

一种基站可移动传感器网络再编程协议

方效林, 高 宏, 李建中

(哈尔滨工业大学计算机科学与技术学院, 黑龙江哈尔滨, 150001)

摘 要: 本文针对基站可移动传感器网络实现了一再编程协议 MovPro. 该协议可以将新的二进制程序通过多跳的形式下发到网络内的节点上并使之运行. 该协议的大致过程描述为, 当基站在网络内移动时, 基站将数据发送给它移动轨迹上的节点. 节点收到部分二进制代码后通过窗口交换的形式将二进制代码传播到整个网络. MovPro 是第一个在基站可移动传感器网络的真实系统. 本文通过多种方式减少通信开销, 并通过二级存储的方式减少外部 flash 的写次数. 实验表明 MovPro 适用于基站可移动传感器网络.

关键词: 无线传感器网络; 再编程; 可移动基站

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2013)05-1007-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.05.028

MovPro: Data Dissemination for Reprogramming in Wireless Sensor Network with Mobile Sink

FANG Xiao-lin, GAO Hong, LI Jian-zhong

(School of Computer Science and Technology Harbin Institute of Technology, Harbin, Heilongjiang 150001)

Abstract: A reprogramming protocol named MovPro was proposed. MovPro is used to disseminating large block of binary image from the mobile sink to all other sensor nodes in wireless sensor network over multiple hops. When the sink walks in the sensor network, it reliably transmits the new binary image to the sensor nodes along its traveling trajectory. After receiving part of the new binary image, these nodes reliably propagate the received image to the rest of the network. MovPro is the first real reprogramming system in sensor networks with mobile sink. We apply window transfer method to reduce the radio transmissions so as to save energy. And a BitVector method is also adopted in order to reduce the number of flash writes. Experiments demonstrate that MovPro works well in sensor networks with mobile sinks.

Key words: wireless sensor network; reprogramming; mobile sink

1 引言

无线传感器网络由大量传感器节点组成, 其布置的区域可能处于自然条件较恶劣的野外或者人类不适宜工作的环境中. 由于用户需求变更, 网络功能更新以及软件故障修复等原因, 传感器网络需要对节点运行的程序进行更新, 即对网络进行再编程. 然而, 将成千上万的传感器节点收回再编程需要消耗大量的时间和人力物力, 而且对布置在极端环境中的传感器节点收回处理困难性非常大, 因此无线再编程协议研究成为传感器网络的一个重要研究内容^[1].

本文研究基站可移动传感器网络再编程协议. 基站可移动传感器网络具有以下优点. 首先, 可移动基站可

以提高数据收发的可靠性. 传感器节点可在信号强度较高时与基站进行数据交换提高可靠性. 其次, 可移动基站可降低网络的通信负载. 节点与基站距离较近时通信可以降低数据传输的跳数, 从而减少网络通信量. 再次, 可移动基站可避免负载不平衡问题. 在静态网络中, 距离基站近的节点通信和存储负载更高. 而且, 可移动基站还可应用于不连通网络.

现有的再编程协议都不适用于基站可移动网络, 本文提出一个基站可移动传感器网络再编程协议 MovPro. 该协议将包含两个阶段: (1) 基站将新的二进制程序分成数据包, 并将数据包按顺序下发到传感器节点; (2) 通过数据交换的方式, 将二进制程序传输到整个网络^[2]. 本文通过窗口数据交换技术来减少通信开销, 并通过监

收稿日期: 2012-07-23; 修回日期: 2013-01-14

基金项目: 国家重点基础研究发展计划(973计划)项目(No. 2012CB316200); 国家自然科学基金重点项目(No. 61033015, No. 60933001); 国家自然科学基金重大项目(No. 61190115); 国家自然科学基金(No. 61173022, No. 61100030); 国家自然科学基金国际(地区)合作与交流项目(No. 60831160525); 中国博士后科学基金(No. 20110491060)

听手段抑制信息广播次数. 由于外部 flash 写操作开销较大, 本文使用二级存储记录方式减少外部 flash 的写操作次数.

2 相关工作

XNP 是第一个基于 TinyOS 操作系统的再编程系统^[3]. 它只能对一跳内的节点进行再编程. MOAP 是一个远程多跳再编程协议^[4]. 该协议是基于完整转发技术的再编程协议. 节点只有接收到完整的二进制程序后才进行转发.

Deluge 是传感器网络应用最广泛的再编程系统^[5]. 该系统中, 二进制程序以页的形式传输. 节点只能按编号从小到大逐一接收程序页, 此方式可能浪费很多接收机会. MNP 和 Freshet 是基于 Deluge 的协议. MNP 通过合理选择发送端从而降低在密集网络中的通信开销. Freshet 使用睡眠机制减少在大规模网络中的能量开销. Stream 也是基于 Deluge 的协议^[6]. 它将重量级的再编程协议预装载在外部 flash 中, 以减少运行时系统开销. 节点接收到再编程命令后, 从外部 flash 加载重量级的再编程协议并进入再编程模式.

Mate 是针对传感器网络而设计的虚拟机^[7,8]. 它为用户提供接口, 使得再编程代码非常简短. 然而虚拟机的接口较粗, 使得生成的程序效率较低. 而且在资源有限的节点上运行虚拟机需要非常大的开销.

Elon 通过扩展 NesC 编译器区分可替换组件和 TinyOS 内核组件^[9]. Elon 建立一系统转移表, 记录内核组件的入口地址. 可替换组件可以通过系统转移表调用内核组件. Reijers 等人使用 UNIX 中的 diff 命令区分新旧二进制程序^[10], 通过只需发送不同的部分来减少数据传输量. 不同的部分通过插入、复制和修复操作更新到旧程序中. 但是这种方法会引起大量的地址偏移, 导致大量的地址偏移修复信息. Zephyr 是一个增量式的再编程协议^[11,12]. 它通过对新旧程序使用 Rsync 算法进行比较, 产生中间代码并下发给节点. 它与文献^[10]中的方法具有相同的地址偏移问题. Flexcup 在节点对二进制程序进行链接^[13]. 基站只需要将有更新的组件下发给节点, 节点对程序进行链接生成可执行程序. 这种方法也会引起地址偏移问题.

Koshy 和 Pandey 针对地址偏移问题提出一种解决方法^[14]. 它为每个函数保留足够的代码空间, 某些函数的更新不会改变剩余函数的入口地址. 这种方法极大地减少了地址偏移带来的信息传输量, 但是它会占用大量的代码空间.

SYNAPSE 使用喷泉码下发二进制程序^[15]. 这一方法通过压缩解压的方法进行数据传输.

以上协议都是基于静态传感器网络的再编程协

议, 它们并不能完全适用于基站可移动传感器网络. ReMo 是可移动传感器网络再编程协议^[16]. 其是与本文最相关的工作之一. 不同的是本文针对的是基站可移动网络, 而 ReMo 针对所有节点可移动网络, 两者的通信方式完全不同. ReMo 基于概率广播模型, 它通过估计链路质量和相对距离选择最佳节点进行代码交换. ReMo 在 SunSPOT 上实现, 而本文的 MovPro 是在 TinyOS 操作系统实现.

Gappa 利用无人机进行对野外传感器网络进行再编程^[17], 其与本文工作最相关. 但是它只是提出了一再编程架构, 并没有在真实的传感器节点上实现. 本文的 MovPro 是在真实节点 Telosb 实现的再编程协议.

3 MovPro 再编程协议

3.1 MovPro 简介

MovPro 需要考虑通信开销问题. 由于二进制程序数据量大, 会引起大量的通信, 消耗大量能量, 因此在数据下发过程中需要尽可能减少数据通信次数.

MovPro 还需要考虑存储空间的问题. 二进制程序不能直接暂存在有限的内存中, 一般暂存在外部 flash 中. 对外部 flash 操作与内存操作不同. 外部 flash 读操作速率几乎与内存读操作一致, 但是在外部 flash 写操作之前需进行擦除操作, 而且 flash 写操作比 RAM 写操作慢一个数量级, 因此再编程协议需要考虑对外部 flash 写操作次数. 每个节点的外部 flash 可分为几个卷, 卷号可在分卷的时候指定. MovPro 使用两个卷, 其中一卷用于暂存二进制程序, 另一卷用于记录接收到哪些数据包.

MovPro 包含两个阶段: (1) 基站至节点阶段; (2) 节点到节点阶段. 基站到节点传输阶段, MovPro 通过移动基站将数据下发给网络. 节点到节点传输阶段, 二进制程序以窗口交换的形式扩散到整个网络.

3.2 基站至节点阶段

基站接收到再编程命令后, 将指定的二进制程序划分为多个数据包, 并将数据包下发给节点. 基站与节点之间通信的消息格式为 (unihash, pktNum, imgNum, packet), 其中 unihash 二进制程序的唯一标识, pktNum 表示数据包编号, imgNum 表示存储二进制程序的外部 flash 分卷号, packet 表示数据包内容, 如图 1 所示. 图中, TOSH_DATA_LENGTH 是 TinyOS 通信可携带的消息长度.

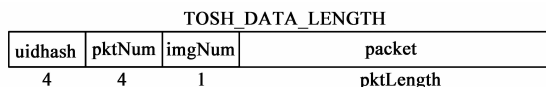


图1 基站至节点消息格式

当节点收到基站发来消息, 它首先判断自身是否

正在准备接收标识为 unihash 的程序.若未准备接收,则节点将卷号为 imgNum 和 BitMap 的外部 flash 分卷擦除,为暂存新的二进制程序做好准备.若已准备好接收标识为 unihash 的程序,则判断是否已经接收过该数据包.未接收过则将数据包写入 imgNum 分卷并记录,否则忽略. imgNum 分卷用于存储二进制程序. BitMap 分卷用于记录数据包是否收到,收到的置 1,否则为 0.由于 flash 写操作开销大,本文使用二级存储记录方式. MovPro 在内存中开辟一空间 BitVector,用作内存和外部 flash 之间的缓存.这一记录方式类似于内存缓存磁盘数据的关系.

节点将数据包写入 BitMap 分卷后,向基站发送 ACK 消息.基站只有接收到 ACK 消息后才发送下一个数据包.基站发送数据后,如果多个节点同时发送 ACK,会造成数据冲突. MovPro 通过概率 ACK 的方式减少数据冲突.基站发送数据包后,设置一计时器.当计时器触发时仍未收到 ACK 消息,则重传.节点以 k/N 的概率回复 ACK,其中 k 是一可调参数, N 为邻居节点个数. k 越大,则回复 ACK 概率越高.图 2 表示 11 个 Telosb 节点的网络中 k 对数据包重传的影响.其中白色和灰色柱条表示不同大小的二进制程序的影响.从图 2 可知,随着 k 增大重传的次数急剧减少.当 k 为 3 时,已有 99.7% 的数据包一次性发送成功.

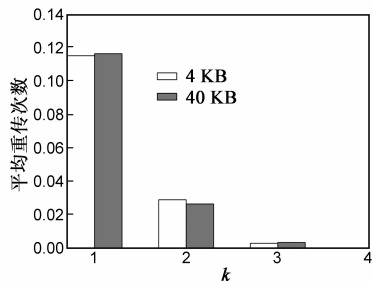


图2 不同 k 对数据包重传的影响

基站将整个二进制程序发送到网络后,向全网广播一个准备接收新程序消息.该消息包含 uidhash, size 和 imgNum,其中 uidhash 是二进制程序的唯一标识, size 表示新程序大小, imgNum 表示将存入的外部 flash 分卷号.

3.3 节点至节点阶段

3.3.1 数据包记录交换

节点接收到数据包后,通过周期性广播数据包记录方式告知邻居节点.节点通过交换记录进行判断是否需要请求数据包或者发送数据包.节点交换记录时可能会引起信息冗余.多个节点广播相同的记录将导致记录重复广播,本文采用抑制广播重复记录的方法减少通信^[2].抑制广播重复记录的方法如图 3 所示. s 与 s' 为相邻节点,两节点分别在各自的后半周期生成

随机时间.假设 s 与 s' 广播的数据包记录相同,在第一个周期,节点 s 生成的随机时间 t_1 早于节点 s' 的 t_1' ,则 s' 抑制广播.在第二个周期,节点 s 生成的随机时间 t_2 晚于节点 s' 的 t_2' ,则 s 抑制广播.

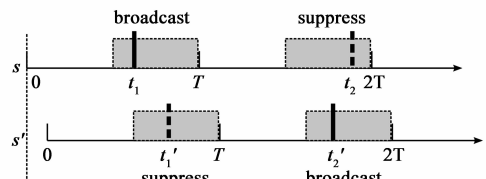


图3 记录广播策略

3.3.2 数据包传输

基站可移动网络与静态网络不同,基站可移动网络中,节点接收到的数据包可能是无序的.本文采用窗口的方式进行记录交换.窗口表示为 (offset, width), 其分别为偏移和长度.

节点周期性地从分卷号为 BitMap 的外部 flash 分卷检查数据包记录,并选取最长为 BitVector 大小的窗口片段进行广播,如图 4 所示.窗口大小设为 BitVector 大小是为了满足二级存储记录策略中 BitVector 的限制.一次广播可携带多个窗口信息.节点 s' 接收到窗口信息后向 s 请求数据包.图 4 中灰色部分表示数据包已接收,节点 s 将向 s' 广播的窗口信息为 w_0', w_1', w_1'' 和 w_2 .窗口 w_1 划分为两个窗口 w_1' 和 w_1'' ,其中 w_1' 刚好满足 BitVector 大小的限制.

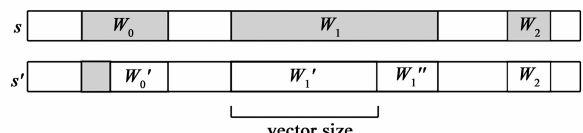


图4 窗口传输

假如数据包不总是连续的,这会使窗口大小非常小,从而导致广播次数增加.以图 5 为例,图中窗口非常小但却较集中,这会导致广播窗口信息的次数非常多.本文采用最小窗口限制的方式进行窗口信息广播.令最小窗口限制为 w_{min} ,小于 w_{min} 的窗口将会以比特向量的形式广播信息.节点也将以比特向量的形式请求数据包.图 5 中将广播 3 个比特向量 v_0, v_1, v_2 以及一个窗口 w_i .

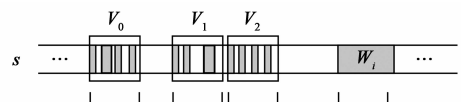


图5 最小窗口传输

3.3.3 数据包合并

TinyOS 编程中无线通信的消息长度是固定的.消息能携带的数据长度为 TOSH_DATA_LENGTH.例如,默认情况下, Telosb 和 Micaz 可携带数据长度为 28 字

节.由图 1 可知,基站至节点的消息中包含 uidhash, pktNum 和 imgNum.但是在节点至节点传输过程中,消息中只需要包含 pktNum 即可,此时会有 5 字节的剩余空间,如图 6 所示.

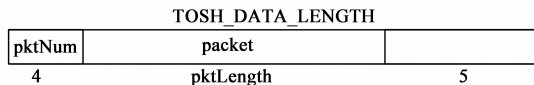


图6 未合并消息格式

将多个原数据包进行合并,形成新的消息格式,可充分利用消息可携带的数据空间,如图 7 所示.新的消息格式包含偏移 offset 以及相对编号 num.新的数据包的真实偏移可表示为 $offset + num * mLen$.易得,以新消息格式发送数据,可节省约 20% 通信次数.

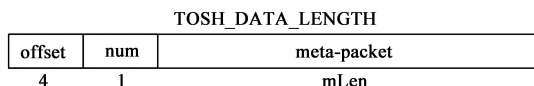


图7 合并后的新消息格式

4 系统实现及实验结果

MovPro 在 Telosb 传感器节点上实现. Telosb 节点由 8MHz TI MSP430 微处理器, 10kB RAM, 48KB 可编程代码区以及 1MB 的外部 flash 组成. 节点通信采用 IEEE 802.15.4/ZigBee 标准. 通信频率在 2.4GHz 至 2.4835GHz 之间, 通信速率为 250kbps.

MovPro 的再编程命令与 Deluge 的兼容^[18]. 再编程命令包含“-p”,“-i”,“-r”,“-d”,“-dr”和“-e”等可选指令,其分别表示:(1)检查节点外部 flash 状态;(2)写二进制程序到外部 flash;(3)重启;(4)下发新的二进制程序;(5)下发新的二进制程序并重启;(6)擦除外部 flash.

4.1 再编程所需时间

本文通过两种 11 个 Telosb 节点组成的网络测试系统性能. 第一种网络是线性布置网络,即节点排成一条直线. 第二种网络是随机散布网络,即节点随机布置在区域内. 由于网络布置的空间有限,节点发送功率等级设为 3,以降低通信功率增加网络跳数. 本节通过下发两种不同大小的二进制程序,测试 MovPro 的再编程所需时间. 测试的两种二进制程序大小分别为 3696 字节

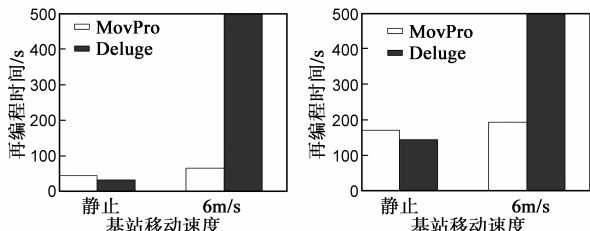


图8 线性布置网络再编程时间比较

和 40128 字节. 图 8 和图 9 分别表示线性布置网络和随机散布网络再编程所需时间. 通过与 Deluge 再编程系统进行比较,可见在静态网络中,本文的 MovPro 再编程所需时间略长于 Deluge. 但是,在基站可移动网络中,当基站的移动速度为 6m/s 时,采用页传输技术的 Deluge 由于不能完成页传输导致再编程失败,而 MovPro 与静态网络所需的再编程时间几乎相同.

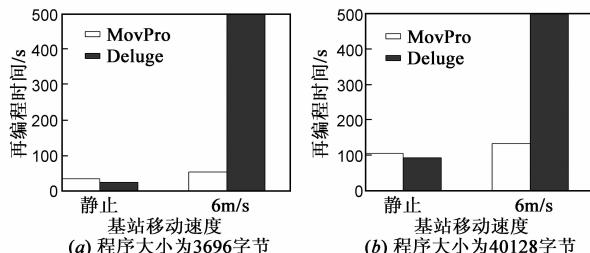


图9 随机散布网络再编程时间比较

4.2 通信开销

MovPro 的通信开销主要由窗口交换组成. 在基站至节点阶段,基站移动轨迹上的节点接收到部分数据包. 在节点至节点阶段,初始时,节点交换的窗口较小,可能导致大量的窗口信息交换. 但是随着时间推移,节点接收到的数据包越来越多,数据包片段越来越连续,从而使窗口变大,窗口信息交换变少. 随着窗口变大,数据包可以以新消息格式进行传输,从而减少通信次数. 使用新消息格式传输可节省约 20% 通信次数.

4.3 存储空间开销

MovPro 将占据两个分卷的外部 flash 空间,一个用于暂存数据包,一个用于记录数据包. 外部 flash 空间较大,其使用量与分卷划分大小有关. 而内存空间较小,一般只有几 K 至十几 K 大小,因此再编程协议需考虑内存空间开销. 本文采用二级存储的方式记录接收了哪些数据包,其内存使用量为 BitVector 大小,其设为 64 字节.

5 结论

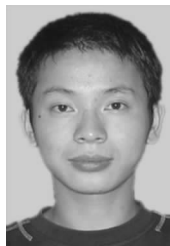
本文针对基站可移动传感器网络实现了一再编程系统 MovPro. 该系统包含基站至节点阶段和节点至节点阶段两个阶段. 在基站至节点阶段,节点采用概率 ACK 的方式减少数据冲突. 在节点至节点阶段, MovPro 采用窗口信息交换、抑制重复记录发送以及数据包合并的方式减少通信次数. MovPro 采用二级存储技术记录接收了哪些数据包,以减少外部 flash 写操作的次数.

参考文献

[1] Qiang Wang, Yaoyao Zhu, Liang Cheng. Reprogramming wireless sensor networks: challenges and approaches[J]. Network,

- IEEE, 2006, 20(3): 48 – 55.
- [2] Philip Levis, Neil Patel, David Culler, Scott Shenker. Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks[A]. In Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation[C]. Berkeley, CA, USA, 2004. 15 – 28.
- [3] Inc. Crossbow Technology. Mote in-network programming user reference version 20030315[S]. 2003.
- [4] Thanos Stathopoulos, John Heidemann, Deborah Estrin. A remote code update mechanism for wireless sensor networks[R]. In Technical Report CENS-TR-30, 2003.
- [5] Jonathan W Hui, David Culler. The dynamic behavior of a data dissemination protocol for network programming at scale[A]. In Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM[C]. New York, NY, USA, 2004. 81 – 94.
- [6] R K Panta, I Khalil, S Bagchi. Stream: low overhead wireless reprogramming for sensor networks[A]. In 26th IEEE International Conference on Computer Communications[C]. Anchorage, Alaska, USA, 2007. 928 – 936.
- [7] Philip Levis, David Culler. Mate: a tiny virtual machine for sensor networks[A]. In Proceedings of the 10th international conference on Architectural support for programming languages and operating systems, ACM[C]. New York, NY, USA, 2002. 85 – 95.
- [8] Philip Levis, David Gay, David Culler. Active sensor networks[A]. In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation[C]. Berkeley, CA, USA, 2005. 343 – 356.
- [9] Wei Dong, Yunhao Liu, Xiaofan Wu, Lin Gu, Chun Chen. Elon: enabling efficient and long-term reprogramming for wireless sensor networks[A]. In SIGMETRICS 2010[C]. New York, NY, USA, 2010. 49 – 60.
- [10] Niels Reijers, Koen Langendoen. Efficient code distribution in wireless sensor networks[A]. In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, WSNA'03, ACM[C]. New York, NY, USA, 2003. 60 – 67.
- [11] Rajesh Krishna Panta, Saurabh Bagchi, Samuel P Midkiff. Zephyr: efficient incremental reprogramming of sensor nodes using function call indirections and difference computation[A]. In Proceedings of the 2009 conference on USENIX Annual technical conference, USENIX'09[C]. Berkeley, CA, USA, 2009. 411 – 424.
- [12] Jaemin Jeong. Incremental network programming for wireless sensors[A]. In Proceedings of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks[C]. Santa, Clara, CA, 2004. 25 – 33.
- [13] Pedro Josle Marrlon, Matthias Gauger, Andreas Lachenmann, Daniel Minder, Olga Saukh, Kurt Rothenmel. Flexcup: a flexible and efficient code update mechanism for sensor networks[A]. In Proceedings of the Third European Workshop on Wireless Sensor Networks[C]. Zurich, Switzerland, 2006. 212 – 227.
- [14] Joel Koshy. Remote incremental linking for energy-efficient reprogramming of sensor networks[A]. In Proceedings of the second European Workshop on Wireless Sensor Networks[C]. Istanbul, Turkey, 2005. 354 – 365.
- [15] M Rossi, G Zanca, L Stabellini, R Crepaldi, A F Harris, M Zorzi. Synapse: a network reprogramming protocol for wireless sensor networks using fountain codes[A]. In 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks[C]. San Francisco, CA, USA, 2008. 188 – 196.
- [16] Pradip Deand, Yonghe Liu, Sajal K Das. Energy-efficient reprogramming of a swarm of mobile sensors[J]. IEEE Transactions on Mobile Computing, 2010, 9(5): 703 – 718.
- [17] Limin Wang, Sandeep S. Kulkarni. Gappa: gossip based multi-channel reprogramming for sensor networks[A]. In DCOSS 2006[C]. San Francisco, California, USA, 2006. 119 – 134.
- [18] Chieh-Jan Mike Liang, Razvan Musaloiu-E. Deluge t2-programming manual[J/OL]. <http://www.tinyos.net/tinyos-2.x/doc/html/deluge-t2-manual.html>, 2008.

作者简介



方效林 男, 博士, 主要研究领域为海量数据计算、无线传感器网络、CPS。

E-mail: xlforu@hit.edu.cn



高宏 女, 博士, 教授, 主要研究领域为海量数据计算、无线传感器网络、CPS、数据质量、数据挖掘。

E-mail: honggao@hit.edu.cn