加而减小.虽然广播和更新消息的数量增加,但是控制动作以及控制请求的数量更大,因此,控制流量增加.比较看来,对于所有的节点,距离控制器越远,控制流量越大.原因是随着部署的传感器节点远离控制器,传感器邻居节点数大于或者等于 k 的可能性变小.因此,从所有传感器节点到控制器的控制请求增加,结果是节点离控制器越远,控制流量越大.

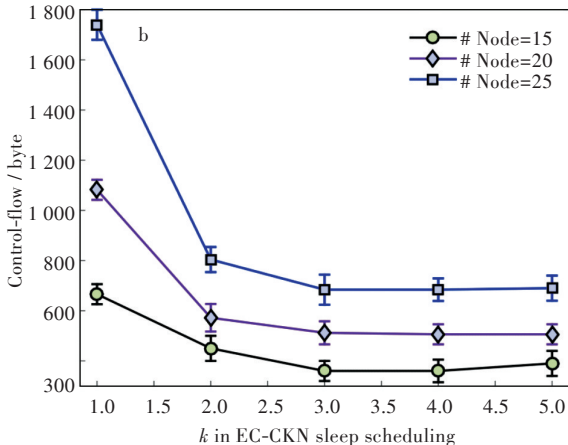


图 8 不同 k 值对应的控制流量:(a)距离控制器最大 3 跳;(b)距离控制器最大 5 跳

Fig. 8 Different k values corresponding to the control flow

(a) Deployed nodes with maxmum 3-hop distance-to-controller; (b) Deployed nodes with maxmum 5-hop distance-to-controller

5 结 语

在 WSN 中引入了 SDN 架构,并且在此架构中,引入具有低控制负载的睡眠调度的方案,在延长网络生命周期的同时,克服 SDWSN 中控制负载过高的缺陷.提出的流表项设计在控制流量和平均响应时间方面表现出显著的优势,根据 k 值的变化,平均响应时间降低了 5.1% 到 32.6%.实验结果还表明,随着部署的传感器节点远离控制器,控制流量增加.总体上,笔者所提出的方案能够最小化整个网络中总控制流量,降低控制负载.

基于 SDN 技术的无线传感器网络只在小型无线传感器网络中进行研究,控制平面只需一个控制器.而大型无线传感器网络中,控制器需要为每一项数据业务制定优化的路由策略,运算压力较大,且这一压力会随着传感器网络节点数量的增加呈几何级上升,这时可能需要多个控制器进行

控制,这种情况还需要进行更深一步分析,也会引起其他问题,如网络带宽、数据冗余等问题.这些问题对于网络架构的影响也需要进行考虑.本文的研究内容只是其中的一小部分,由于时间和能力有限,还有许多问题需要在现有的研究基础上进行更进一步研究与扩展.

参考文献:

[1] HAQUE I T, ABU-GHAZALEH N. Wireless software defined networking: a survey and taxonomy[J]. IEEE Communications Surveys & Tutorials, 2016, 18(4): 2713–2737.

[2] LUO T, TAN H P, QUEK T Q S. Sensor openflow: enabling software-defined wireless sensor networks[J]. IEEE Communications Letters, 2012, 16(11): 1896–1899.

[3] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. Openflow: enabling innovation in campus networks

[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69–74.

[4] GALLUCCIO L, MILARDO S, MORABITO G, et al. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks [C]//Computer communications, 2015. Hong Kong: IEEE, 2015: 513–521.

[5] DEGANTE A, ASLAN M, MATRAWY A. Smart wireless sensor network management based on software-defined networking [C]// Communications, 2014. Kingston: IEEE, 2014: 71–75.

[6] VISSICCHIO S, CITTADINI L. Flip the (flow) table: Fast lightweight policy-preserving sdn updates [C]// Computer communications, 2016. San Francisco: IEEE, 2016: 1–9.

[7] ZENG D, LI P, GUO S, et al. Minimum-energy reprogramming with guaranteed quality-of-sensing in software-defined sensor networks[C]//Communications, 2014. Sydney:IEEE,2014: 288–293.

[8] MIYAZAKI T. Dynamic function alternation to realize robust wireless sensor network [J]. International Journal of Handheld Computing Research, 2012, 3 (3): 17–34.

[9] DEOLIVEIRA B T,GABRIELL B,MARGI C B.TinySDN: Enabling multiple controllers for software-defined wireless sensor networks [J]. IEEE Latin America Transactions, 2015, 13(11): 3690–3696.

[10] NATH S, GIBBONS P B. Communicating via fireflies: geographic routing on duty-cycled sensors [C]// Information processing in sensor networks, 2007. Cambridge: IEEE, 2007: 440–449.

[11] WANG Y, CHEN H, WU X, et al. An energy-efficient SDN based sleep scheduling algorithm for WSNs [J]. Journal of Network and Computer Applications, 2016, 59: 39–45.

[12] WANG L, YUAN Z, SHU L, et al. An energy-efficient CKN algorithm for duty-cycled wireless sensor networks [J]. International Journal of Distributed Sensor Networks, 2012(3):184–195.

[13] CERPA A, WONG J L, KUANG L, et al. Statistical model of lossy links in wireless sensor networks [C]// The 4th international symposium on information processing in sensor networks, 2005. Los Angeles: IEEE, 2005: 81–88.

[14] BERA S, MISRA S, ROY S K, et al. Soft-WSN: Software-defined WSN management system for IoT applications [J]. IEEE Systems Journal, 2016 (99): 1–8.

[15] OLIVIER F, CARLOS G, FLORENT N. SDN based architecture for clustered WSN [C]//Innovative Mobile and Internet Services in Ubiquitous Computing, 2015. Blumenau: IEEE, 2015: 342–347.

[16] 张朝昆,崔勇,唐嵩伟,等. 软件定义网络(SDN)研究进展[J]. 软件学报,2015,26(1):62–81.

ZHANG C K, CUI Y, TANG H Y, et al. Research progress of software defined network [J]. Journal of Software, 2015, 26(1): 62–81.

本文编辑:陈小平