

事件驱动传感器网络中基于数据聚合的自适应路由算法

聂雅琳^a, 刘三阳^b, 陈智斌^c, 齐小刚^b

(西安电子科技大学 a. 计算机学院, b. 数学与统计学院, c. 微电子学院, 西安 710071)

摘要: 为优化事件驱动传感器网络总能耗, 提出一个基于数据聚合的自适应路由算法, 它能够实现低控制开销的事件域节点分布式成簇, 计算并借助于路由汇聚中心, 建立一棵基于事件的近似 Steiner 树, 有效减少网内数据分组与控制分组的传输量. 理论分析与实验表明, 该算法的路由结构建立与维护开销较少, 能优化数据聚合效率, 实现高效的数据收集, 提升网络性能.

关键词: 事件驱动传感器网络; 簇; 数据聚合; 路径重叠

中图分类号: TP393

文献标志码: A

Adaptive routing algorithm based on data aggregation in event-driven wireless sensor networks

NIE Ya-lin^a, LIU San-yang^b, CHEN Zhi-bin^c, QI Xiao-gang^b

(a. School of Computer Science and Technology, b. School of Mathematics and Statistics, c. School of Microelectronics, Xidian University, Xi'an 710071, China. Correspondent: LIU San-yang, E-mail: liusanyang@126.com)

Abstract: To optimize the total energy consumption for event-driven sensor networks, an adaptive routing algorithm based on data aggregation is proposed, which clusters nodes within event areas in a distributed manner with low control overhead, and builds an approximation Steiner Tree based on events in the help of the routing aggregation center to decrease the amount of transmission for data packets and control packets effectively. Theory analysis and experiments show that the proposed algorithm has better overhead on routing structure building and maintenance, optimizes the data aggregation ratio, achieves energy-efficient data collection, and promotes network performance eventually.

Key words: event-driven sensor networks; cluster; data aggregation; routes overlapping

0 引言

数据传输耗能是传感器网络能量消耗^[1]的主要部分, 与网内传输数据量密切相关. 网内数据聚合技术^[2-3]能有效减少网内传输数据量, 大大降低数据传输耗能. 对于一些统计查询类应用, 如 MIN、SUM 等, 不论原始数据量是多少, 其聚合数据大小固定, 此类聚合被称为完全聚合^[3]. 节点成簇^[4-7]能高效地实现数据的聚合, 增强网络可扩展性. 将簇优化与数据聚合优化结合起来研究新的传感器网络路由节能技术是目前的一个研究热点.

传感器网络根据数据收集的方式可分为两大类: 周期性数据收集传感器网络和事件驱动传感器网络^[8]. 在事件驱动传感器网络中, 若网内数据聚合采用完全聚合方式, 则网络路由结构的最优化等价于

寻找一棵包含事件域节点的 Steiner 树^[9]. 已有研究表明, 从一幅图中寻找连接子集的 Steiner 树问题是一个 NP-难问题, 只能采用启发式的方法来获得该问题的近似解^[10-11], 各种针对事件驱动传感器网络的近似 Steiner 树建立方案被相继提出^[12-15].

本文重点研究事件驱动传感器网络中针对完全数据聚合的路由策略. 受 InFRA^[15]算法的启发, 提出一个新的基于数据聚合的自适应路由算法 (ARA). 该算法能够实现事件域节点的分布式成簇, 具有较小的路由结构建立与维护开销, 通过简单的启发式策略, 可保证所建事件簇与 Sink 间的最短路径能够较大程度地重叠, 获得一棵近似 Steiner 树. 理论分析与实验表明, ARA 算法具有较低的控制开销和较好的数据聚合效率, 能大大提升网络性能.

收稿日期: 2013-05-04; 修回日期: 2013-11-10.

基金项目: 国家自然科学基金项目(61373174); 国家自然科学基金青年基金项目(11205080).

作者简介: 聂雅琳(1981—), 女, 博士生, 从事无线传感器网络算法的研究; 刘三阳(1959—), 男, 教授, 博士生导师, 从事最优化、网络算法等研究.

1 相关工作

针对周期性数据收集传感器网络, Heinzelman等^[4]提出了LEACH算法. 该算法是最早的成簇算法之一, 能有效地延长网络生存期. 之后的许多成簇算法均基于它的基本框架而获得. Soro等^[5]提出的CPCP算法以覆盖率作为准则进行簇头的选择及簇成员的休眠调度, 能够尽量延长网络的高覆盖率工作时间. DACP^[6]是一个基于数据预测和成簇的高效率数据聚合协议, 簇头充分利用预测数据来决定是否发送聚合数据, 改进了数据聚合的效率. Jung等^[7]提出了一个基于网络状态的动态成簇技术, 提高了数据聚合效率和能量利用率.

SPT^[12]使得每个事件域内的节点将其采样数据沿着最短路径传输至Sink, 机会数据聚合(Opportunistic data aggregation)发生在每个路径交叉节点上. CNS^[12]算法选取距离Sink最近的节点为聚合节点, 其余节点的数据在该节点聚合后沿着最短路径发送至Sink. 基于最短路径树和簇结构, DRINA^[13]使各个事件簇的路径能较大程度地重叠, 找到一棵近似Steiner树, 但该树的建立与事件的发生顺序有关, 且自适应性较差. Villas等^[14]提出了3种路由建立策略YEAST-CF/FF/BC, 均能建立与事件发生顺序无关的近似Steiner树. InFRA^[15]算法能实现事件域节点的自组织成簇, 建立一棵从簇头到Sink的动态最短路径树, 并能增大路径的重叠程度, 具有较好的数据聚合效率. 但是InFRA算法存在以下缺点: 1) 簇头选举过程的控制开销较大; 2) 为了使所有节点获得到簇头的距离信息, 簇头需向全网洪泛控制分组. InFRA算法的网络可扩展性较差, 同时也不适合于事件发生频率较高的传感器网络.

2 基于数据聚合的自适应路由算法(ARA)

ARA算法适用于事件驱动传感器网络, 由4个阶段构成: 网络初始化阶段、簇建立阶段、路由更新阶段和数据传输阶段.

2.1 网络初始化阶段

当传感器节点布置到监测区域之后, Sink通过洪泛(Hop configuration message, HCM)报文, 建立一棵最短路径树(以跳数计, 不妨称为跳树). HCM报文是一个三元组 $\langle \text{Type}, \text{ID}, \text{HTS} \rangle$. 其中: Type指明消息类型, ID是转发该报文的节点的标识符, HTS(Hop-to-Sink)表示到Sink的距离.

下述算法1给出了跳树建立算法.

算法1 跳树建立算法.

1) The Sink floods an HCM message with the value: $\text{HTS} = 0$;

2) For each node u that received an HCM message

3) If $\text{HTS}(u) > \text{HTS}(\text{HCM}) + 1$

4) $\text{NH}(u) = \text{ID}(\text{HCM})$; %NH(u) is the next hop of u in the whole network routing structure.

5) $\text{HTS}(u) = \text{HTS}(\text{HCM}) + 1$, $\text{ID}(\text{HCM}) = \text{ID}(u)$, $\text{HTS}(\text{HCM}) = \text{HTS}(u)$;

6) u retransmits the HCM message to its neighbors;

Sink的HTS值为0, 其余节点的初始HTS值为 ∞ . 任何一个接收到HCM报文的节点将比较报文中的HTS值与自己的HTS值. 如果存在一条到达Sink的更短路径, 则节点更新相关信息并转发HCM报文(步骤3)~步骤6); 否则, 丢弃该HCM报文. 该过程一直重复, 直到整个网络的跳树路由结构建立起来.

2.2 簇建立阶段

事件发生后, 事件域内的节点将分布式地形成簇结构. 簇头负责簇内成员的管理与簇内数据的聚合, 它的选取是簇形成过程中最关键的部分, 可采取不同的策略, 如最大节点度、最大剩余能量、最小标识符等. 为便于比较, 本文采用与InFRA相同的簇头选取策略: 具有最小标识符(ID)的事件域节点将成为簇头. 算法2描述了成簇算法.

算法2 成簇算法.

1) For each node u that detected the event

2) u sends a DM to its neighbors and waits for a proper time to receive DMs;

3) If $\text{ID}(u)$ is smaller than any $\text{ID}(v)$ % v is a neighbor of u and detected the same event.

4) $\text{Role}(u) = \text{CH}$, $\text{CH_ID1}(u) = \text{ID}(u)$, $\text{CH_ID2}(u) = \text{NULL}$;

5) Else % w is the neighbor of u with the smallest ID;

6) $\text{Role}(u) = \text{CM}$, $\text{CH_ID1}(u) = \text{NULL}$, $\text{CH_ID2}(u) = w$;

7) If $\text{Role}(u) == \text{CH}$

8) u calculates the delay of CA according to the formula (1) and sets a timer;

9) u sends a CA out within the event scope while timeout happens;

10) While u receives a CA

11) If $\text{CH_ID1}(u) == \text{NULL}$

12) If $\text{CH_ID2}(u) < \text{CH_ID}(\text{CA})$

13) u discards the CA;

14) Else

- 15) CH_ID1(u) = CH_ID(CA);
- 16) NH_C(u) = S_ID(CA);
- 17) S_ID(CA) = u ;
- 18) u retransmits the CA;
- 19) Else;
- 20) If CH_ID1(u) > CH_ID(CA)
- 21) Do the same operations as shown in lines 15~18;
- 22) Else
- 23) u discards the CA;

本阶段需利用DM (Detecting message) 和CA (Cluster-head announcement) 报文. DM 报文是一个三元组 (Type, ID, E_ID), ID 为节点标识符, E_ID 为事件标识符. 节点使用该报文确定邻居节点的事件监测情况. CA 报文也是一个三元组 (Type, CH_ID, S_ID), CH_ID 为簇头标识符, S_ID 为转发该CA报文的节点的标识符, 用于事件域节点竞争簇头. 相应地, 每个节点维持4个域: Role, CH_ID1, CH_ID2, NH_C. Role 为节点的角色 (簇头CH、簇内成员CM), CH_ID1 为节点所确定跟随的簇头, CH_ID2 为节点所确定跟随的暂时簇头, NH_C 记录节点在簇内的下一跳节点.

节点监测到事件后向其邻居发送DM 报文, 并同监测到该事件的邻居进行比较: 若它的ID 最小, 则它将成为候选簇头; 否则, 它只能成为簇成员, 并选择具有最小ID 的邻居作为其暂时簇头 (步骤3)~步骤6). 接下来, Role 为CH 的节点根据式(1) 确定发送CA报文的延迟, 即

$$t = T \times \frac{X}{X_{\max}}. \quad (1)$$

其中: T 是簇头选择与簇内路由形成的持续时间, X 是节点的ID, X_{\max} 是所有节点的最大ID (注意: X 与 X_{\max} 也可为其他的性能参数, 由具体的簇头选举策略来决定).

通过CA报文的转发, ID 最小的节点将成为簇头, 同时确定相应的簇内路由结构 (步骤10~步骤23). 引入时间延迟的目的是使ID 号较小的候选簇头能较早地发出CA 报文, 减少ID 号较大的候选簇头的CA 报文转发次数, 节省控制开销.

2.3 路由更新阶段

簇形成之后与事件结束之时, 簇头均需将自己的位置信息沿着已有的跳树路由结构通知给Sink. Sink 依据下式计算网络的路由汇聚中心, 并将其在全网广播:

$$\mathbf{x}_{\text{RAC}} = \frac{\mathbf{x}_{\text{Sink}} + \sum_{i \in \text{CHSet}} \mathbf{x}_i}{1 + |\text{CHSet}|}. \quad (2)$$

其中: CHSet 为簇头节点集合, $\mathbf{x}_{i \in \text{CHSet}}$ 、 \mathbf{x}_{RAC} 、 \mathbf{x}_{Sink} 分别为簇头、路由汇聚中心和Sink 的位置.

网内节点获得该路由汇聚中心的位置信息后, 依据下述路由更新策略更新下一跳路由.

算法3 路由更新算法.

- 1) If an event occurs or finishes
- 2) The cluster head of the event will report the case to the Sink;
- 3) The Sink calculates the routing aggregation center (RAC) according to the formula (2) and broadcasts it to the whole network;
- 4) Each node u finds a neighbor v who satisfies:
 - ① the HTS level is lower than that of u ,
 - ② the distance to the RAC is the smallest;
- 5) NH(u) = v ;

2.4 数据传输阶段

数据传输分为两部分: 簇内与簇间. 簇内数据传输由NH_C 域指定; 而从簇头开始, 数据传输由NH 域指定. 由于簇头为事件域内ID 最小的节点, 簇头的NH 所指定的下一跳可能是其簇成员节点 (即回传), 这将导致一定的资源浪费. 为了避免这种不必要的传输, 本文采用与InFRA 类似的角色迁移策略. 出现这种情况时, 相关节点将其NH_C 的值设置为其NH 的值, 并相应修改角色. 这样数据既能在簇内正常路由, 又能被正确路由出簇, 而且避免了数据回传带来的资源浪费.

3 理论分析与仿真实验

3.1 理论分析

由于ARA 算法是对InFRA 算法的改进, 本文将分析比较ARA 算法与InFRA 算法.

1) ARA 算法的事件域节点成簇机制 (ARA_C) 比InFRA 算法的事件域节点成簇机制 (InFRA_C) 更节省控制开销.

InFRA_C 分为两阶段: ① 节点与邻居节点比较, 获得簇头竞争权; ② 参与簇头竞争的节点将竞争信息 (CA) 洪泛至簇内所有节点. ARA_C 也分为两阶段: 第1 阶段与InFRA_C 相同, 第2 阶段不同. 假设事件域内节点的个数为 N_C , 在第1 阶段操作完成后, 事件域内的候选簇头个数为 N_{CH} , 簇内节点的平均簇内邻居数为 N_N . InFRA_C 中CA 的发送次数为 $N_C N_{CH}$, 接收次数为 $N_C N_N N_{CH}$. 对于ARA_C, CA 的发送延迟机制与转发限制使得最优候选簇头的CA 能够洪泛至整个事件域, 而非最优候选簇头的CA 的洪泛范围尽可能地缩小. 当候选簇头按照ID 从大到小的顺序在事件域内发送CA 时, 出现最差情况, 此

时CA的发送次数上限为 $N_C N_{CH}$,接收次数上限为 $N_C N_N N_{CH}$.当最优候选簇头最早发送CA而其他竞争者均保持沉默时,CA的发送与接收次数最少,分别为 N_C 和 $N_C N_N$.因此ARA_C的CA平均发送与接收次数分别为 $(N_{CH} + 1)N_C/2$ 和 $(N_{CH} + 1)N_C N_N/2$.

由上可见,ARA_C比InFRA_C更节省控制开销.

2) ARA算法的路由更新机制(ARA_R)比InFRA算法的路由更新机制(InFRA_R)更节能.

事件发生后,InFRA_R需要节点获知其到所有簇头的距离之和.节点到簇头的距离是通过簇头向全网洪泛自身信息,建立一棵以簇头为根的最短路径树来获得.假设网络节点数为 N ,距离Sink最远的节点的跳数为 D ,事件个数为 N_E ,网络范围为 A ,节点通信半径为 r ,则节点的平均邻居数为 $\pi r^2 N/A$.InFRA_R的控制报文至少发送 NN_E 个,接收 $\pi r^2 N^2 N_E/A$ 个.而ARA_R在事件发生后,簇头仅将自身信息沿着已有路由结构报告给Sink,其控制报文传输次数最多为 DN_E (少于 NN_E),接收次数最多为 $(D - 1)N_E$.Sink广播路由汇聚中心信息,每个节点接收一次信息,共 NN_E 次.因此,接收控制报文的总次数最多为 $(N + D - 1)N_E$.为保证网络连通性,应有 $\pi r^2 N/A > 8$,而 $N + D - 1 < N + N < 2N$,因此 $(N + D - 1)N_E < \pi r^2 N^2 N_E/A$.

综上所述,ARA_R比InFRA_R更节能.

3) ARA算法生成的路由结构的数据聚合性能与InFRA算法生成的路由结构性能相似,当节点密度较大时,ARA算法的性能更好.

InFRA与ARA在路由更新阶段规定每个节点选取距离Sink最近的节点作为下一跳.当存在多个选择时,InFRA选取与所有簇头的跳数距离和最小的节点作为下一跳,而ARA选取距离路由汇聚中心最近的节点作为下一跳.这两种方法均使下一跳的选择既能保证最短路径又能增大路径重叠程度.因此ARA生成的路由结构的数据聚合性能与InFRA生成的路由结构的性能相似.当节点密度较大时,与所有簇头的跳数距离和最小的节点的个数增加较多,而距路由汇聚中心最近的节点的个数增加较少,因此,InFRA的下一跳更新随机化程度较高,表明ARA算法的性能更好一些.

此外,Nakamura等^[15]证明了InFRA能够获得一棵具有较好近似比的近似Steiner树.通过以上分析易知,ARA算法也能够获得一棵基于事件域的较优的近似Steiner树.

3.2 仿真实验

为了说明本文算法的有效性,选择InFRA、SPT

与CNS进行实验比较.采用与文献[15]相同的能量消耗模型,并假设事件域形状为圆形,事件发生的位置、时间随机.默认的仿真参数如表1所示.

表1 默认仿真参数

参数	数值
网络范围/m ²	1000×1000
节点数量	5184
节点通信半径r/m	35
数据分组大小/bits	4000
控制分组大小/bits	200
事件通知间隔/s	40
事件域半径Revent/m	60
事件持续时间/h	2~4,均匀分布
并发事件数	2
不活跃时间/h	0.5
网络运行时间/h	12

仿真实验对比了以下网络性能参数.

1) 数据分组数(data): 整个网络中数据分组的传输次数.

2) 控制开销(overhead): 为了维持网络的正常运行而发送的控制分组的个数.

3) 路由效率(efficiency): 源节点产生的数据分组数与传输的数据分组总数的比值.

4) 能量消耗(energy): 网络消耗的总能量.

3.2.1 成簇开销

变化事件域半径,分别统计ARA_C与InFRA_C的CA发送和接收次数(见图1).显然ARA_C的CA发送与接收次数少于InFRA_C,而且随着事件域半径的增大,InFRA_C的CA发送与接收次数的增加更为剧烈.实验结果与理论分析结果一致.

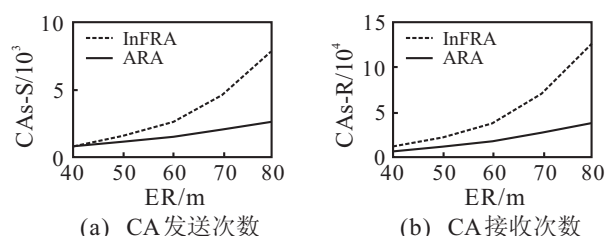


图1 成簇开销比较

3.2.2 事件范围变化

事件域范围变化时,各算法性能如图2所示.图2(a)表明,ARA的数据分组传输量最小,其次为InFRA.原因如下:ARA与InFRA采用簇机制来聚合事件域内数据,并在路由形成过程中利用启发式方法增大路径的重叠程度,促使数据聚合尽早发生.此外,由于节点密度较大,ARA的可选下一跳节点数比InFRA的少,它所获得的路径重叠度更高,数据聚合

性能更优. 由图 2(b) 可知, InFRA 的控制开销最大, 且其增长幅度大于 ARA. 这是由于 InFRA 需要全网洪泛来更新所有节点到各个簇头的距离和, 而且随着事件域半径的增大, InFRA_C 的开销增长幅度比 ARA_C 大. 由于采用了完全聚合, 各算法的路由效率随着事件域半径的增大而增大. ARA 的数据传输分组数最少, 因此它的路由效率最高, 这与图 2(c) 相符. 图 2(d) 表明, ARA 的总能量消耗最少, 原因显而易见.

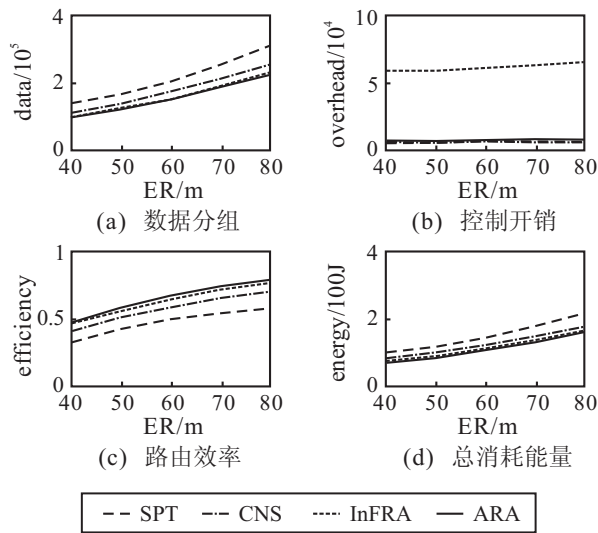


图 2 事件半径变化

3.2.3 通信半径变化

通信半径的增大将减少节点到 Sink、簇内节点到簇头的跳数距离. 因此通信半径增大时, 各算法的数据和控制分组传输量减少, 如图 3(a) 和图 3(b) 所示. 图 3 表明, 通信半径变化时, ARA 算法的数据分组传输量最少、总控制开销较小、路由效率最高、总能耗最小. 这是因为 ARA 能以较小的路由构建和维护开销获得具有较优数据聚合性能的路由树.

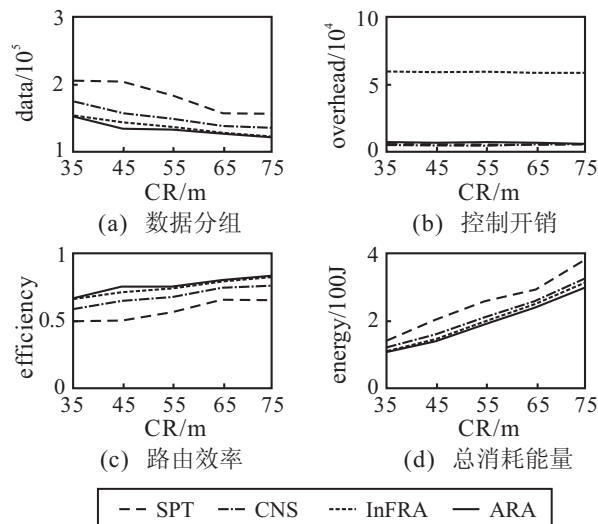


图 3 通信半径变化

3.2.4 网络范围变化

保持节点密度为 20, 变化网络范围 (相应节点数从 1600 变化到 7744), 实验结果如图 4 所示. 网络范围的增大导致参与数据传输的节点数增多, 因此各算法的数据分组传输量增大、路由效率降低、网络能耗增大、控制开销增大, 其中 InFRA 的控制开销增长剧烈. 由于 ARA 能使数据尽早聚合且控制开销较小, 其性能在网络范围变化时仍为最优.

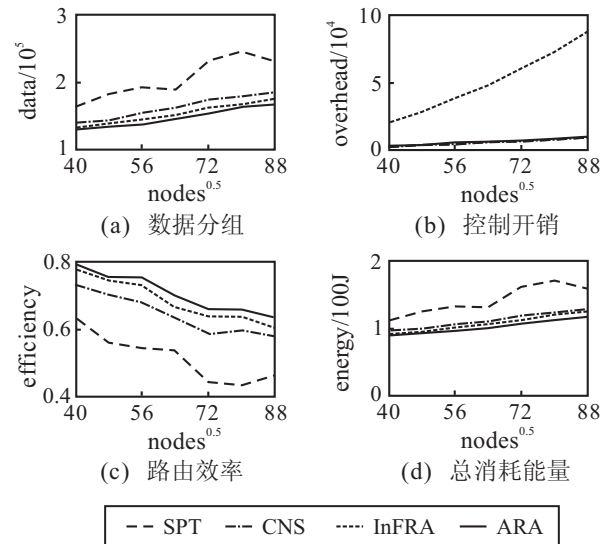


图 4 网络范围变化

3.2.5 网络密度变化

保持网络范围 (1000 m × 1000 m) 与通信半径 (35 m) 不变, 变化网络节点数, 得到网络密度变化时各个算法的性能情况 (图 5). 当密度增大时, 事件域内的节点将增多, 因此各算法的数据分组传输量、控制开销、路由效率和总能量消耗均增大. 由于 ARA 能够获得较优的数据聚合树且其控制开销较少, ARA 的总体性能仍最优.

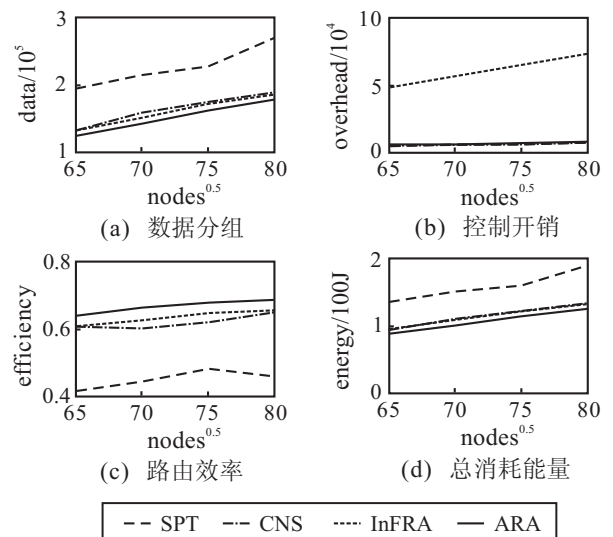


图 5 网络密度变化

4 结 论

本文研究了针对高效率数据聚合的传感器网络路由问题,提出了一个新的事件驱动传感器网络中基于数据聚合的自适应路由算法.该算法设计了事件域节点的分布式成簇方式,利用路由汇聚中心更新下一跳路由,最终建立一棵近似Steiner树.理论分析与仿真实验表明,ARA算法能够实现数据的最短路径传输,加大路径的重叠程度,有效提高簇间数据聚合的机率,降低路由结构建立与维护的控制开销.此外,本算法还具有较好的网络可扩展性.

ARA算法是针对完全数据聚合的路由算法,适合于高相关性的数据采集应用.但在实际中,事件间的数据相关性不同,因此,如何挖掘事件间的数据相关性,设计适应不同数据相关性的针对高效数据收集的路由算法是今后的研究目标之一.

参考文献(References)

- [1] 唐伟.无线传感器网络节能策略及算法研究[D].成都:电子科技大学通信抗干扰技术国家级重点实验室,2011:10-15.
(Tang W. On energy reserving policies and algorithms for wireless sensor networks[D]. Chengdu: National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, 2011: 10-15.)
- [2] Liu Q, Chang Y N, Jia X H. A hybrid method of CSMA/CA and TDMA for real-time data aggregation in wireless sensor networks[J]. *Computer Communications*, 2013, 36(3): 269-278.
- [3] Souza E, Nikolaidis I. An exploration of aggregation convergecast scheduling[J]. *Ad Hoc Networks*, 2013, 11(8): 2391-2407.
- [4] Heinzelman W, Chandrakasan A, Balakrishnan H. An application specific protocol architecture for wireless microsensor networks[J]. *IEEE Trans on Wireless Communications*, 2002, 1(4): 660-670.
- [5] Soro S, Heinzelman W B. Cluster head election techniques for coverage preservation in wireless sensor networks[J]. *Ad Hoc Networks*, 2009, 7(5): 955-972.
- [6] Meng L J, Zhang H Z, Zou Y. A data aggregation transfer protocol based on clustering and data prediction in wireless sensor networks[C]. *The 7th Int Conf on Wireless Communications, Networking and Mobile Computing*. Wuhan: IEEE Press, 2011: 1-5.
- [7] Jung W S, Lim K W, Ko Y B, et al. Efficient clustering-based data aggregation techniques for wireless sensor networks[J]. *Wireless Networks*, 2011, 17(5): 1387-1400.
- [8] Kwon S, Ko J H, Kim J, et al. Dynamic timeout for data aggregation in wireless sensor networks[J]. *Computer Networks*, 2011, 55(3): 650-664.
- [9] Al-Karaki J N, Ul-Mustafa R, Kamal A E. Data aggregation in wireless sensor networks-exact and approximate algorithms[C]. *Workshop on High Performance Switching and Routing*. Phoenix: IEEE Press, 2004: 241-245.
- [10] Hougardy S, Promel H J. A 1.598 approximation algorithm for the steiner problem in graphs[C]. *Proc of the 10th Ann ACM-SIAM Symp on Discrete Algorithms*. Baltimore: Society for Industrial and Applied Mathematics, 1999: 448-453.
- [11] Robins G, Zelikovsky A. Improved steiner tree approximation in graphs[C]. *Proc of the 11th Ann ACM-SIAM Symp on Discrete Algorithms*. Sanfrancisco: Society for Industrial and Applied Mathematics, 2000: 770-779.
- [12] Krishnamachari B, Estrin D, Wicker S B. The impact of data aggregation in wireless sensor networks[C]. *Proc of the 22nd Int Conf on Distributed Computing Systems*. Vienna: IEEE Press, 2002: 575-578.
- [13] Villas L A, Boukerche A, Ramos H S, et al. DRINA: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks[J]. *IEEE Trans on Computers*, 2013, 62(4): 676-689.
- [14] Villas L A, Boukerche A H, Oliveira A B F, et al. A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks[J]. *Ad Hoc Networks*, 2014, 12: 69-85.
- [15] Nakamura E F, Ramos H S, Villas L A, et al. A reactive role assignment for data routing in event-based wireless sensor networks[J]. *Computer Networks*, 2009, 53(12): 1980-1996.

(责任编辑:李君玲)