

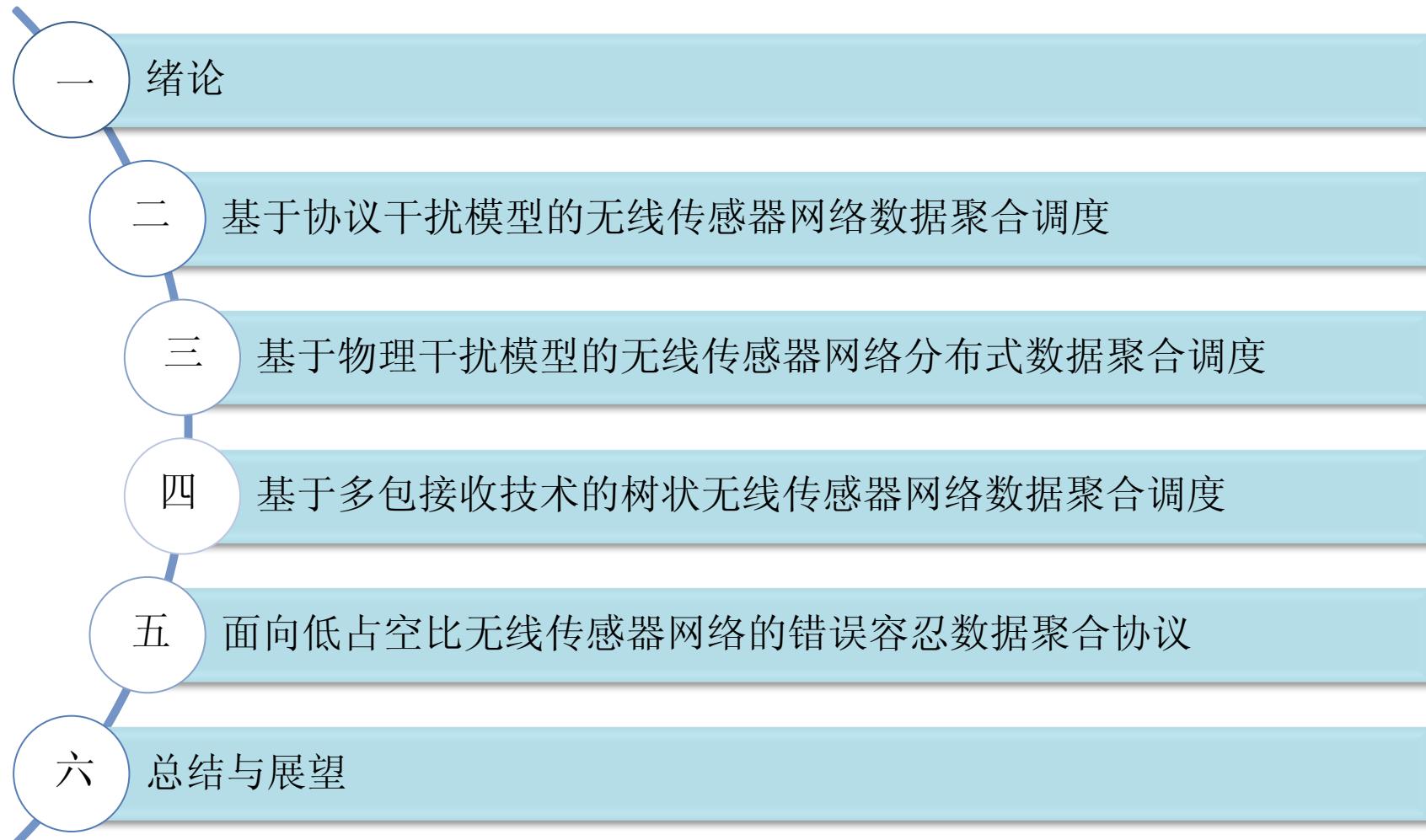


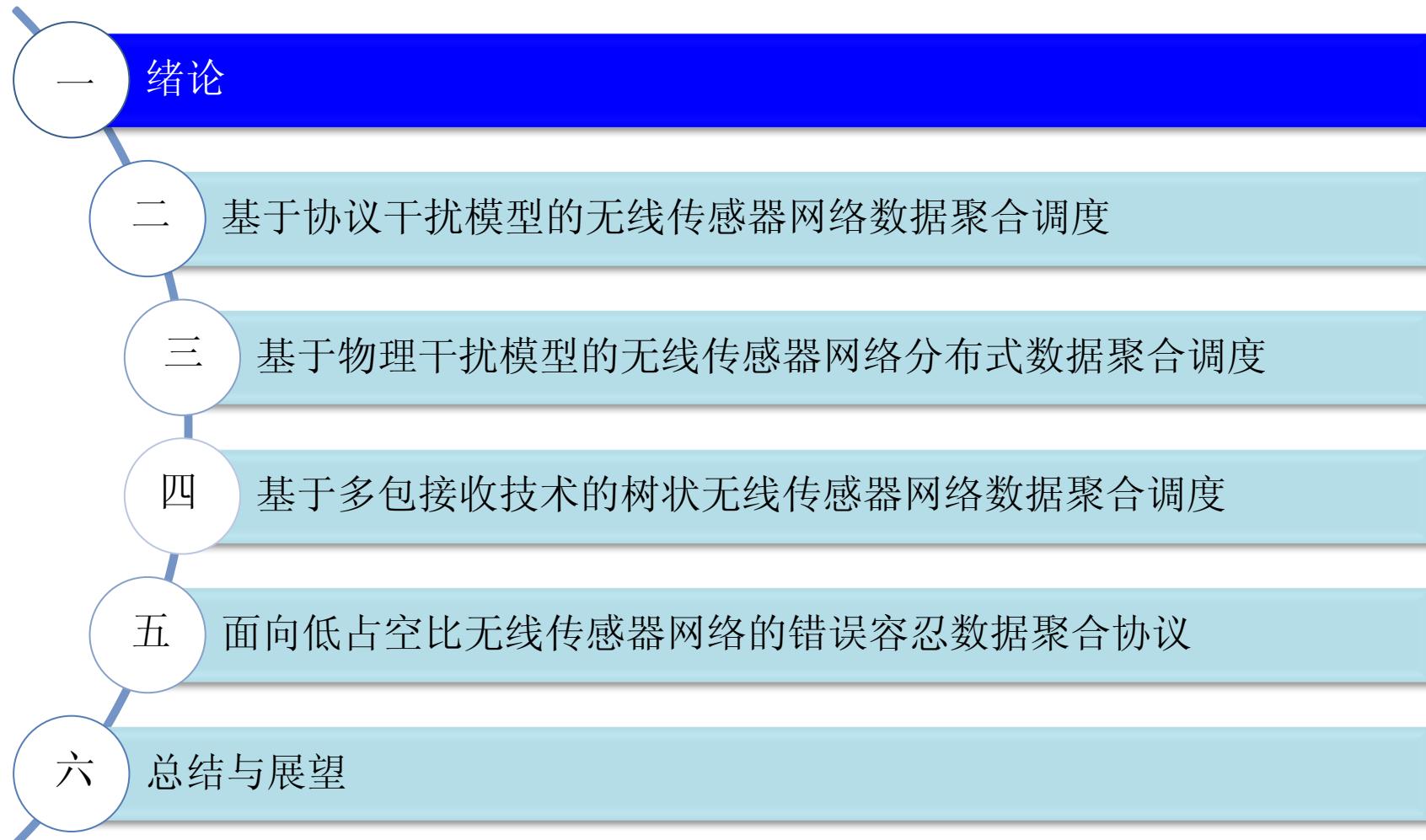
中国科学院上海微系统与信息技术研究所
无线传感网实验室

面向数据聚合的无线传感器网络 传输调度方法研究

指导老师：袁晓兵 研究员
李宝清 研究员

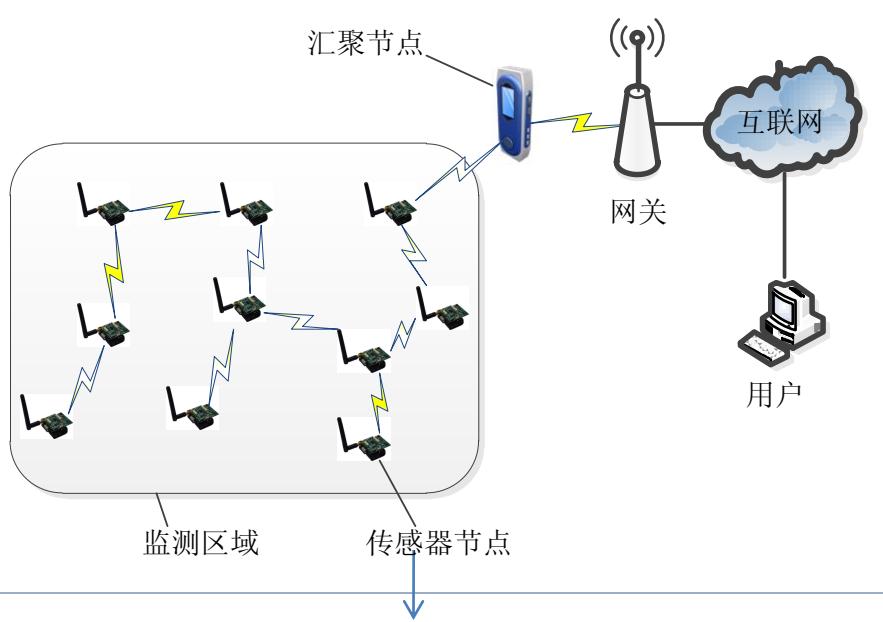
肖世良 博士生
2015-05



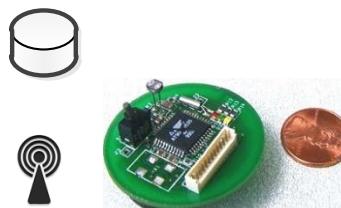
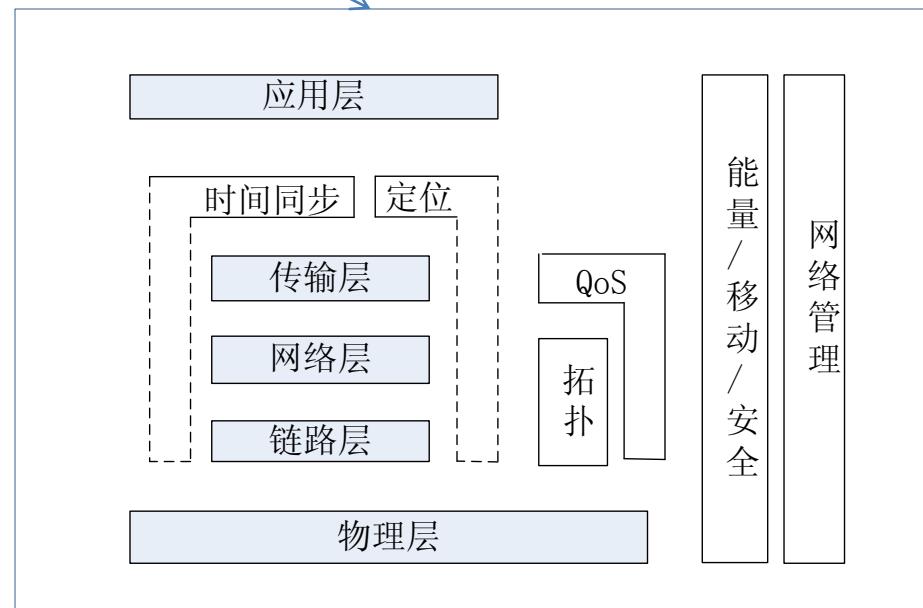


无线传感器网络(Wireless Sensor Networks, WSNs): 由大量布设在监测区域中的具有感知、计算、存储和通信功能的节点(借助一整套无线通信协议)自组织而成的网络。

WSNs体系架构



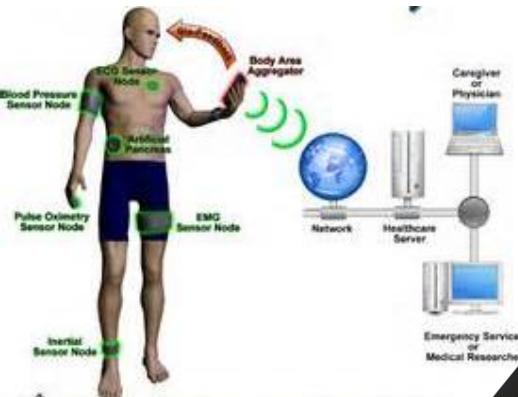
WSNs通信协议栈及其辅助技术



低功耗、多跳、应用相关的无线网络



环境监测



健康医疗



智能家居



安全&国防



工业自动化



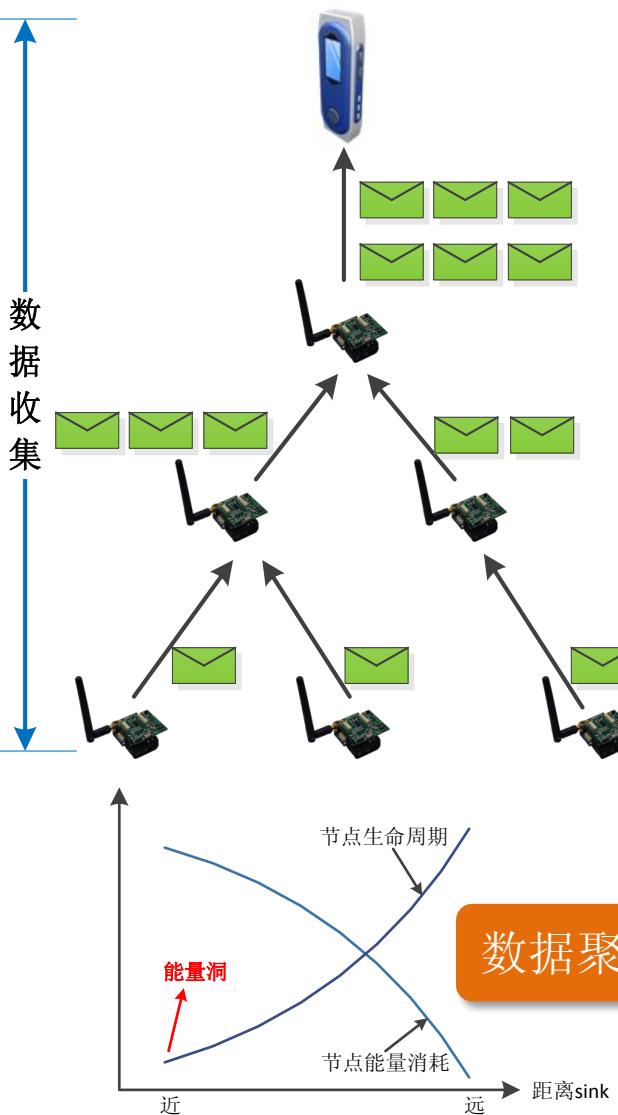
移动感知系统



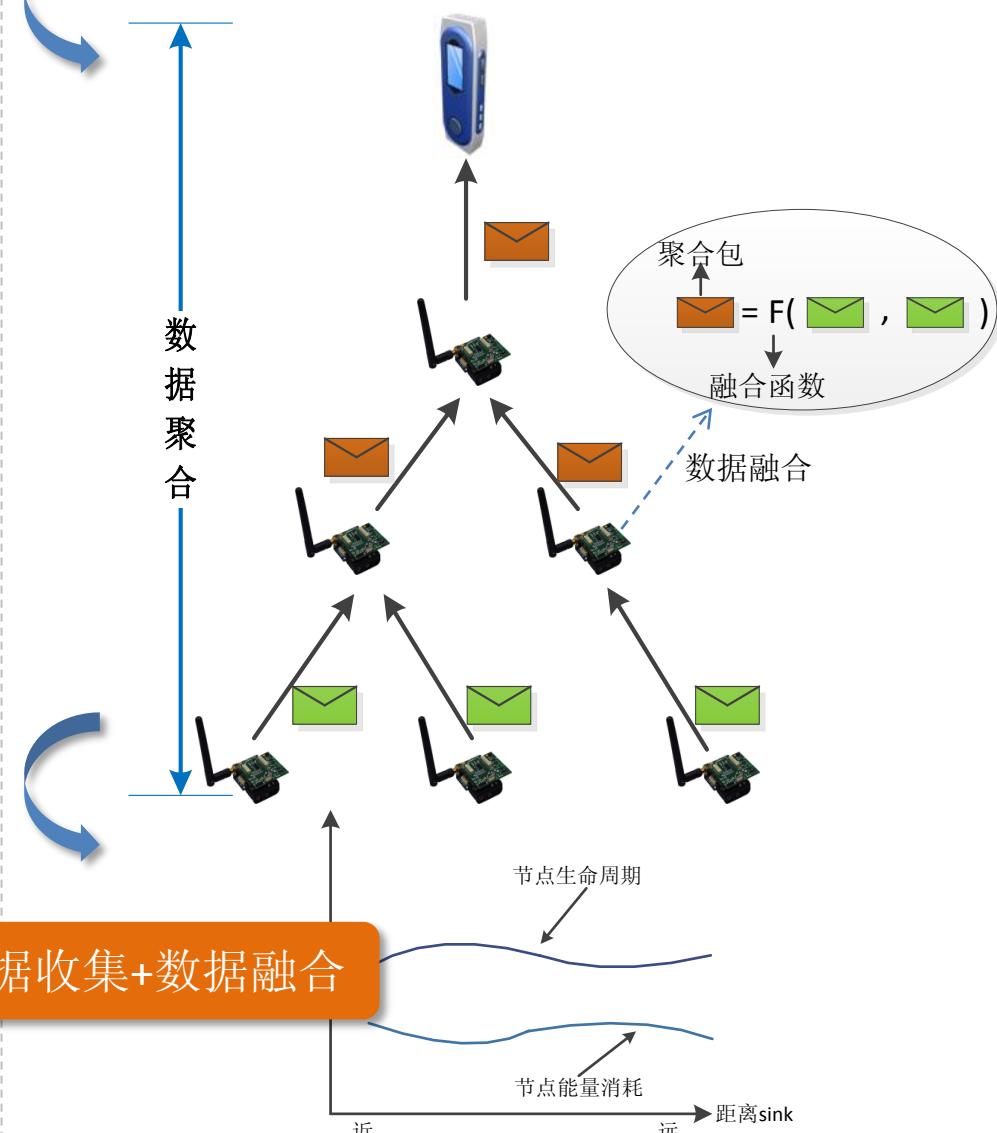
智能电网

无线传感器网络数据聚合

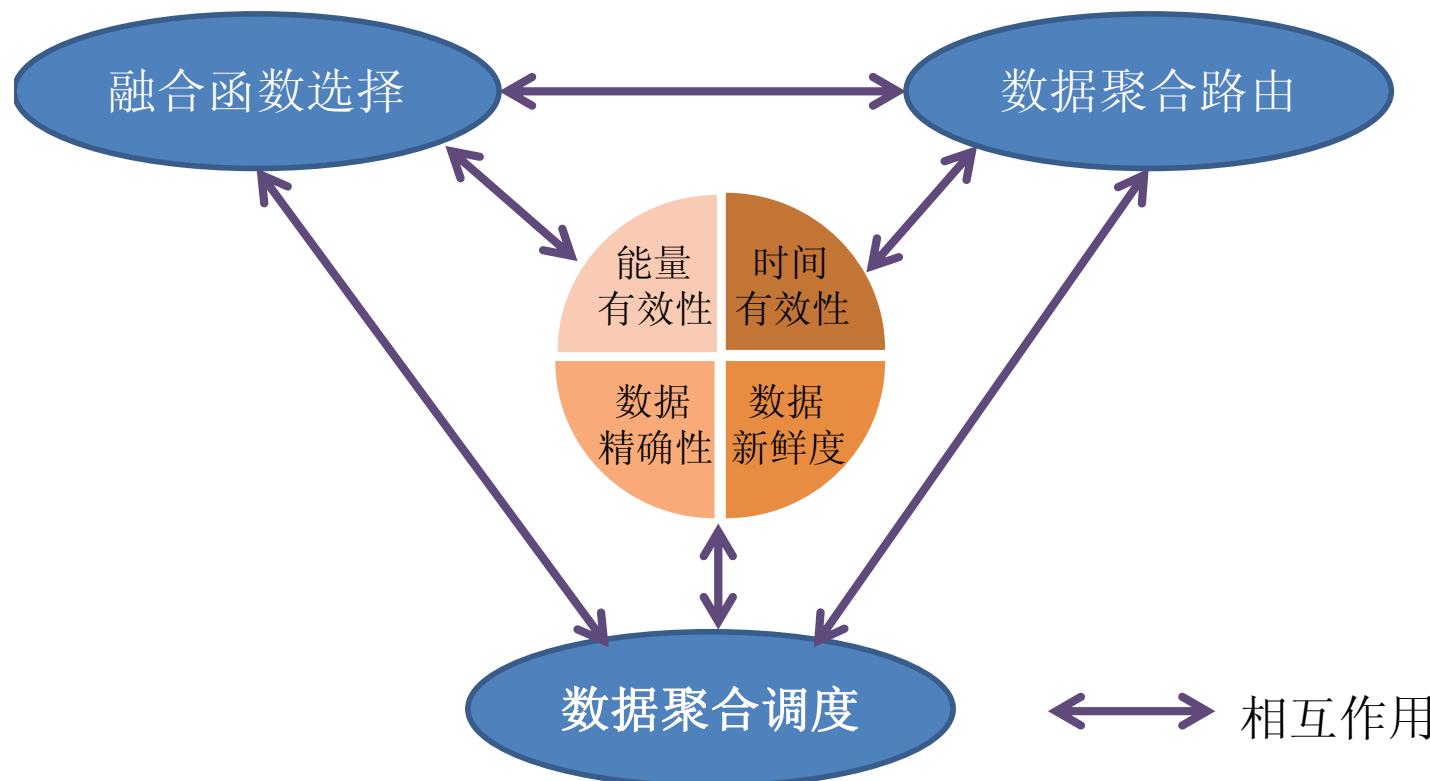
无线传感器网络的基本功能是从物理世界收集数据



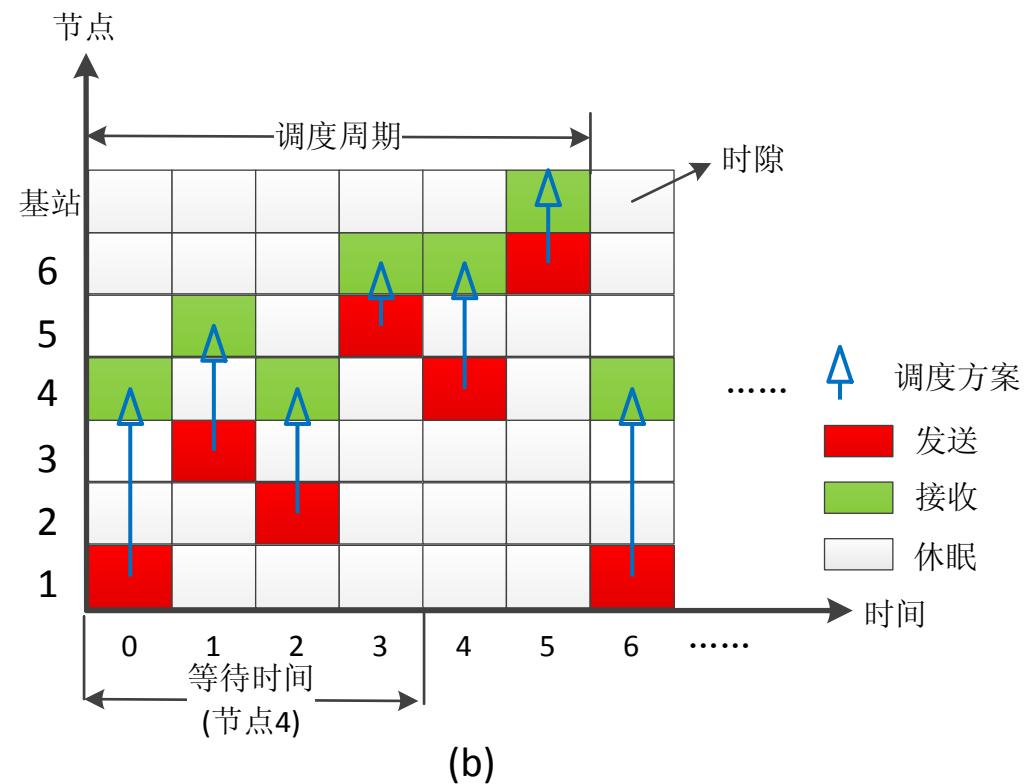
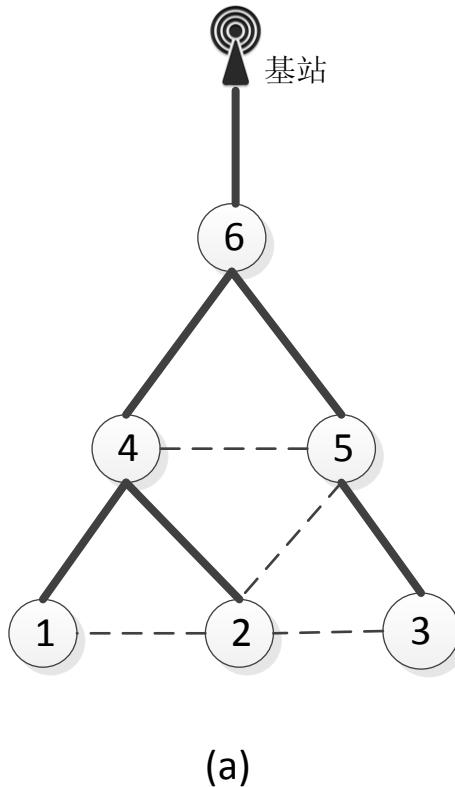
利用数据之间相关性，对来自不同传感器的数据包进行融合



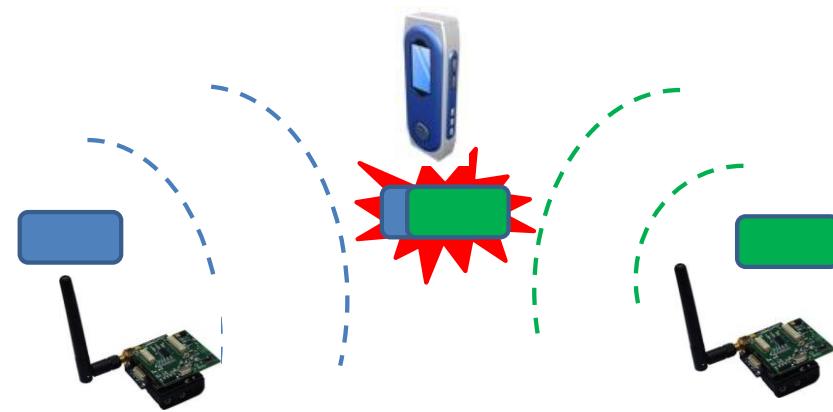
数据聚合=数据收集+数据融合



数据聚合调度：基本概念



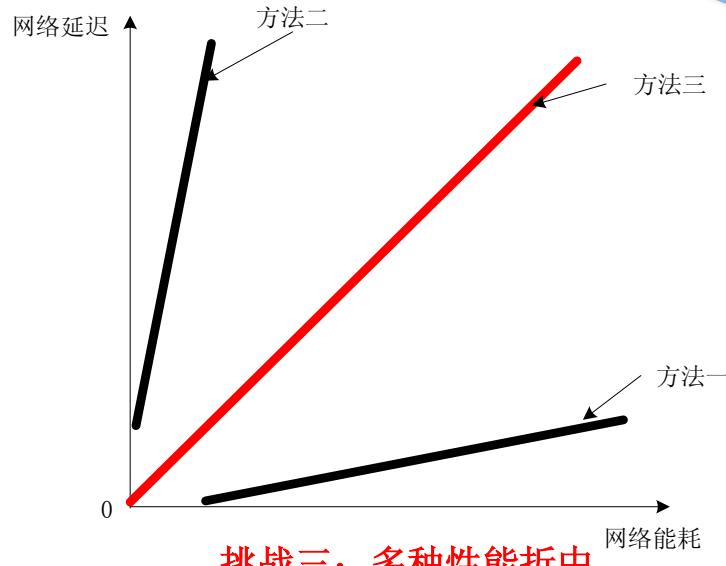
狭义聚合调度 vs 广义聚合调度



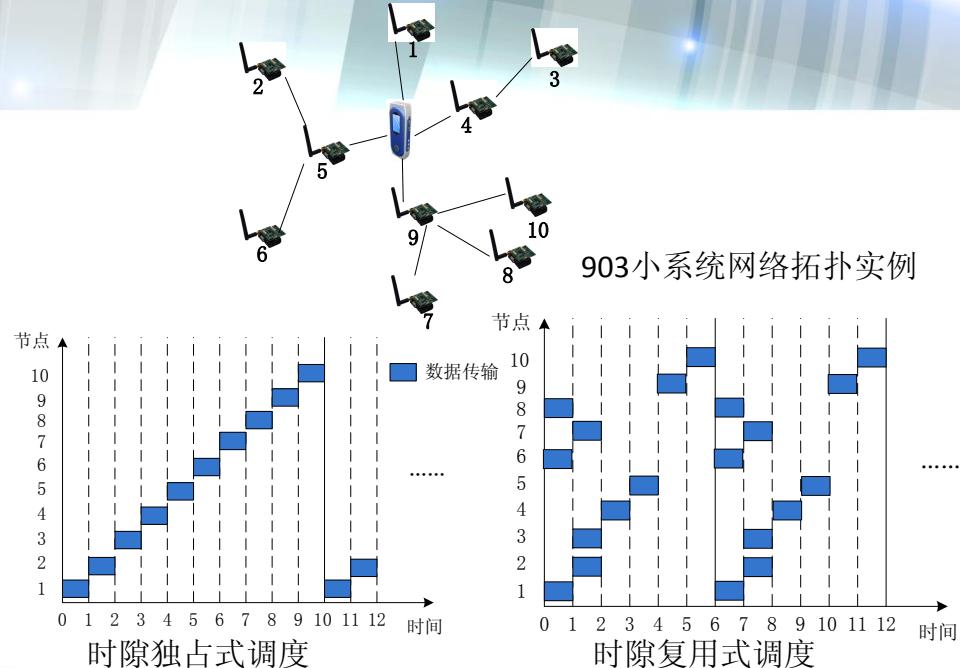
挑战一：克服干扰影响

数据聚合调度研究挑战

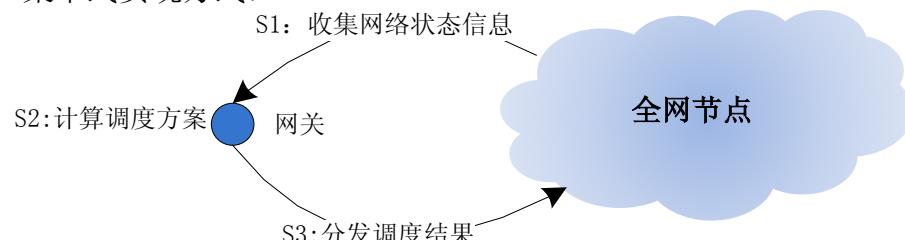
挑战二：提高时间效率



挑战三：多种性能折中



集中式实现方式:



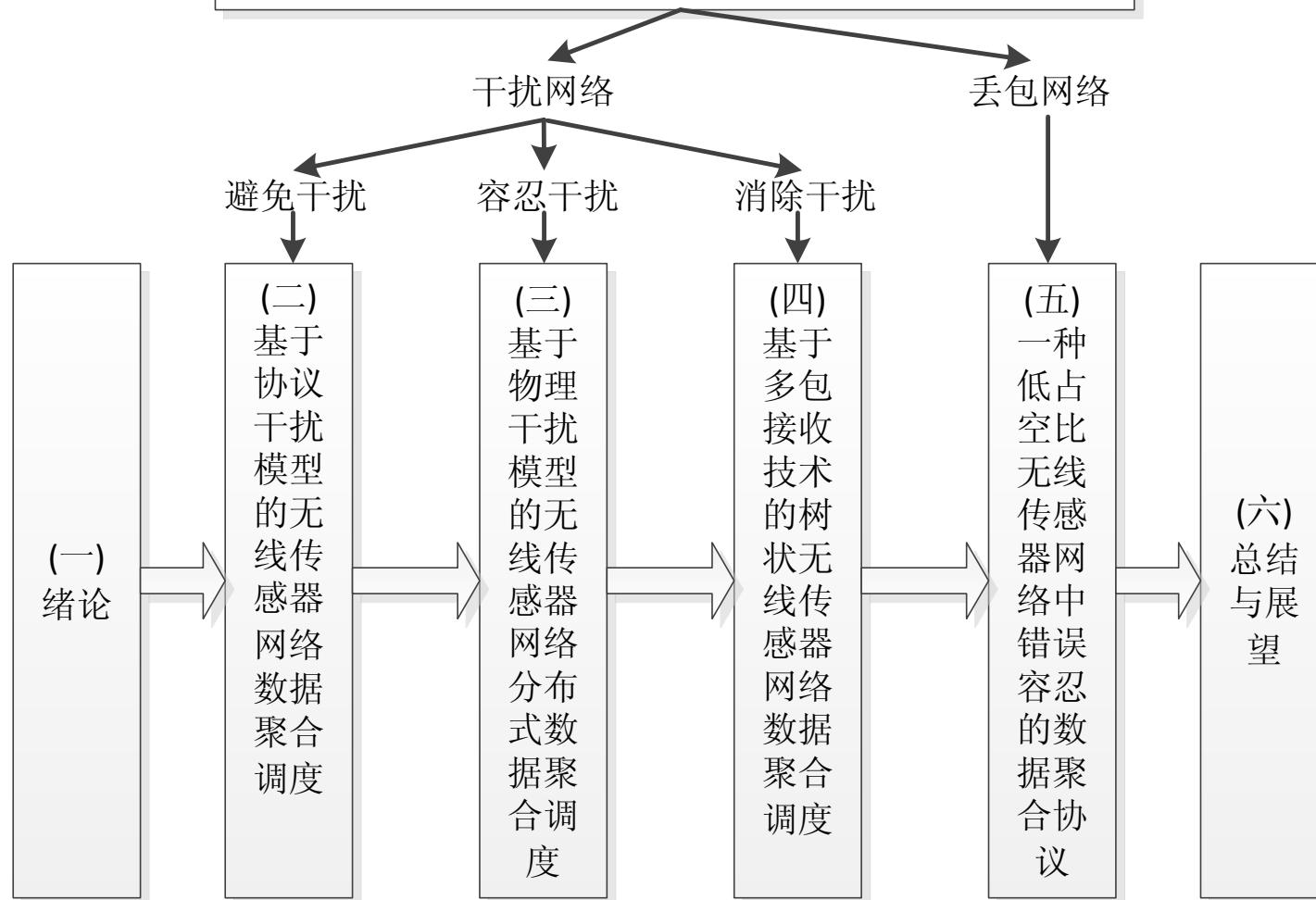
分布式实现方式:

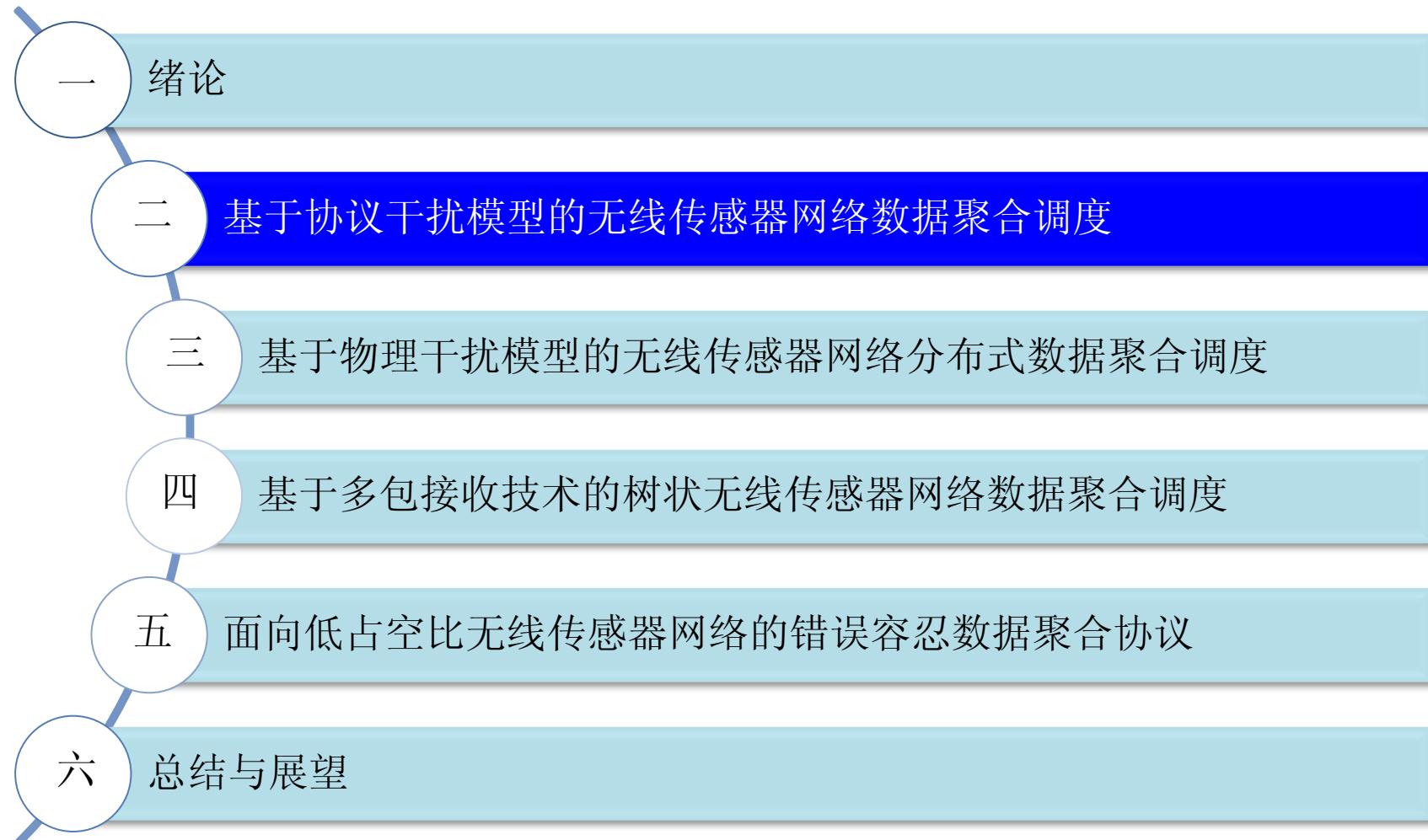


挑战四：可分布式实现

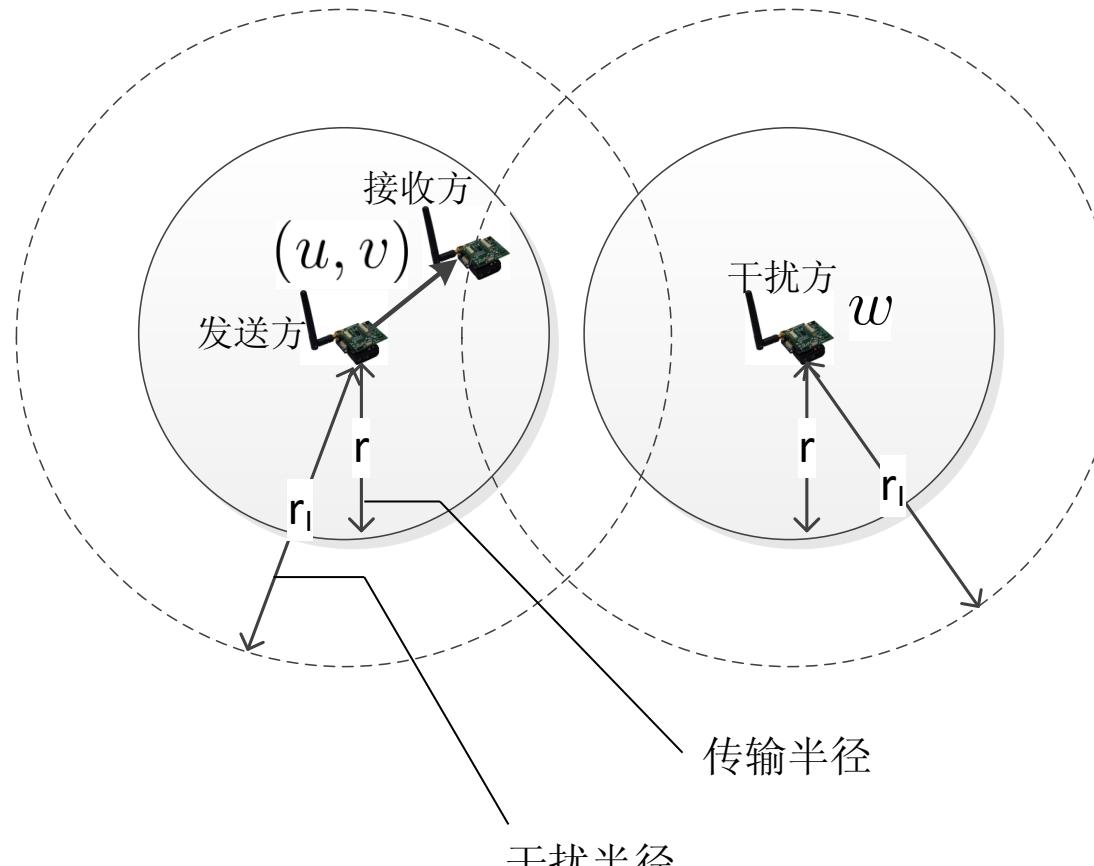
本博士论文针对复杂场景，如多用户干扰、链路丢包等情形下的无线传感器网络数据聚合调度问题展开深入研究，目的是设计可行和高效的调度算法与技术，以提升网络时延、数据质量等性能指标，从而更好满足用户的需求。

面向数据聚合的无线传感器网络传输调度 方法研究





- 时延(latency)是重要的网络性能指标
- 最小化(minimizing)无线传感器网络数据聚合时延是一个重要研究方向
 - 理论意义
 - 实际意义
- 设计可行(feasible)和高效(efficient)的聚合调度(scheduling)算法/协议以达到或者逼近这一理论最小时延
- 干扰的建模与处理是聚合调度算法设计的基础
- 文献中广泛采用的模型有：主干扰模型、单位圆盘图干扰模型等
- 本章考虑更为一般化的模型：协议干扰模型



链路 (u, v) 成功传输 \longleftrightarrow 链路 (u, v) 满足 $d(u, v) \leq r$ 且 $d(w, v) > r_I, \forall w \in \mathcal{S}_u \setminus \{u\}$

- 干扰关系是局部的(local)和二元的(binary)
- 方便采用图论工具来进行调度/网络算法的设计

问题描述：最小时延数据聚合调度(MLDAS)

MLDAS: Minimum Latency Data Aggregation Scheduling

假设(Assumptions):

- 2D平面/半双工的收发器/单信道
- 时间同步/时隙/超帧
- 协议干扰模型

目标(Objective): 最小化数据聚合时延(超帧长度)

满足(subject to):

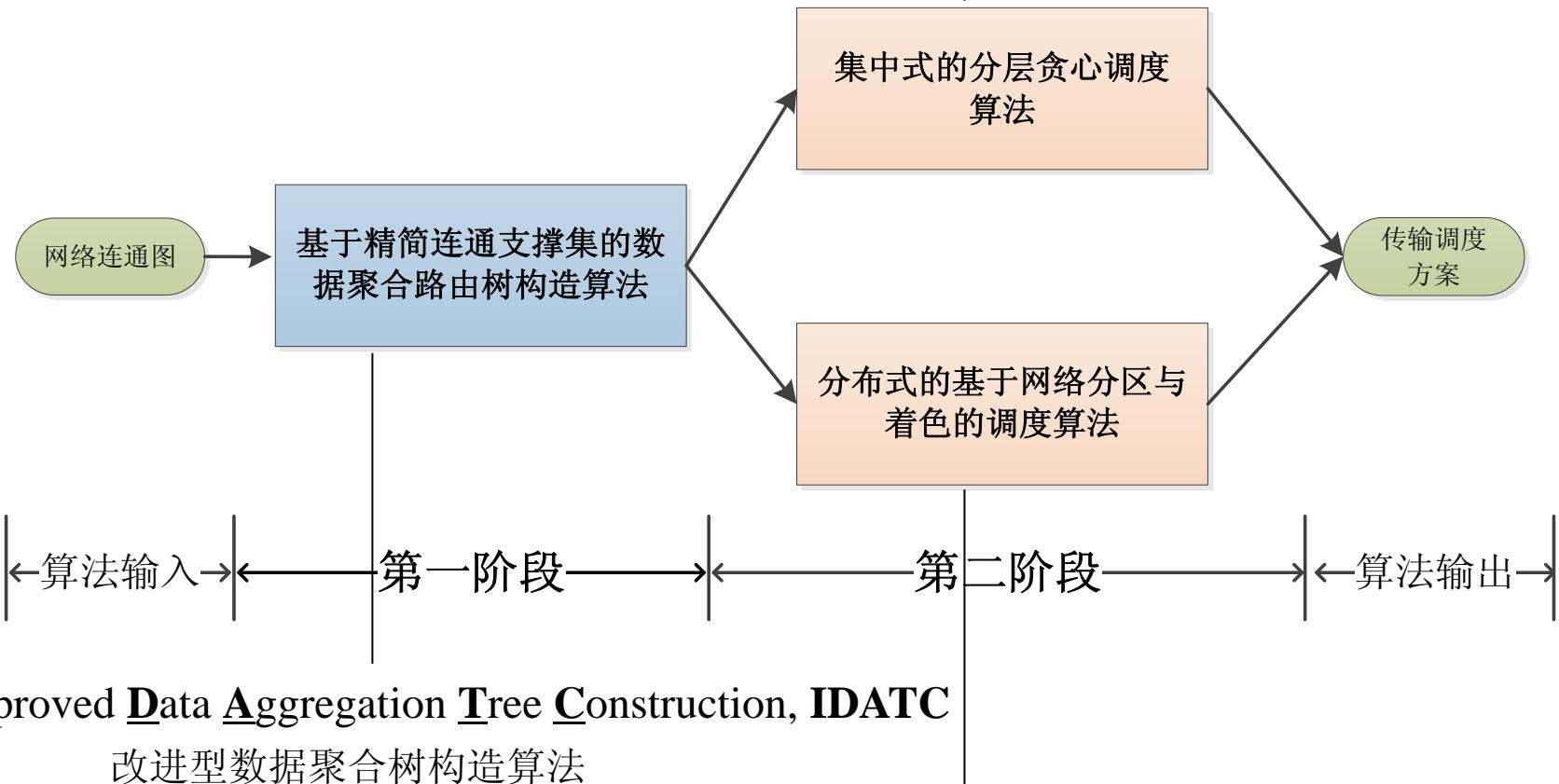
- 通信范围限制
- 半双工限制
- 干扰限制(协议干扰模型)
- 数据聚集限制
- 因果性限制

➤ 问题复杂度: NP-hard

➤ 时延下界(lower bound): $\max(R, \Delta)$, 其中 R 为网络跳数半径, Δ 为最大节点度(node degree)

分层贪心聚合调度算法

Layered Greedy Aggregation Scheduling, **LGAS**



Network Partition & Coloring based Scheduling, **NPCS**
基于网络分区与着色的调度算法



基本思想：有策略性地将图中节点划分为骨干节点和普通节点，分别作为生成树的内部节点和外部(叶)节点，实现“内疏外密”。

Algorithm :改进型数据聚合树构造算法(IDATC)

输入: 图 $G(\mathcal{V}, \mathcal{E})$

输出: 树 $T(\mathcal{V}_T, \mathcal{E}_T)$

```

1: 在图 $G$ 中, 以 $v_s$ 为根节点, 执行宽度优先搜索(BFS), 构造一颗BFS树 $T_{bfs}$ ;
2: 以 $v_s$ 为出发点, 在 $G$ 中一层一层地构造极大独立集(MIS)  $\mathcal{U}$ ;
3:  $\mathcal{V}_T \leftarrow \mathcal{V}$ ,  $\mathcal{E}_T \leftarrow \emptyset$ ;
4: for  $i$ 从1到 $R$  do
5:   for  $\mathcal{U}_i$ 中的每一个支配节点 $u$  do
6:     将节点 $p(u)$ 加入集合 $\mathcal{W}$ , 并将其标记为灰色(GREY);
7:     将 $(u, p(u))$ 和 $(p(u), d(p(u)))$ 加入 $\mathcal{E}_T$ ;
8:   end for
9: end for
10: for  $\mathcal{W}$ 中的每一个连接节点 $w$  do
11:   if IsRedundantConnector( $w$ ) then
12:     将 $w$ 标记为白色(WHITE), 并将边 $(w, d(w))$ 从 $\mathcal{E}_T$ 中移除;
13:     for  $w$ 在树 $T_{bfs}$ 中的每一个孩子支配节点 $c(w)$  do
14:       将 $(c(w), w)$ 从 $\mathcal{E}_T$ 中移除, 并将 $(c(w), w^*)$ 和 $(w^*, d(w))$ 添加到 $\mathcal{E}_T$ ;
15:     end for
16:   end if
17: end for
18: for  $\mathcal{V}$ 中的每一个被支配节点 $v$  do
19:   将 $(v, d(v))$ 中加入 $\mathcal{E}_T$ ;
20: end for
21: return  $T(\mathcal{V}_T, \mathcal{E}_T)$ 

```

Step 4) 形成数据聚合树

**Step 1) 构造独立节点集合
(黑色节点)**

**Step 2) 构造连接节点集合
(灰色节点)**

Step 3) 移除冗余连接节点

Algorithm :冗余连接节点判定算法(IsRedundantConnector)

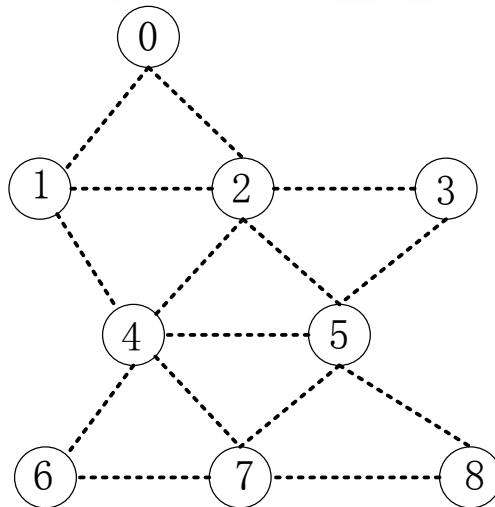
输入: 图 $G(\mathcal{V}, \mathcal{E})$, BFS树 T_{bfs} , 连接节点 w

输出: 如果 w 为冗余连接节点, 则返回TRUE; 否则, 返回FALSE

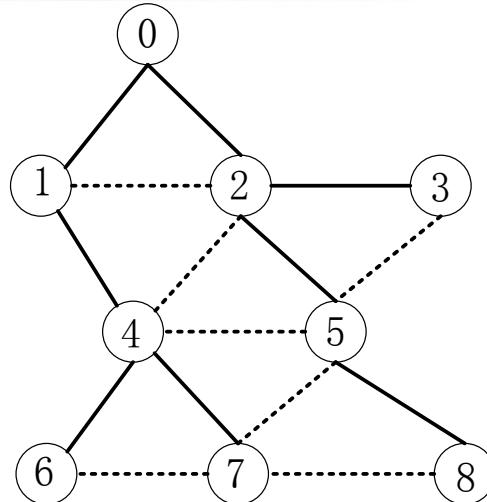
```

1: for  $w$ 在树 $T_{bfs}$ 中的每一个孩子支配节点 $c(w)$  do
2:   if  $(\mathcal{W}_{c(w)} \setminus \{w\}) \cap (\mathcal{W}_{d(w)} \setminus \{w\}) = \emptyset$  then
3:     return FALSE (注: 这里 $\mathcal{W}_i$ 表示节点 $i$ 的相邻连接节点集合);
4:   end if
5: end for
6: return TRUE

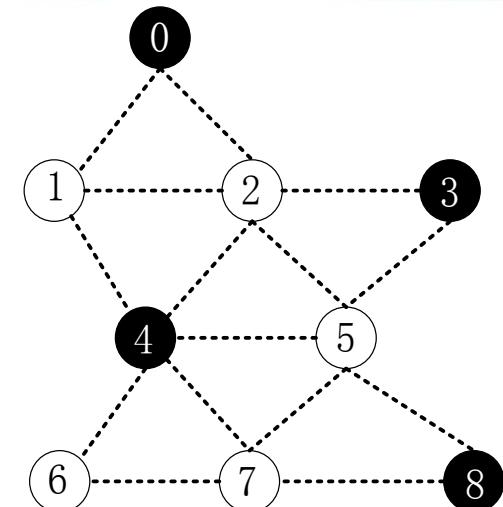
```



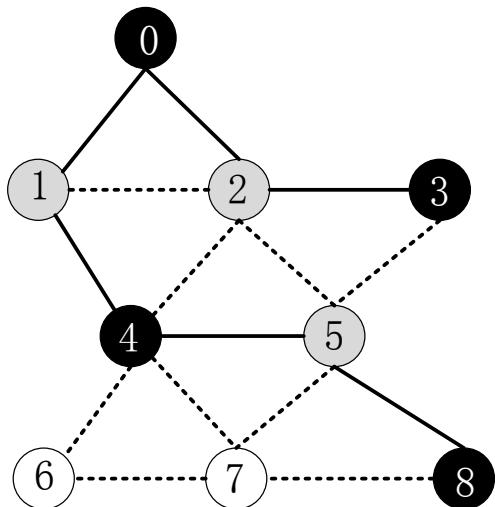
(a) 网络连通图



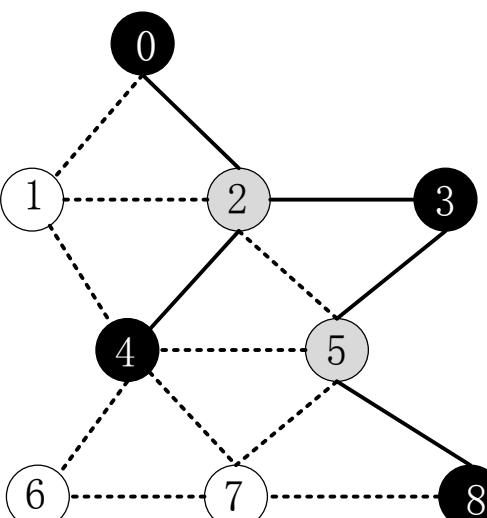
(b) BFS树



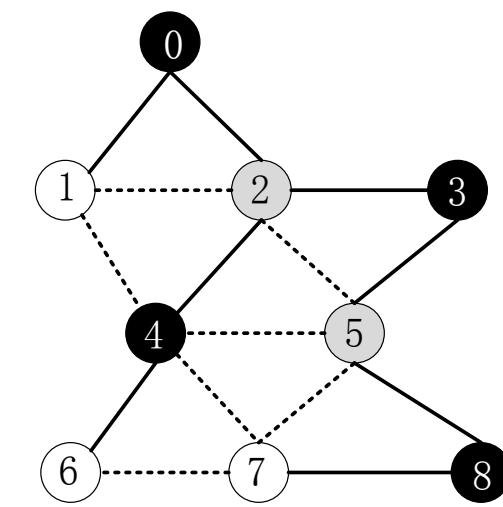
(c) 极大独立集



(d) 连通支撑集



(e) 移除冗余连接节点



(f) 数据聚合树

● 支配节点

● 连接节点

○ 被支配节点

基本思想：以分层地方式，先调度白色节点，然后自底向上交替调度黑色节点和灰色节点；每个过程均采用贪心调度策略，在不破坏各种约束条件下，构造极大的(Maximal) 可行(feasible)传输链路集合

Algorithm :分层贪心聚合调度算法(**LGAS**)

输入：数据聚合树 $T(\mathcal{V}_T, \mathcal{E}_T)$, 图 $G(\mathcal{V}, \mathcal{E})$

- 1: 调用**GSSP**($\mathcal{V} \setminus (\mathcal{U} \cup \mathcal{W})$, G , T);
- 2: 将 $\mathcal{V} \setminus (\mathcal{U} \cup \mathcal{W})$ 中的所有节点从 T 中移除, 得到骨架树 T_B ;
- 3: **for** i 从 d_B 到1 **do**
- 4: 调用**GSSP**(B_i, G, T_B);
- 5: **end for**

▷ 贪心调度子算法(**GSSP**)

输入：发送节点集合 \mathcal{A} , 图 G , 树 T

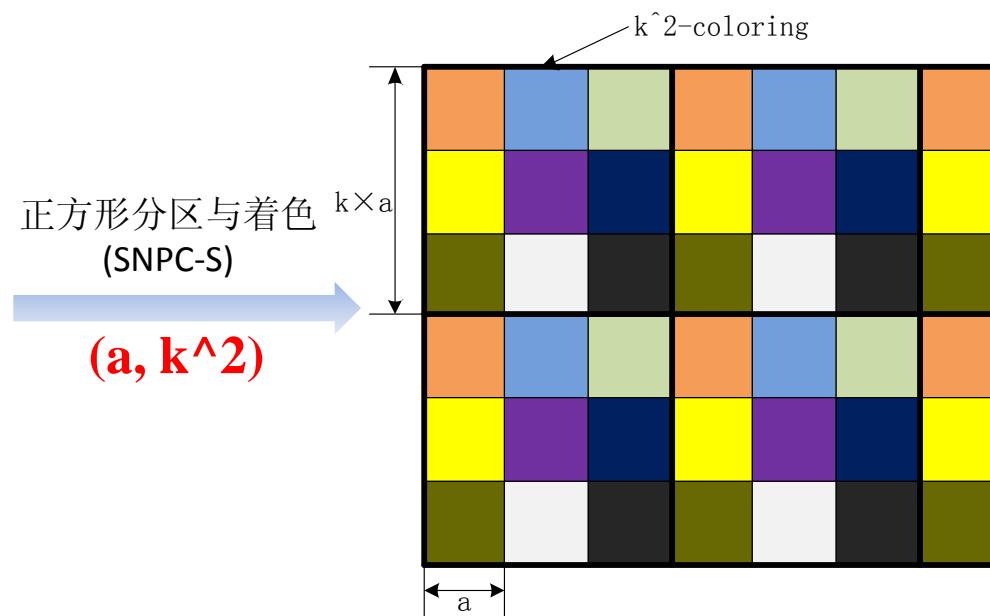
输出：调度序列 $\mathcal{A}_1, \mathcal{A}_2, \dots$

- 1: 初始化: $h \leftarrow 0$;
 - 2: **while** $\mathcal{A} \neq \emptyset$ **do**
 - 3: 从 \mathcal{A} 中随机选择一条链路, 并移动到 \mathcal{A}_h 中, 同时让 $h \leftarrow h + 1$, $\mathcal{D} \leftarrow \emptyset$;
 - 4: **for** 集合 $\mathcal{A} \setminus \mathcal{D}$ 中的每条个发送节点 v **do**
 - 5: **if** $\forall u \in \mathcal{A}_h$, 满足 $d(v, p(u)) > \rho \cdot r$ 且 $d(u, p(v)) > \rho \cdot r$ **then**
 - 6: 将节点 v 从 \mathcal{A} 移动到 \mathcal{A}_h 中;
 - 7: **else**
 - 8: 将节点 v 从 \mathcal{A} 移动到 \mathcal{D} 中;
 - 9: **end if**
 - 10: **end for**
 - 11: **end while**
-

➤ 收敛性分析

➤ 时间复杂度分析

基本思想：首先利用位置信息进行网络分区与着色，然后在某个时隙，任一小区最多只允许一个节点发送或者接收数据包；并且，只允许处于相同颜色小区的节点同时进行数据包发送和接收。





Algorithm :基于网络分区与着色的调度算法(NPCS)

输入: 图 G , 数据聚合树 T

输出: 每个节点 u 的调度(传输时隙序号): $Sche(u)$

/*初始化阶段*/

- 1: 初始化: $a = r/\sqrt{2}$, $k = \lceil (\rho + 1) \cdot \frac{r}{a} \rceil + 1$;
 - 2: 支配节点根据预设或定位技术获得自己在二维平面中的位置坐标 (x, y) ;
 - 3: 支配节点计算自己所在小区的颜色序号: $Color[\cdot] = \lfloor \frac{x\%(ka)}{a} \rfloor + \lfloor \frac{y\%(ka)}{a} \rfloor \times (k - 1)$;
- /*消息传递阶段*/
- 4: 对于支配节点 u : (1)将 $Color[u]$ 和集合 $\{1, 2, \dots, N_{DeC}(u)\}$ 中的某一个指定数发送给 T 中的每个孩子被支配节点, (2)将 $Color[u]$ 和集合 $\{1, 2, \dots, N_{CoC}(u)\}$ 中的某一个指定数发送给 T 中的每一个孩子连接节点;

/*调度计算阶段*/

- 5: 对于被支配节点 v : 设置 $Sche(v) = Color[p(v)] + (N - 1) \times k^2$;
 - 6: 对于支配节点 u : 设置 $Sche(u) = Color[u] + \frac{d_B - l_B(v)}{2} \times 13k^2 + (\Delta - 1) \times k^2$;
 - 7: 对于连接节点 w : 设置 $Sche(w) = Color[p(w)] + (N - 1) \times k^2 + \frac{d_B - l_B(v) - 1}{2} \times 13k^2 + (\Delta - 1) \times k^2$
-

调度方案可行



$$a \leq r/\sqrt{2}$$

$$k \geq \lceil (\rho + 1) \cdot \frac{r}{a} \rceil + 1$$

定理 2.5: 算法 **LGAS** 产生的调度方案是可行(feasible)的

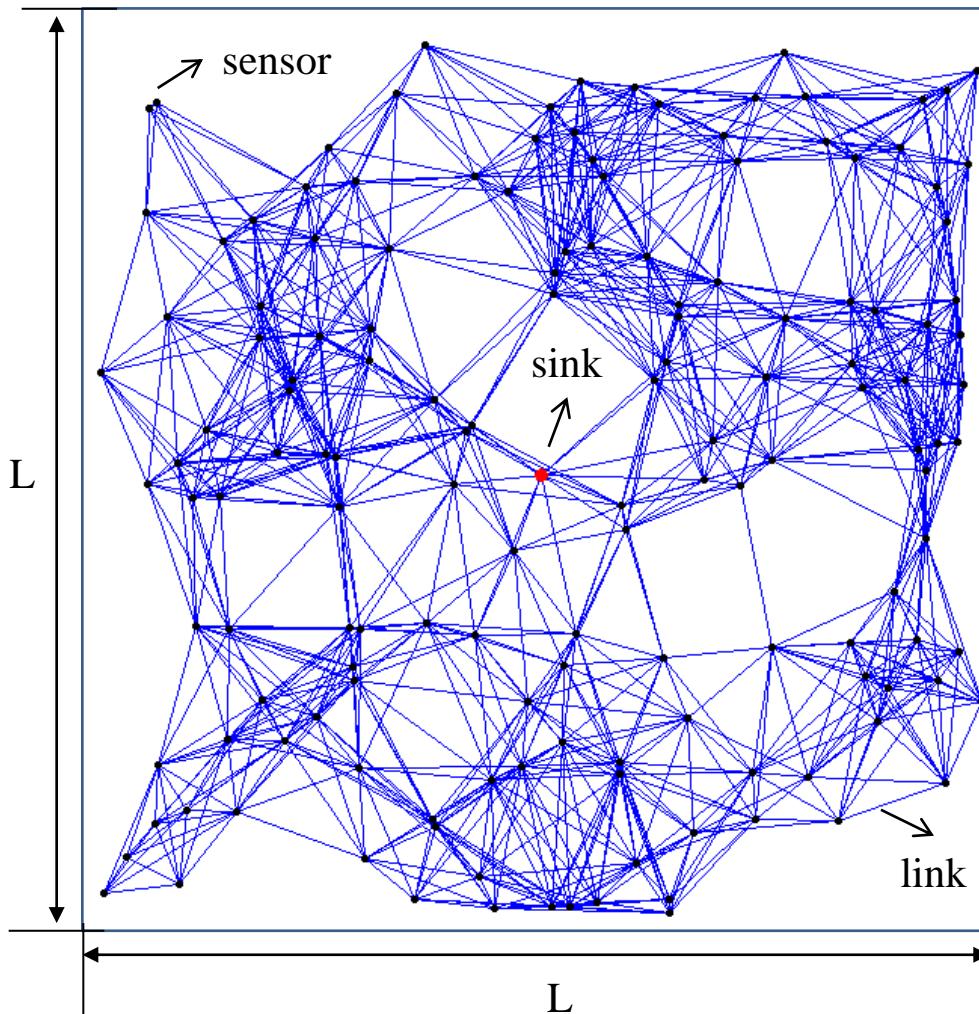
定理 2.6 : 算法 **LGAS** 产生的调度方案在最坏情形下的时延不超过
 $\lfloor \frac{\pi}{\sqrt{3}}(\rho + 1)^2 + (\frac{\pi}{2} + 1)(\rho + 1) + 1 \rfloor \cdot (15R + \Delta - 15)$, 其中 $\rho = r/r_I \geq 1$ 为干扰比, R 和 Δ 分别为图 G 的半径和最大节点度。

定理 2.7: 算法 **NPCS** 产生的调度方案是可行(feasible)的

定理 2.8 : 算法 **NPCS** 产生的调度方案的在最坏情形下的时延不超过
 $(\lceil \sqrt{2}(\rho + 1) \rceil + 1)^2 \cdot (16R + \Delta - 17)$, 其中 $\rho = r/r_I \geq 1$ 为干扰比, R 和 Δ 分别为图 G 的半径和最大节点度。

两种算法均是具有常量近似比的近似算法

实验设置



参数	值
网络监测区域大小	$150m \times 150m$
Sink节点位置	随机
节点数量	200~1000
节点传输半径	15m~40m
干扰比 ρ	1~5
单个时隙长度	1
每次实验重复次数	50

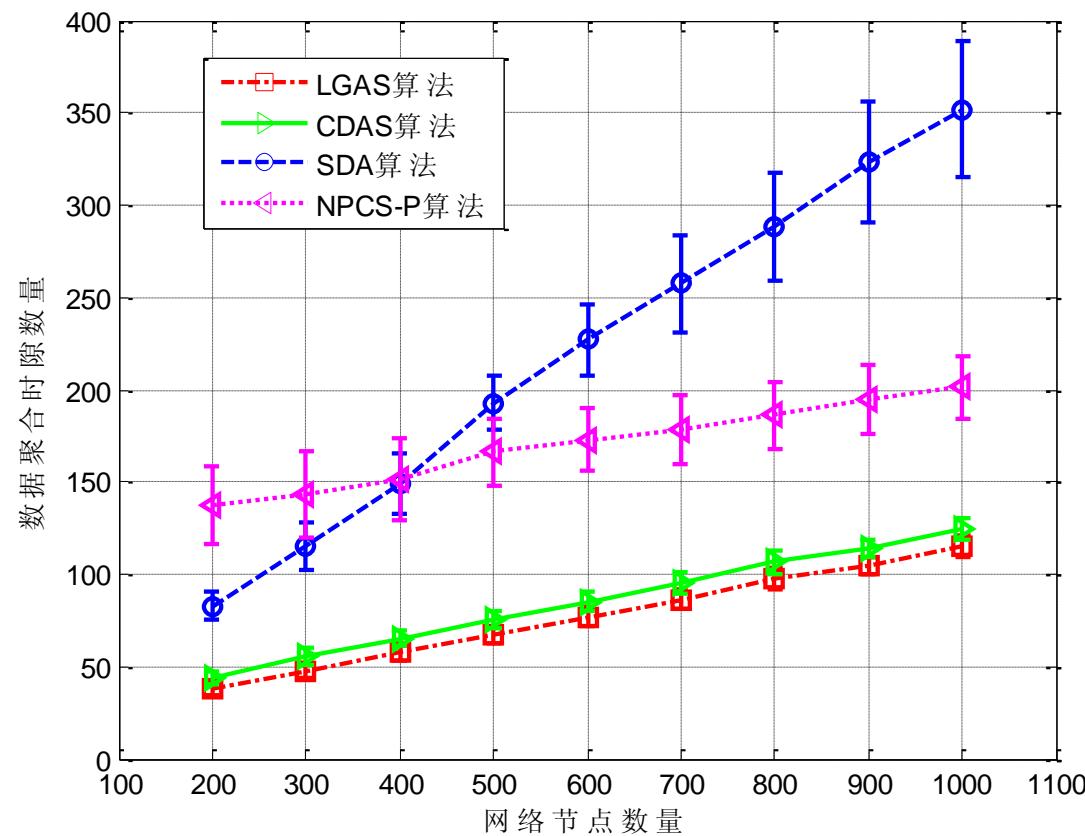
对比算法

- LGAS, LGAS-U(LGAS理论上界)
- NPCS, NPCS-U(NPCS理论上界)
- NPCS-P: 去除空闲时隙的NPCS算法
- SDA^[1]
- CDAS^[2]

[1] X. Chen, X. Hu, and J. Zhu, “Minimum data aggregation time problem in wireless sensor networks,” in *Mobile Ad-hoc and Sensor Networks*. Springer, 2005, pp. 133–142.

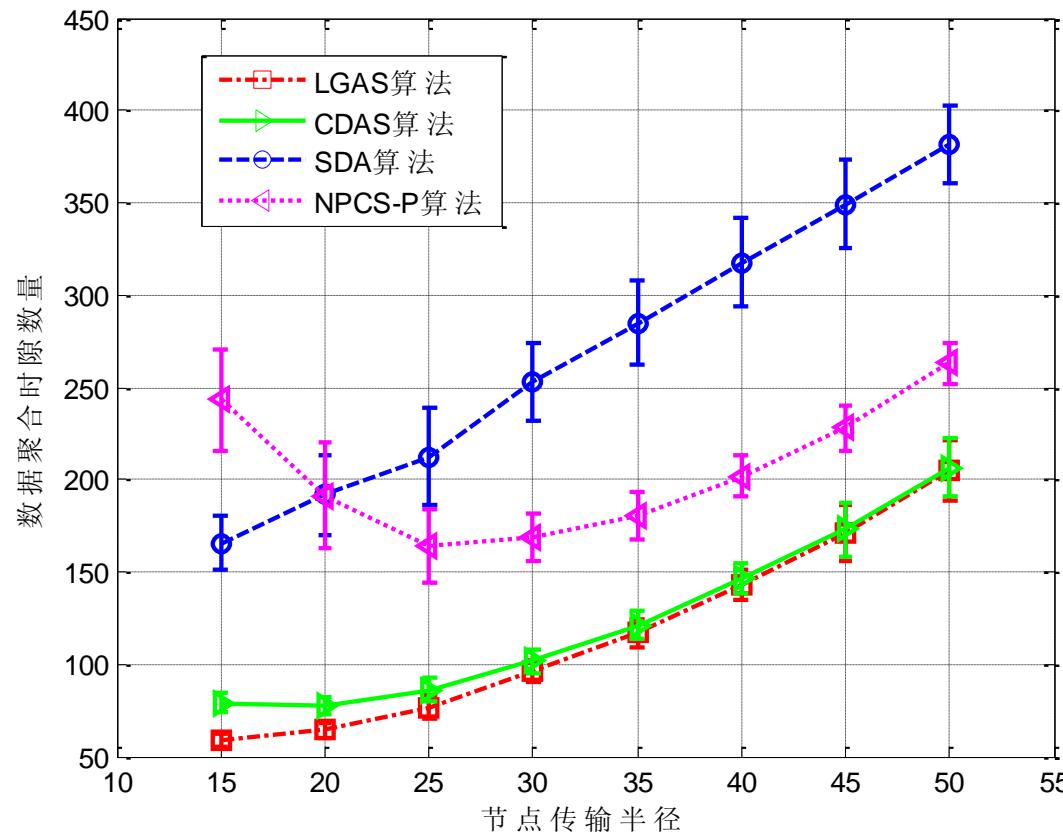
[2] X. Xu, X.-Y. Li, X. Mao, S. Tang, and S. Wang, “A delay-efficient algorithm for data aggregation in multihop wireless sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 163–175, 2011.

实验结果(1)：延时 vs. 节点数量



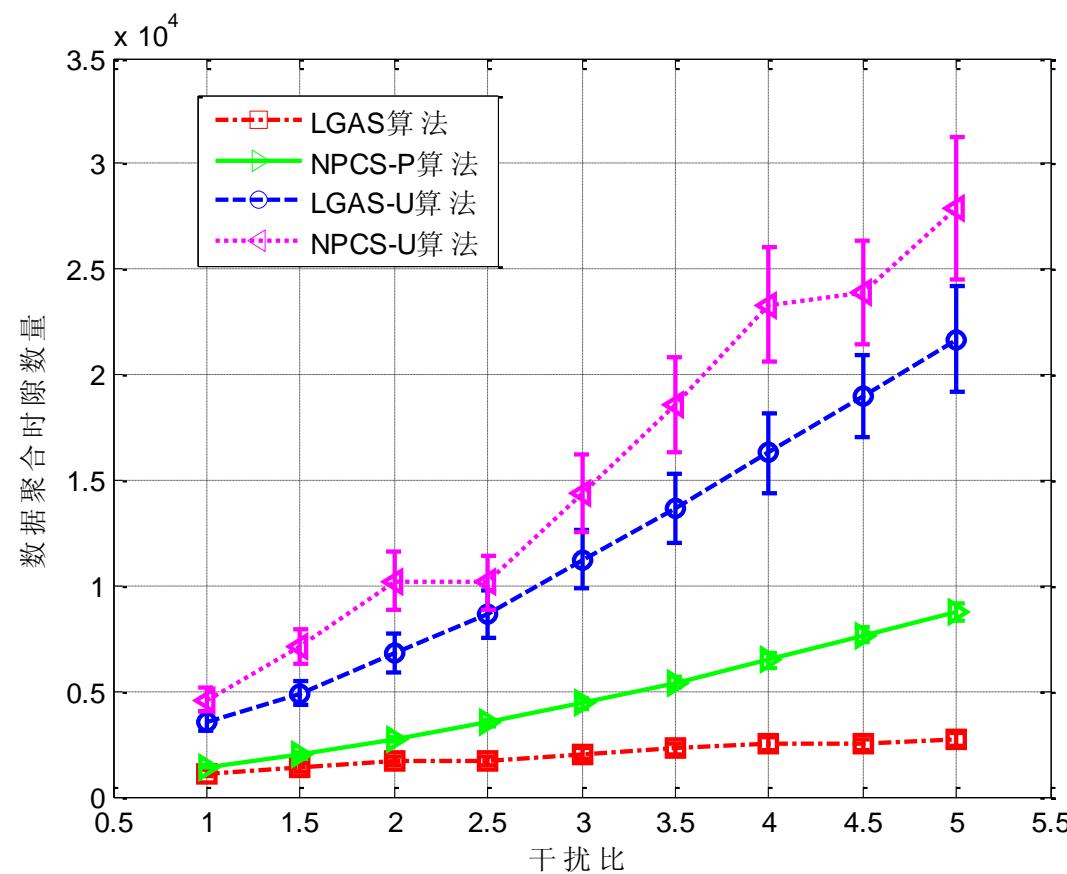
与文献中最好的算法**CDAS**相比，**LGAS**将数据聚合时延降低了8% ($n=1000$) 到14% ($n=200$)，即**10%左右**。

实验结果(2)：时延 vs. 节点传输半径



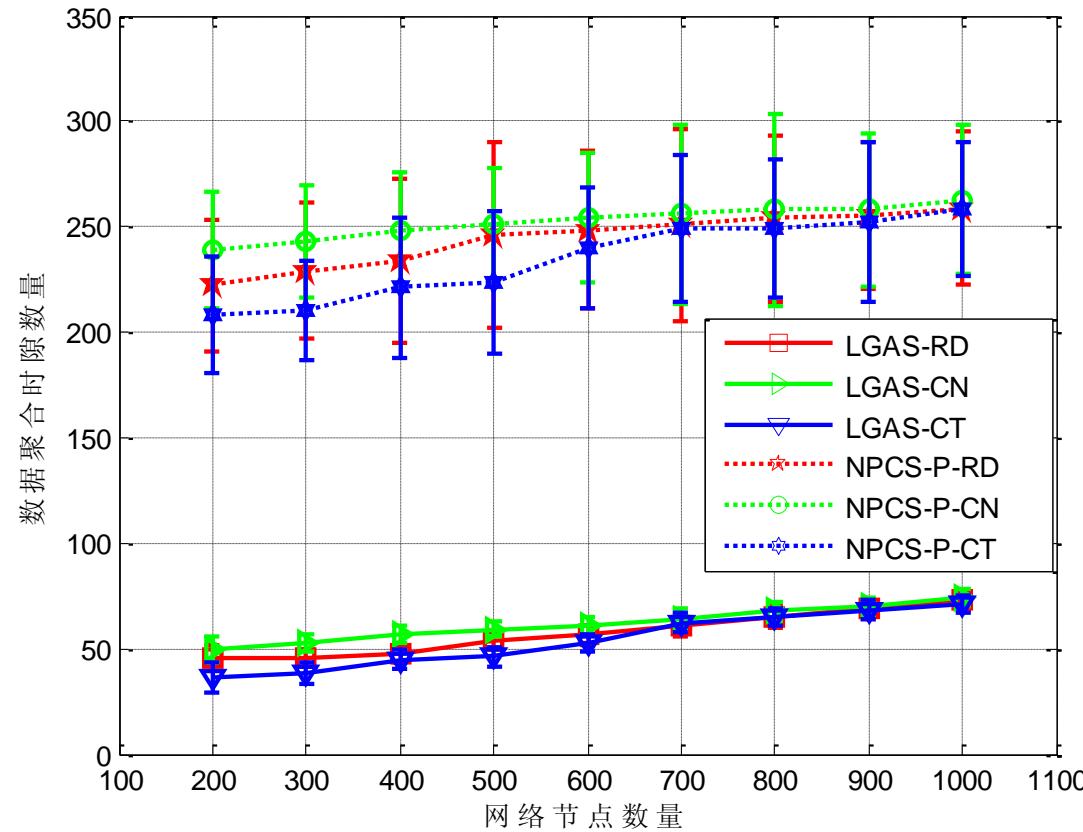
当传输半径=15时，LGAS的时延只有CDAS时延的74%左右；当传输半径>35时，**LGAS**和**CDAS**性能相当。

实验结果(3)：时延 vs. 干扰比



当 ρ 增大时，算法时延均单调增大，但是无论 ρ 如何变化，LGAS和NPCS-P的时延均没有超过并且显著低于各自的理论上界。

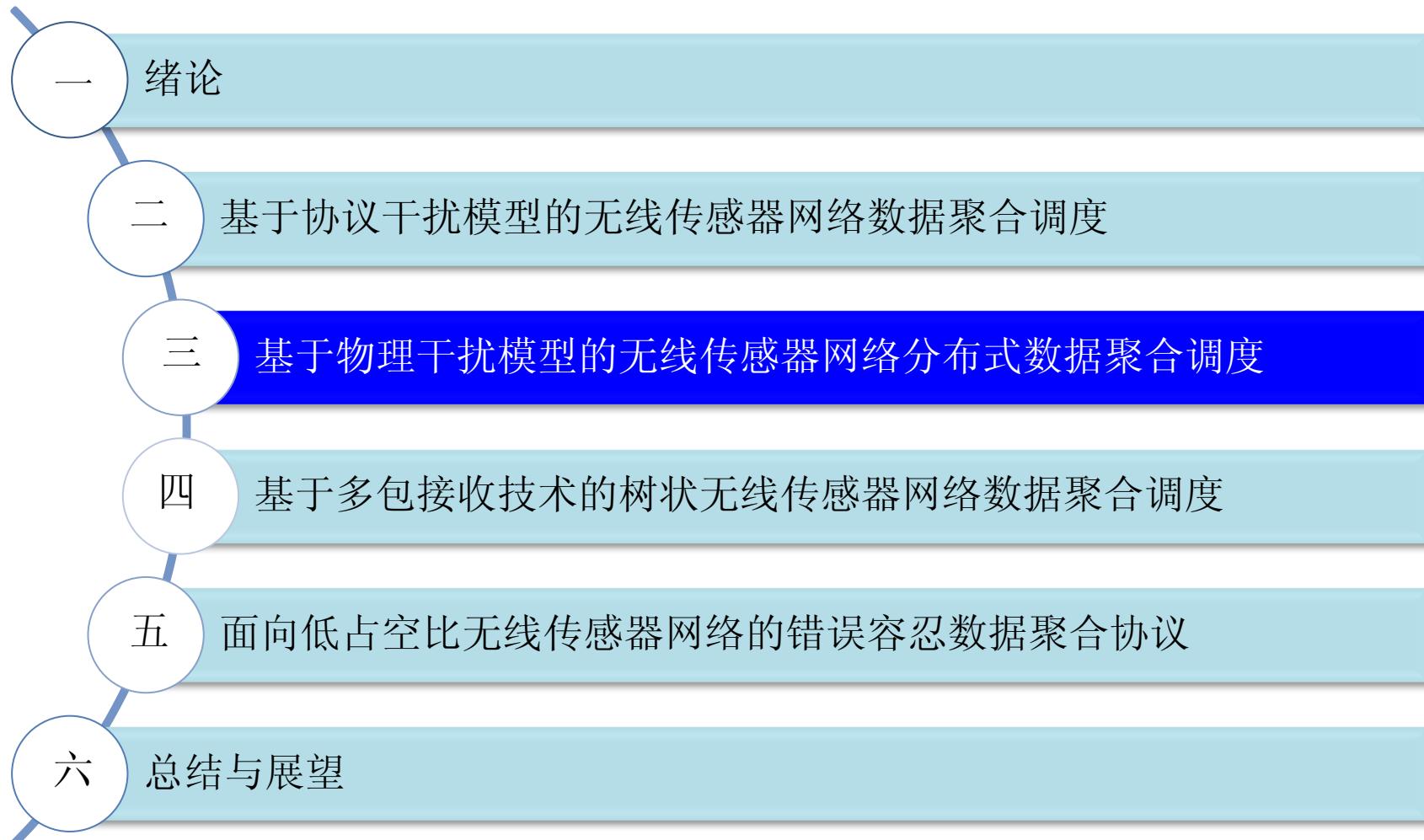
实验结果(4)：时延 vs. sink位置

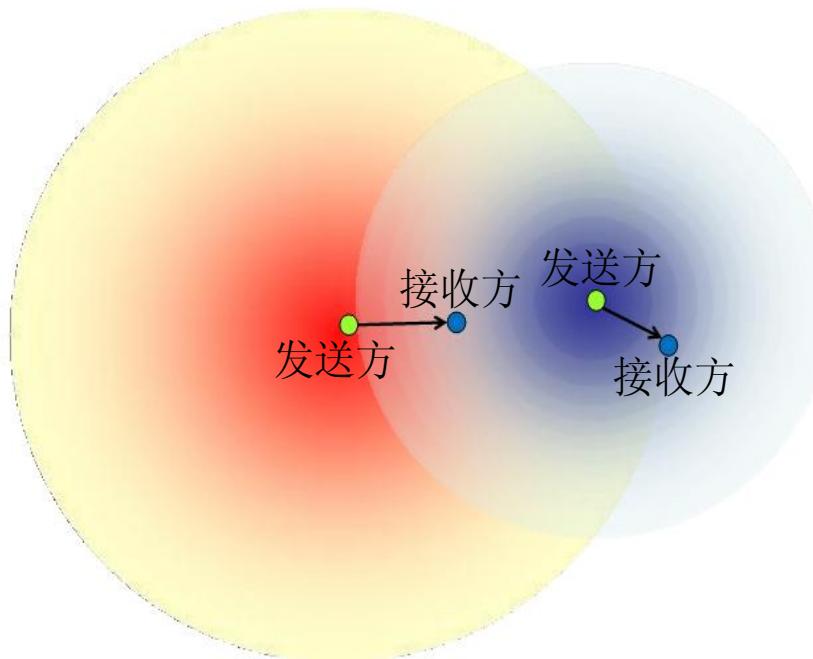


相同情况下，sink边缘布设(**CN**)时算法平均时延最大，随机布设(**RD**)时次之，中心布设(**CN**)时算法平均时延最小。

本章研究了协议干扰模型下的无线传感器网络数据聚合调度问题，取得了以下创新点：

1. 提出了一种两阶段数据聚合方案。首先，提出了基于连通支撑集的改进型数据聚合树构造算法**IDATC**；然后，分别基于分层贪心策略和网络分区与着色策略提出了集中式数据聚合调度算法**LGAS**和分布式数据聚合调度算法**NPCS**。
2. 论证了方案的可行性；通过推导**LGAS**算法和**NPCS**算法的时延上界，证明了两种调度算法均是具有常量比的近似算法。
3. 仿真实验表明，本文调度方案的平均时延性能远好于理论上界。并且与文献中的代表性算法相比，本文算法在多种网络场景下均能一定程度上降低数据聚合的总时延。





并发节点集合

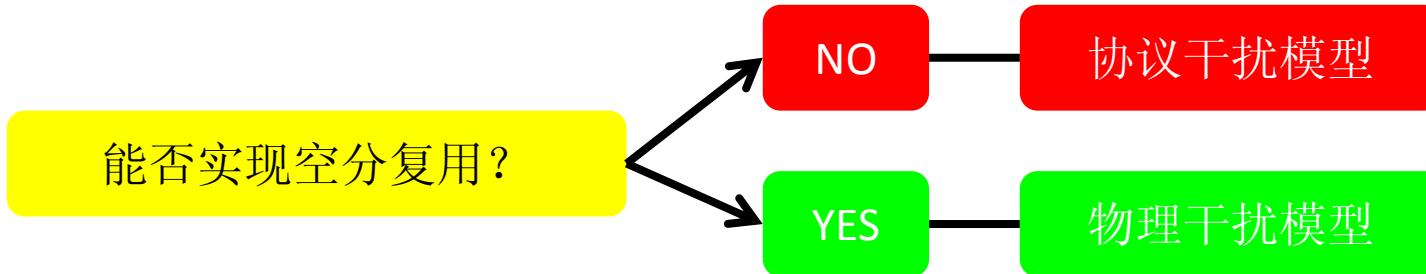
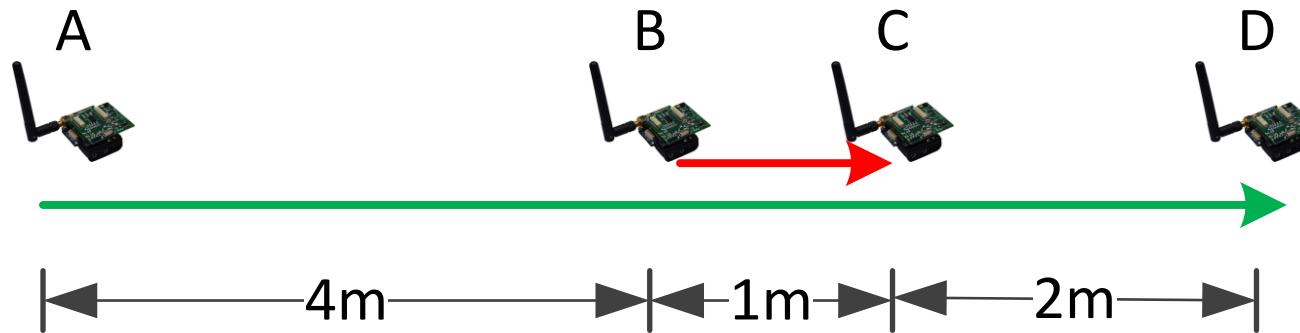
$$SINR(u, v, \mathcal{S}_t) = \frac{Q(u, v)}{\xi + \sum_{w \in \mathcal{S}_t} Q(u, w)} \geq \beta$$

信号强度 $Q(u, v) = \frac{Q_v}{d^\alpha(u, v)}$

干扰强度

α : 路径损耗因子; β : SINR解码门限; ξ : 背景噪声强度

SINR: Signal-to-Interference-plus-Noise Ratio, 信号干扰噪声比模型



设 $\alpha = 3, \beta = 3, \xi = 10nW, Q_A = 1dBm, Q_B = -15dBm$

- 链路A->D的SINR: $\frac{1.26mW/(7m)^3}{0.01\mu W+31.6\mu W/(3m)^3} \approx 3.11 \geq \beta$ ✓
- 链路B->C的SINR: $\frac{31.6\mu W/(1m)^3}{0.01\mu W+1.26mW/(5m)^3} \approx 3.13 \geq \beta$ ✓

物理干扰模型带来的挑战

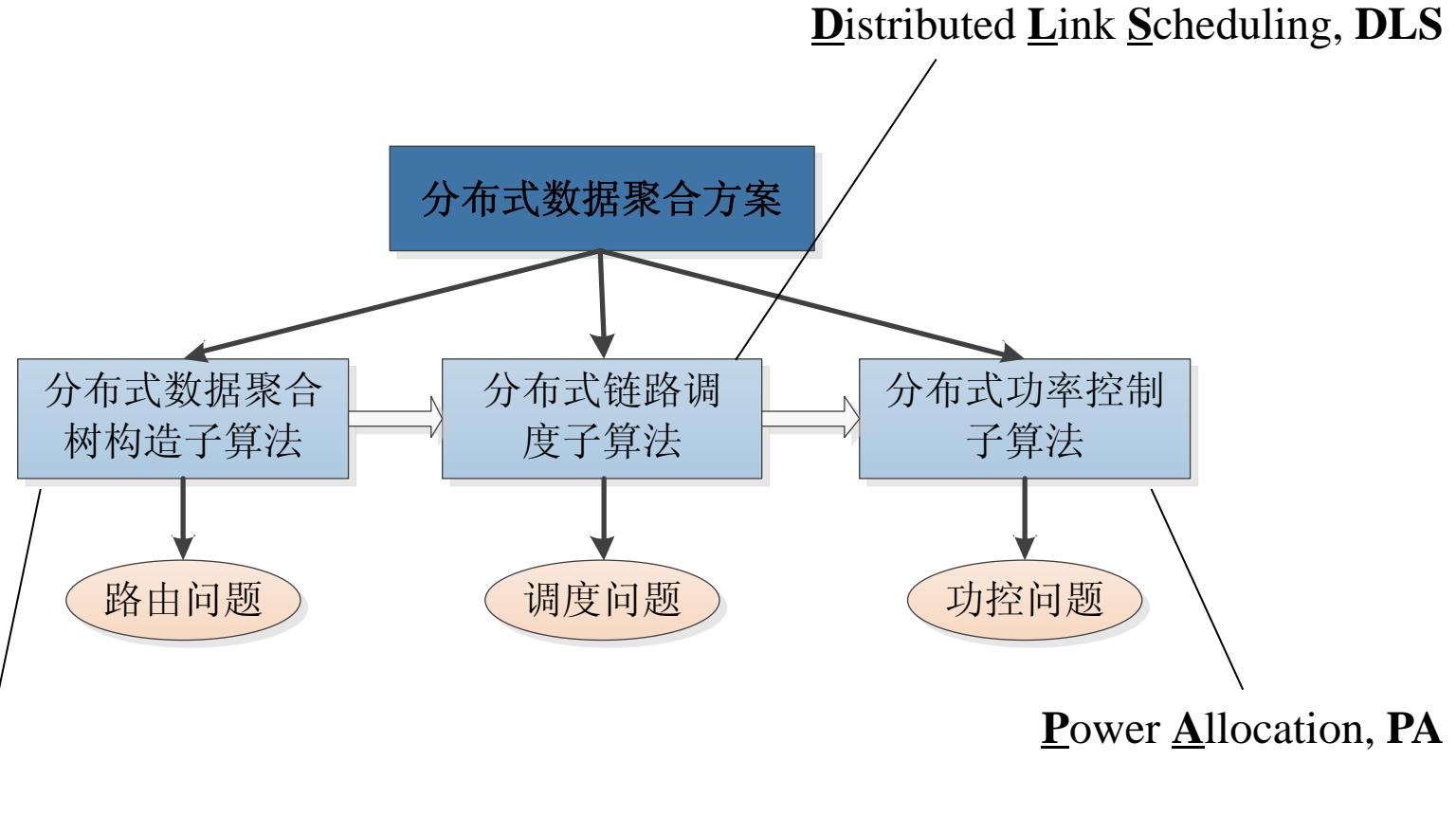
全局性

来自距离很远的并发链路的干扰尽管强度非常微弱，但是不能完全被忽略

非二元性

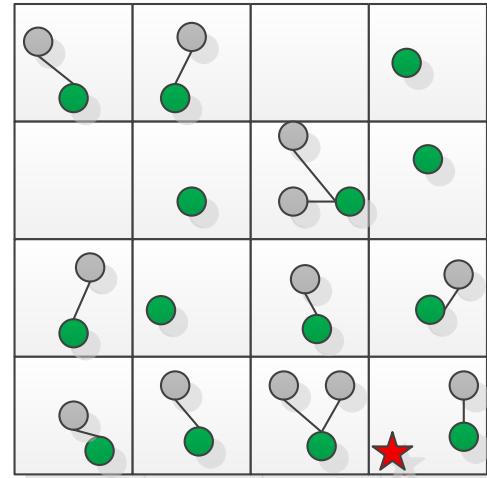
两条链路原本能够共存，但是由于另一条链路的加入，它们的干扰关系可能发生改变而变得无法共存

物理干扰模型给调度算法(尤其是分布式调度算法)的设计带来了严峻挑战

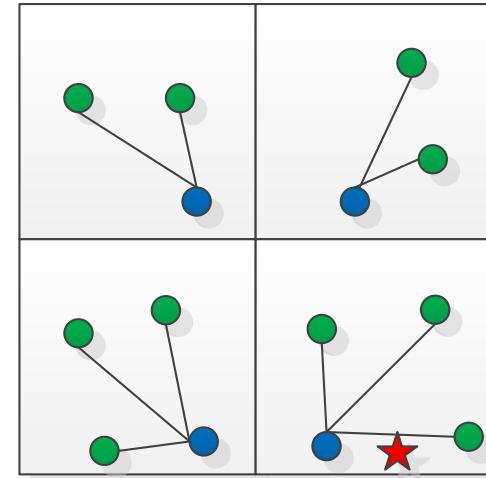


Distributed Data Aggregation Tree Construction, DDATC

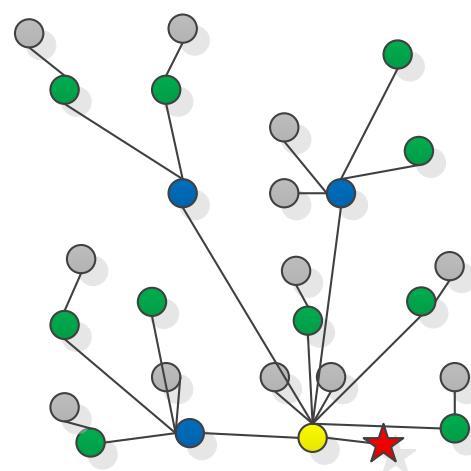
DDATC以迭代的方式工作，每轮迭代进行网络分区和小区首部节点选举，只有当前迭代被选择为首部的传感器节点才参与下一次迭代，直到只剩一个小区为止。



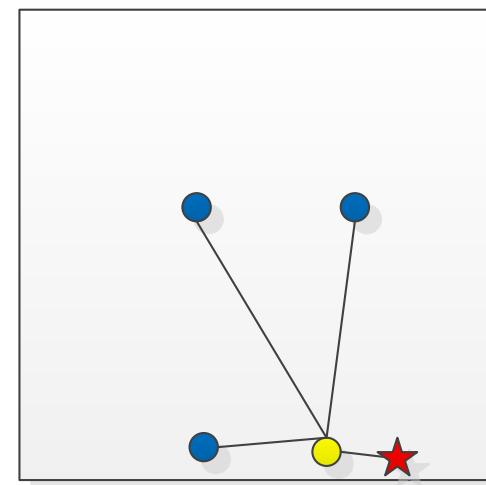
(a)



(b)



(d)



(c)

Algorithm :分布式数据聚合树构造算法(**DDATC**)

输入: 节点集合 \mathcal{V} , sink节点 v_s

输出: 数据聚合树 T , 每个传感器 $v \in \mathcal{V} \setminus \{v_s\}$ 的父节点 P_v

- 1: sink节点 v_s 向全网广播其位置信息;
 - 2: **for** $m = 1, 2 \cdots, m^*$ **do**
 - 3: $\mathcal{V} \setminus \{v_s\}$ 中的每个节点 v 根据预设或定位技术获取自己在平面中的地理位置;
 - 4: 采用一族垂直线 $x = i \cdot 2^{m-1}a, i \in \mathbb{Z}$ 和一族水平线 $y = j \cdot 2^{m-1}a, j \in \mathbb{Z}$ 将二维平面划分为半开半闭、大小为 $2^{m-1}a \times 2^{m-1}a$ 的正方形小区;
 - 5: 节点 v 根据自身位置信息判断所处的小区 $\text{Cell}(v)$, 并采用本地广播算法向半径为 $\sqrt{2} \cdot 2^{m-1}a$ 的范围广播自身位置和身份信息;
 - 6: 节点 v 计算自身及 $\text{Cell}(v)$ 中其他节点到 v_s 的距离, 如果 v 距离 v_s 最近, 则 v 称为所在小区首部Header, 否则 v 将距离 v_s 最近的邻居节点确定为父节点 P_v ;
 - 7: 将 \mathcal{V} 所有父节点已经确定的节点移除, 并开始下一次迭代;
 - 8: **end for**
 - 9: 唯一的小区首部节点将 v_s 确定为父节点;
-

引理 3.1: 在 DDATC 算法构造的数据聚合树 T 中, 设 \mathcal{C}_v 表示节点 v 的孩子节点集合, \mathcal{N}_m 表示在算法在第 m ($1 \leq m \leq m^*$)次迭代中的成员节点集合, 则我们有:

$$|\mathcal{C}_v \cap \mathcal{N}_m| \leq \begin{cases} \lfloor \frac{2a^2}{\sqrt{3}} + 2a + 1 \rfloor & m = 1 \\ 3 & m = 2, \dots, m^* \end{cases} \quad (1a)$$

$$(1b)$$

DLS基于迭代网络分区与着色技术，采用与DDATC一样的分区方法，在每次调度迭代时，进行小区着色，并让具有相同颜色小区中的链路同时传输，同时确保每个小区在任何一个时隙最多只有一条链路传输

Algorithm :分布式链路调度算法(**DLS**)

输入: 数据聚合路由树

输出: 每个传感器 $v \in \mathcal{V} \setminus \{v_s\}$ 的传输时隙 H_v

- 1: **for** $m = 1, 2 \cdots, m^*$ **do**
 - 2: 采用一族垂直线 $x = i \cdot 2^{m-1}a, i \in \mathbb{Z}$ 和一族水平线 $y = j \cdot 2^{m-1}a, j \in \mathbb{Z}$ 将二维平面划分为半开半闭、大小为 $2^{m-1}a \times 2^{m-1}a$ 的正方形小区；
 - 3: 节点 v 根据自身位置坐标 (x, y) 计算所在小区颜色序号： $\text{Color}[v] = \lfloor \frac{x\%(k(m)2^{m-1})}{2^{m-1}a} \rfloor + \lfloor \frac{y\%(k(m)2^{m-1}a)}{2^{m-1}a} \rfloor \times (k(m) - 1)$ ；
 - 4: 每个小区的首部节点 v 向集合 $\mathcal{C}_v \cap \mathcal{N}_m$ 中的每个节点发送一个来自集合 $\{1, 2, \cdots, |\mathcal{C}_v \cap \mathcal{N}_m|\}$ 各异的整数；
 - 5: 每个小区的非首部节点 v 在当前迭代轮次的传输时序确定为： $H_v = (\text{Num}[v] - 1) \cdot k^2 + \text{Color}[v]$ ，其中 $\text{Num}[v]$ 为 v 收到的来自 P_v 的整数；
 - 6: **end for**
 - 7: \mathcal{H}_{m^*} 中的节点向 v_s 发送数据；
-

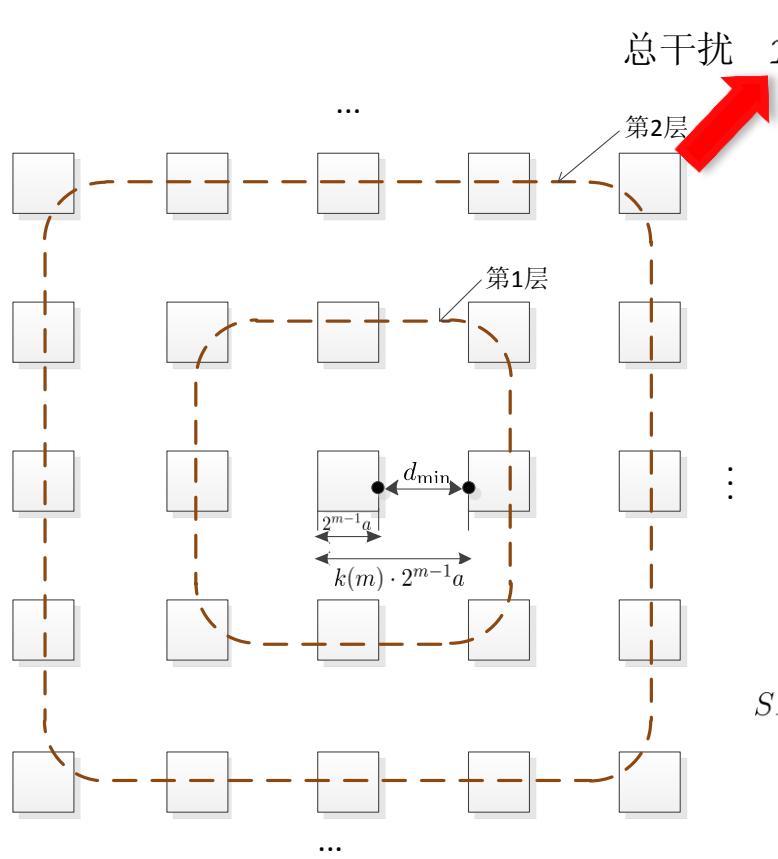
□ 均匀功率分配(Uniform Power Allocation, PA-U)

- 在第 m 个调度阶段进行传输的所有节点采用相同的传输功率 $Q_0(m)$
- 当 $1 \leq m \leq m^*$ 时, $Q_0(m) = \delta \cdot [\beta\xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha]$, 这里, $\delta \geq 1$ 为可调参数
- 当 $m = m^* + 1$ 时, 发射功率取 $\beta\xi \cdot (\sqrt{2}L)^\alpha$

□ 线性功率分配(Linear Power Allocation, PA-L)

- 在任何一个调度阶段, 链路的传输功率均与链路的能量损耗成正比
- 对于长度为 $d(v, P_v)$ 的链路 (v, P_v) , Q_v 与 $d^\alpha(v, P_v)$ 成正比, 即 $Q_v = \epsilon \cdot d^\alpha(v, P_v)$, 这里参数可调 ϵ 满足 $\epsilon \geq \beta\xi$

定理 3.2: 在物理干扰模型下, 当物理层采用均匀功率(或者线性)分配策略时, 如果 $k(m) \geq (\frac{4\delta}{\delta-1}(1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} + 1$ (或者 $k(m) \geq (\frac{4\epsilon}{\epsilon-\beta\xi}(1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} + 1$), 则算法 **DLS** 产生的调度方案是可行的。也就是说, 在任何一个时隙所调度的每一条链路的接收方均符合 SINR 解码条件。



$$\begin{aligned}
 \text{总干扰 } \mathcal{I} &\leq \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} Q_0(m) \cdot ((k(m)-1)2^{m-1}a \cdot \sqrt{i^2 + j^2})^{-\alpha} \\
 &= 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} \left(\sum_{i=1}^{\infty} i^{-\alpha} + \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} (\sqrt{i^2 + j^2})^{-\alpha} \right) \\
 &\leq 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (\zeta(\alpha) + 2^{-\frac{\alpha}{2}} \cdot \sum_{i=1}^{\infty} i^{-\frac{\alpha}{2}} \cdot \sum_{j=1}^{\infty} j^{-\frac{\alpha}{2}}) \\
 &= 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (\zeta(\alpha) + 2^{-\frac{\alpha}{2}} \cdot \sum_{i=1}^{\infty} i^{-\alpha}) \\
 &= 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (1 + 2^{-\frac{\alpha}{2}}) \cdot \zeta(\alpha) \\
 \\
 SINR &= \frac{Q_0(m) \cdot d^{-\alpha}(v, P_v)}{\mathcal{I} + \xi} \\
 &\geq \frac{\delta \cdot [\beta\xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha] \cdot (\sqrt{2} \cdot 2^{m-1}a)^{-\alpha}}{4\delta \cdot [\beta\xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha] \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (1 + 2^{-\frac{\alpha}{2}}) \cdot \zeta(\alpha) + \xi} \\
 &= \frac{\delta\beta\xi}{4\delta\beta\xi \cdot (1 + 2^{-\frac{\alpha}{2}}) \cdot \zeta(\alpha) \cdot (\sqrt{2})^\alpha \cdot [k(m)-1]^{-\alpha} + \xi} \\
 &\geq \beta
 \end{aligned}$$

定理3.3：在最坏情况下，数据聚合调度算法**DLS**所消耗的时隙数量不超过

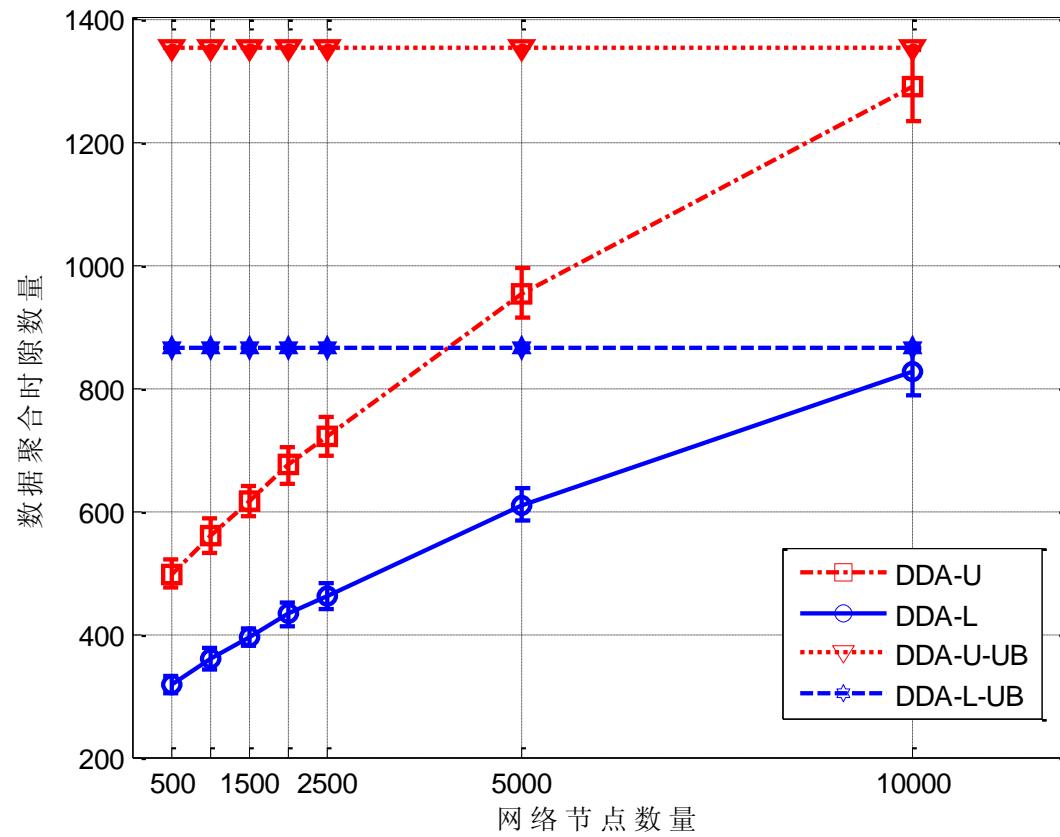
$$\lfloor \frac{2a^2}{\sqrt{3}} + 2a + 1 \rfloor \cdot K + 3K \cdot m^* - 3K + 1.$$

注：这里 $m^* = \lceil \log_2(L/a) + 1 \rceil$ 表示最大迭代次数。 K 是常数，取值与物理层采用的功率控制策略有关。采用均匀功率控制时， K 取 $\lceil (\frac{4\delta}{\delta-1}(1+2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} \rceil + 1$ ；采用线性功率控制时， K 取 $\lceil (\frac{4\epsilon}{\epsilon-\beta\xi}(1+2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} \rceil + 1$ 。

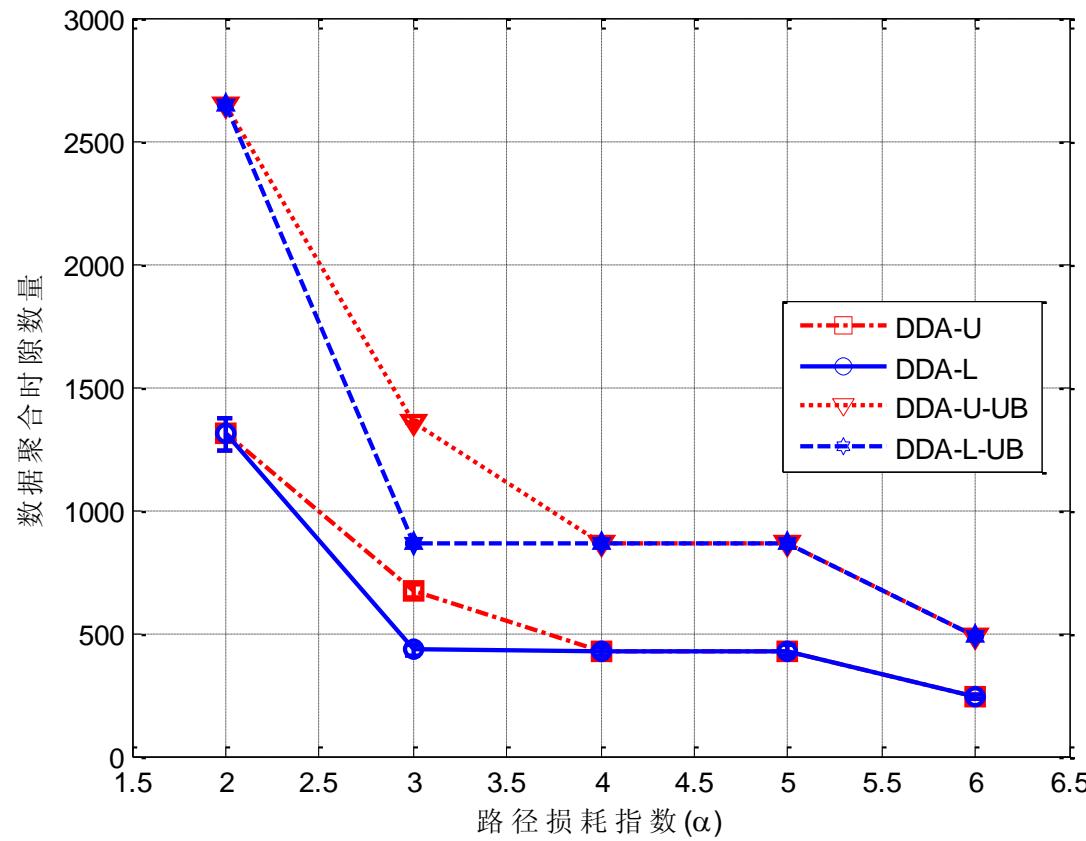
DLS算法的时延上界与网络规模 n 无关，显示其具有良好的可扩展性

- 工具： MATLAB
- 考虑二维平面上的传感器网络， sink节点位于监测区域中心
- 单个时隙的长度归一化为1， 时延=所消耗的时隙数量
- 50次独立实验， 取平均值和标准差
- 对比算法
 - **DDA-U:** DDATC+DLS+PA-U
 - **DDA-L:** DDATC+DLS+PA-L
 - **DDA-U-UB:** DDA-U的理论上界
 - **DDA-L-UB:** DDA-L的理论上界

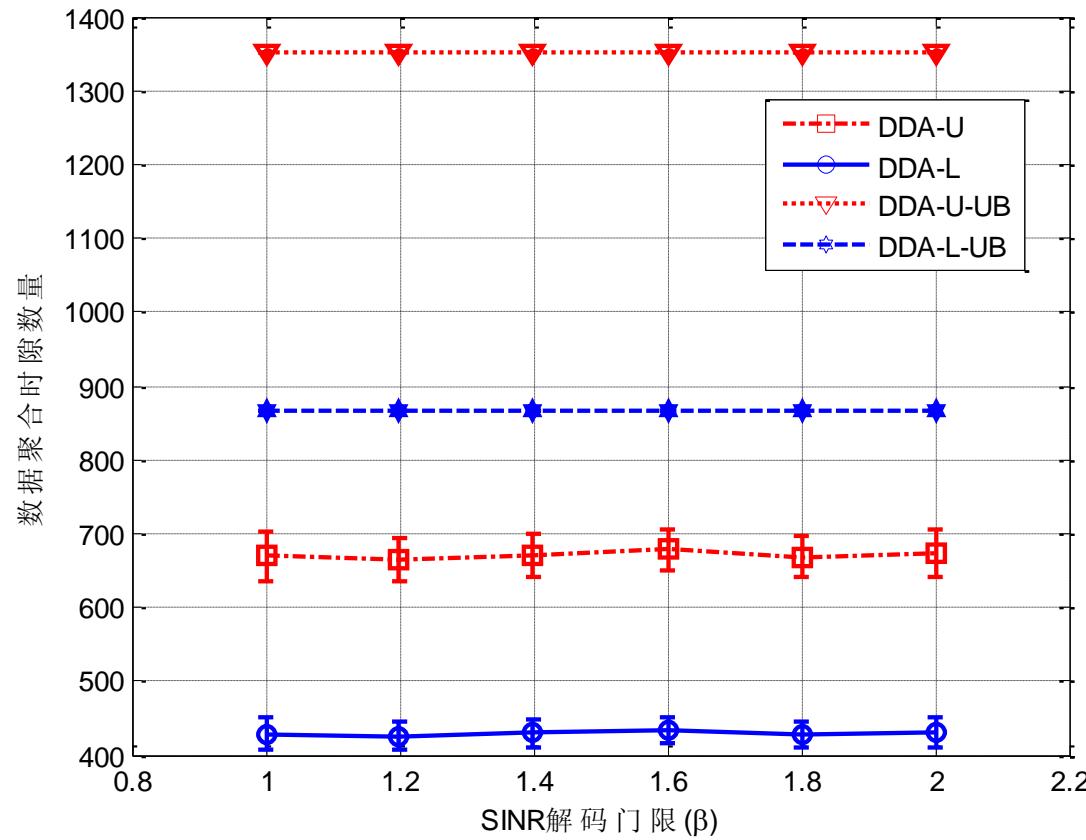
实验结果(1): 时延 vs. 节点数量



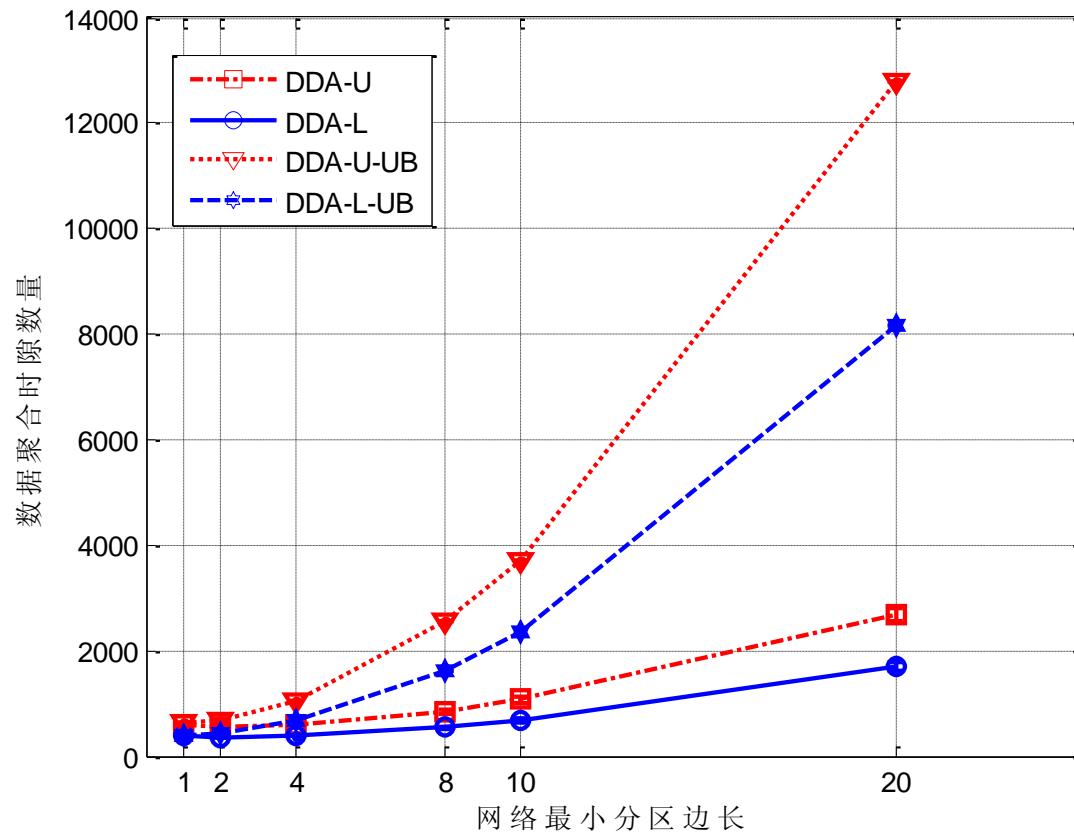
实验结果(2): 时延 vs. 路径损耗指数



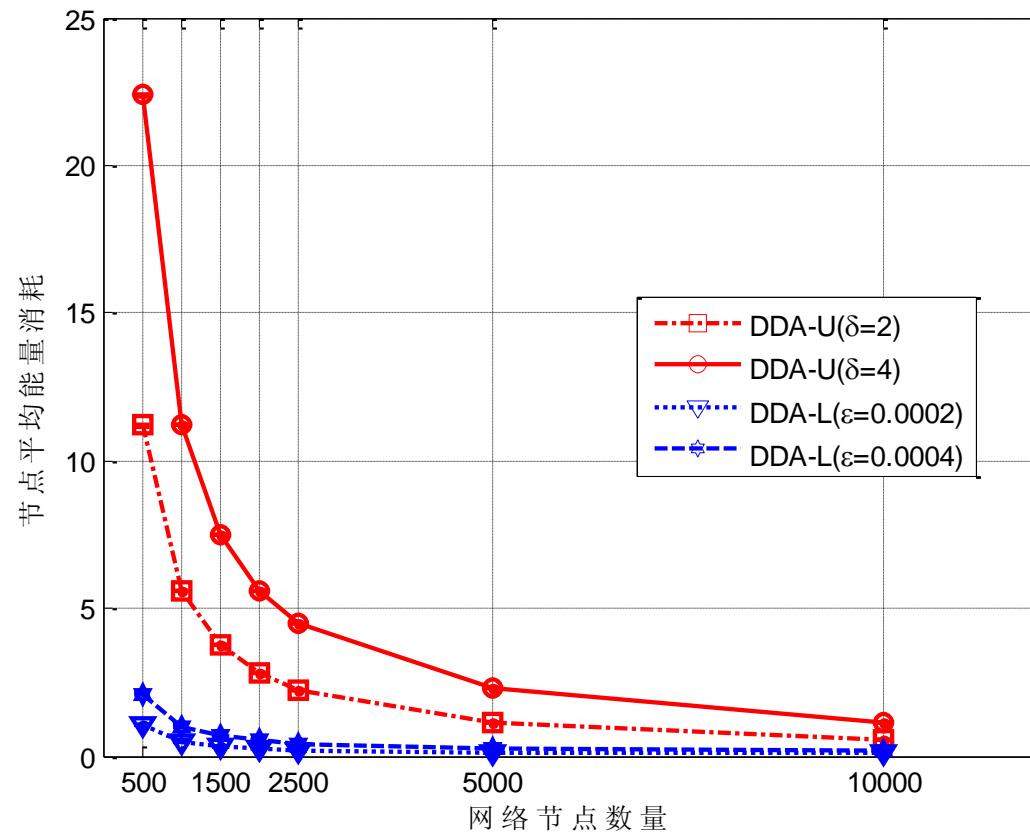
实验结果(3): 时延 vs. SINR解码门限



实验结果(4): 时延 vs. 网络最小分区边长

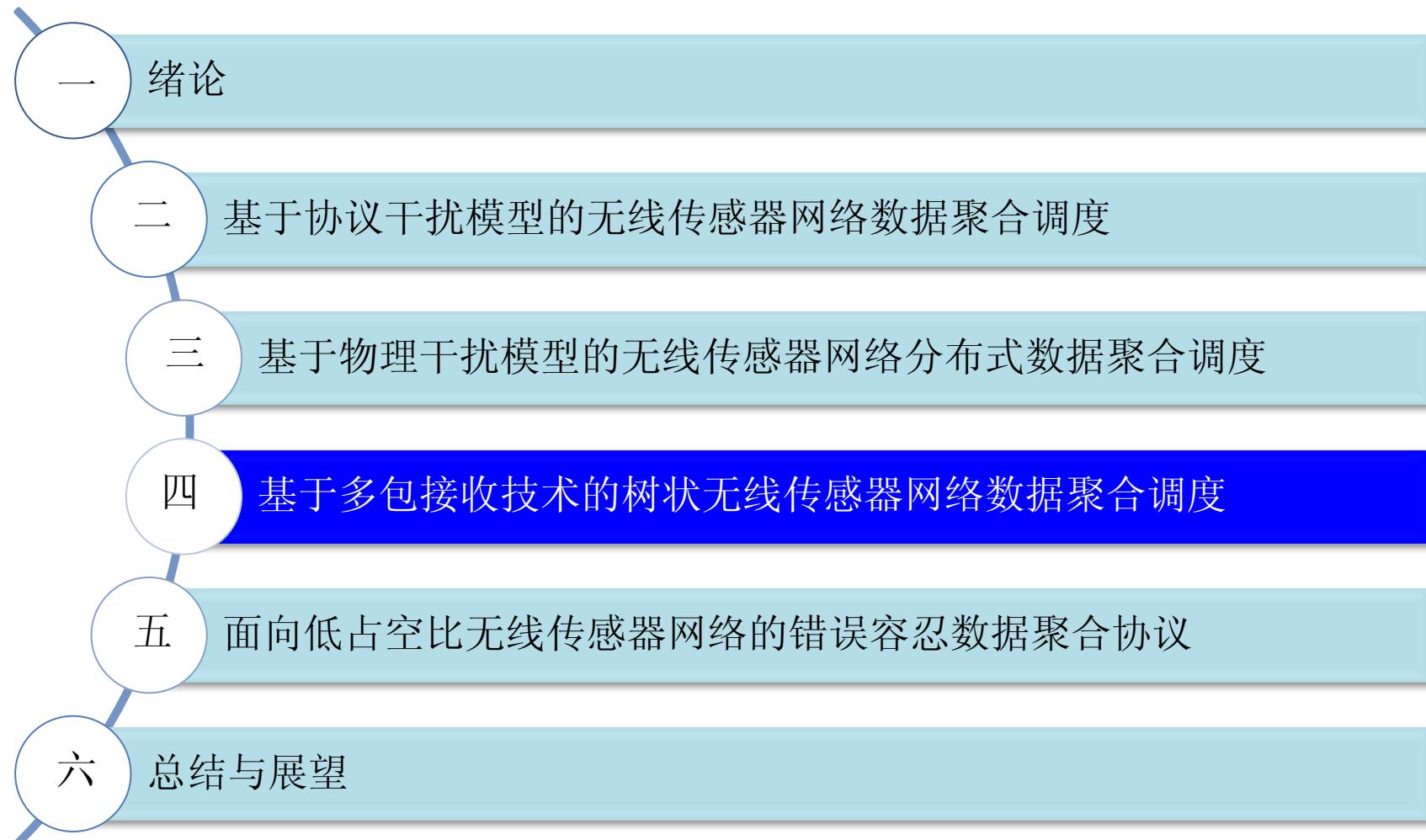


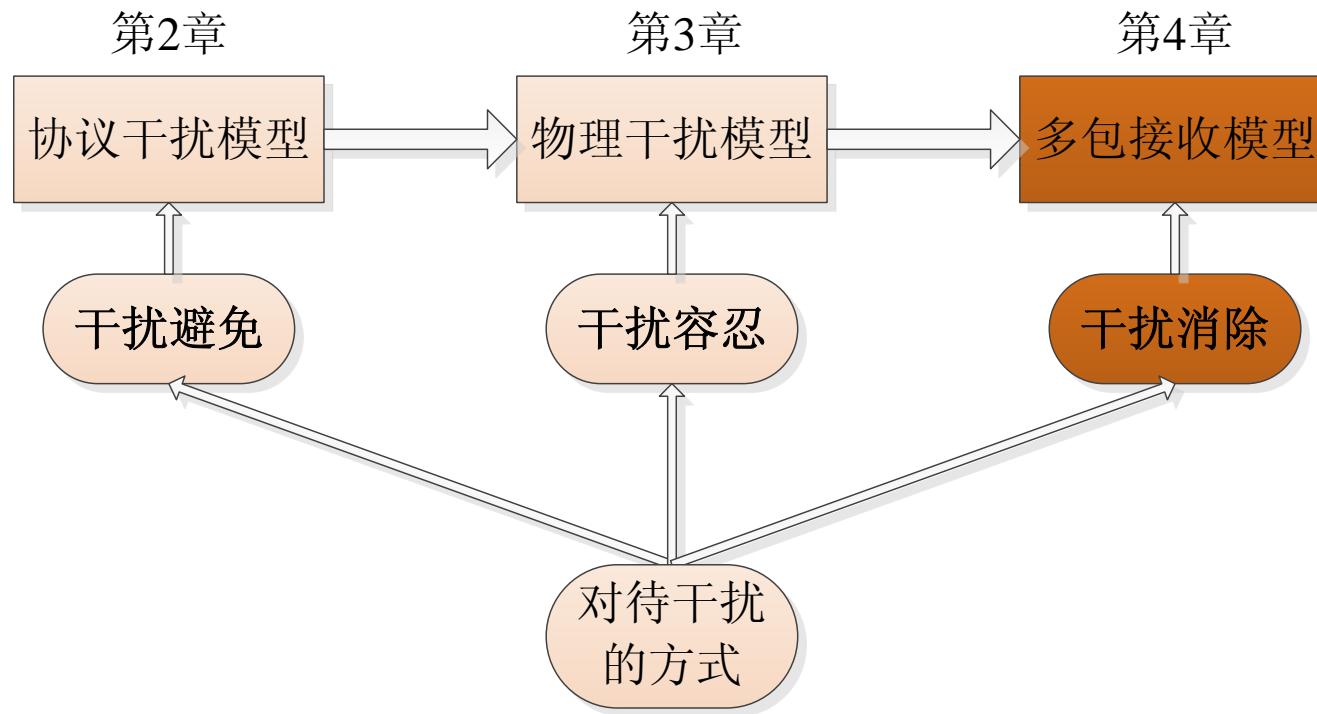
实验结果(5): 节点平均能耗 vs. 功率控制策略



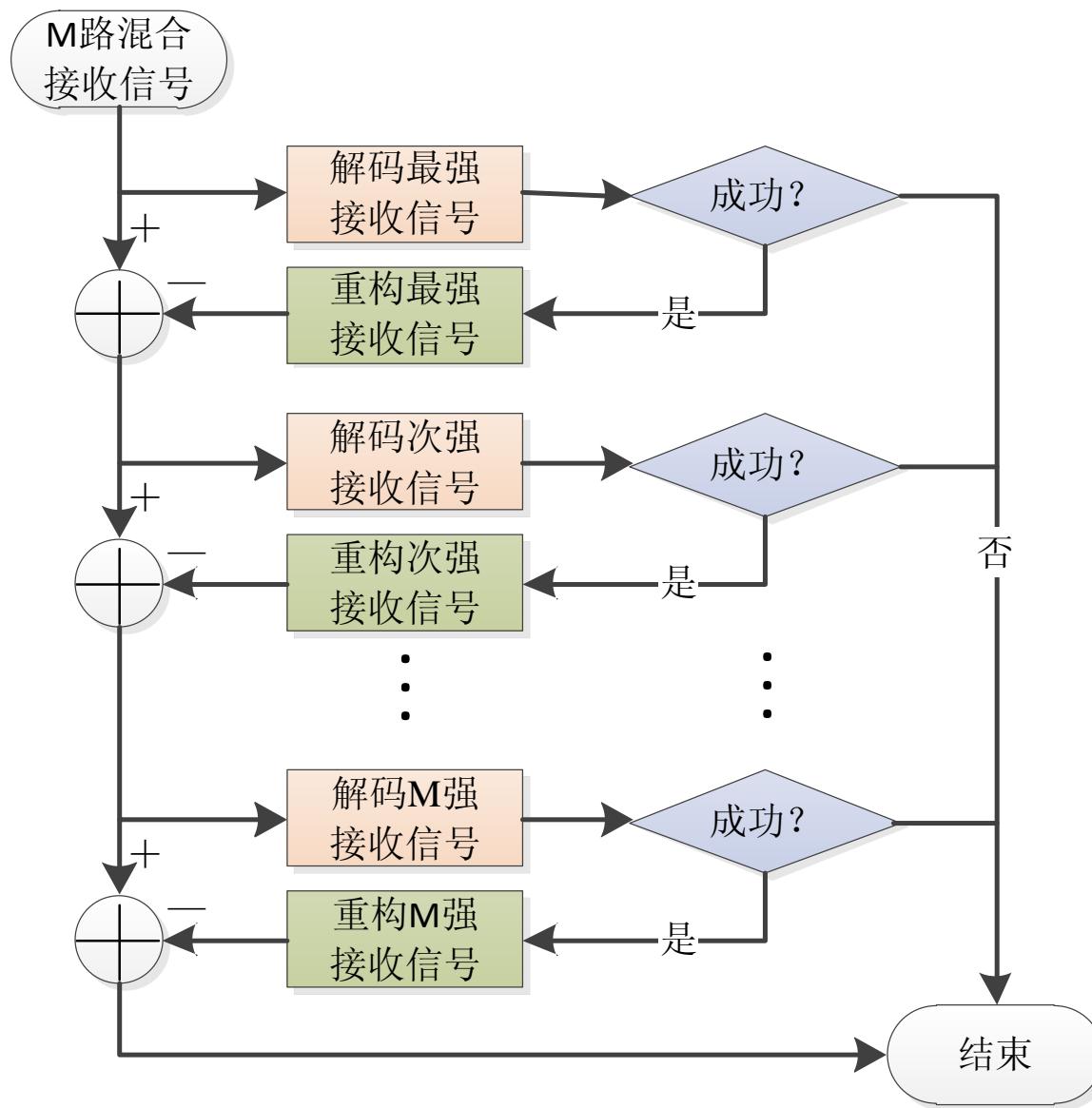
研究了物理干扰模型下的无线传感器网络数据聚合调度问题，取得了以下创新点：

1. 提出了一种分布式数据聚合解决方案**DDA**。该方案以迭代网络分区与着色技术为基础，由分布式数据聚合树构造子算法**DDATC**、分布式链路调度子算法**DLS**和两种功率控制子算法**PA-U**与**PA-L**联合而成。
2. 通过合理配置调度和功率控制模块的参数，证明了该方案在物理干扰模型下的可行性。同时，推导了该方案的理论聚合时延上界。分析结果表明，**DDA**的时延上界与网络规模无关，显示了**DDA**具有良好的可扩展性。
3. 通过大量的仿真实验评价了该方案在不同场景下(网络规模、SINR参数、最小分区边长等)的平均时延、能耗等性能。





基于干扰消除的理念，将物理层多包接收模型引入无线传感器网络数据聚合调度中



在 SIC 模型下，链路 l_v 成功传输的条件是(a)对于每一条满足 $Q(l_u, l_v) > Q(l_v)$, $l_u \in \mathcal{S}_t$ 的链路 l_u :

$$\frac{Q(l_u, l_v)}{\xi + \sum_{\substack{l_w \in \mathcal{S}_t, l_w \neq l_v \\ Q(l_w, l_v) \leq Q(l_u, l_v)}} Q(l_w, l_v)} \geq \beta \quad (4.3.2)$$

同时(b)对于链路 l_v , 满足:

$$\frac{Q(l_v)}{\xi + \sum_{\substack{l_u \in \mathcal{S}_t, l_u \neq l_v \\ Q(l_u, l_v) \leq Q(l_v)}} Q(l_u, l_v)} \geq \beta \quad (4.3.3)$$

定理 4.1: 问题 **MLDAS-SIC** 是 **NP**-难的

定义链路 l_u 对链路 l_v 造成的相对干扰(Relative Interference, RI)为:

$$RI(l_u, l_v) = \begin{cases} 0 & l_u = l_v \\ \infty & d(l_u, l_v) = 0 \ \&\& l_u \neq l_v \\ c(l_v) \cdot \frac{d^\alpha(l_v)}{d^\alpha(l_u, l_v)} & \text{others} \end{cases}$$

引理 4.2: 物理干扰模型下的 SINR 解码条件(4.2.1)等价于 $RI(\mathcal{S}_t, l_v) \leq 1$ 。

引理 4.3: 在 SIC 模型下, 链路 l_v 传输的充要条件((4.2.2)和(4.2.3))等价于: 任给

$l_u \in ALS(\mathcal{S}_t, l_v) \cup \{l_v\}$, 有: $RI(l_u, l_v) \geq \frac{\beta}{\beta+1} \cdot RI(\mathcal{S}_t \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) + \frac{c(l)-\beta}{\beta+1}$ 。

RFGS: Random-link First Greedy Scheduling

Algorithm :基于随机链路优先的贪心调度算法(**RFGS**)

输入: 树 $G(\mathcal{V}, \mathcal{E})$, 叶节点集合 \mathcal{V}_L

输出: 调度序列 $\{\mathcal{S}_t\}$, $t = 1, 2, \dots$, \mathcal{S}_t 包含 t 时隙调度的链路集合

```

1:  $t \leftarrow 1, \mathcal{S}_t \leftarrow \emptyset, \mathcal{M} \leftarrow \emptyset;$ 
2:  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\};$ 
3: while  $\mathcal{V} \neq \{v_s\}$  do
4:   从集合  $\mathcal{L}$  中随机选取一条链路  $l_v$ , 并将其移动到  $\mathcal{S}_t$  中;
5:   for 集合  $\mathcal{L} \setminus \mathcal{M}$  中的每一条链路  $l_u$  do
6:     if RFGS-IC( $l_u$ )  $\neq$  FALSE then
7:        $\mathcal{M} \leftarrow \mathcal{M} \cup \{l_u\};$ 
8:     else
9:        $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{l_u\}, \mathcal{L} \leftarrow \mathcal{L} \setminus \{l_u\};$ 
10:    end if
11:   end for
12:    $\mathcal{E} \leftarrow \mathcal{E} \setminus \mathcal{S}_t$ , 更新  $\mathcal{V}$ 、 $\mathcal{V}_L$ ,  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\}$ ,  $\mathcal{M} \leftarrow \emptyset, t \leftarrow t + 1$ ;
13: end while

```

▷ 干扰判定子算法(**RFGS-IC**)

输入: 调度链路集合 \mathcal{S} , 待调度链路 l_u

输出: 如果链路 l_u 破坏集合 \mathcal{S} 中的干扰规则, 则返回TRUE; 否则, 返回FALSE;

```

1: for 集合  $\mathcal{S} \cup \{l_u\}$  中的每条链路  $l_w$  do
2:   for 集合  $ALS(\mathcal{S}, l_w) \cup \{l_w\}$  中的每条链路  $l_y$  do
3:     if  $RI(l_y, l_w) < \frac{\beta}{\beta+1} \cdot RI(\mathcal{S} \setminus ALS(\mathcal{S}, l_w, l_y), l_u) + \frac{c(l_w)-\beta}{\beta+1}$  then
4:       return TRUE;
5:     end if
6:   end for
7: end for
8: return FALSE;

```

在每个时隙, 采用首次适合(**First-Fit**)策略, 随机选择一条链路进行调度。然后基于贪心法, 从候选的待调度链路集合中选取一个可行的极大(**Maximal**)链路子集进行调度。

$$\mathcal{O}(|\mathcal{E}|^4)$$

定理 4.4: 在 SIC 模型下, 算法 RFGS 产生的调度是可行的。

SFGS: Shortest-link First Greedy Scheduling

Algorithm :基于最短链路优先的贪心调度算法(SFGS)

输入: 树 $G(\mathcal{V}, \mathcal{E})$, 叶节点集合 \mathcal{V}_L

输出: 调度序列 $\{\mathcal{S}_t\}$, $t = 1, 2, \dots$, \mathcal{S}_t 包含 t 时隙调度的链路集合

```

1:  $t \leftarrow 1, \mathcal{S}_t \leftarrow \emptyset, \mathcal{M} \leftarrow \emptyset;$ 
2:  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\};$ 
3: while  $\mathcal{V} \neq \{v_s\}$  do
4:   while  $\mathcal{L} \setminus \mathcal{M} \neq \emptyset$  do
5:     从集合  $\mathcal{L}$  中选取一条长度最短的链路  $l_v$ , 并将其移动到  $\mathcal{S}_t$  中;
6:     for 集合  $\mathcal{L} \setminus \mathcal{M}$  中的每一条链路  $l_u$  do
7:       if SFGS-IC( $l_u$ ) == TRUE then
8:          $\mathcal{M} \leftarrow \mathcal{M} \cup \{l_u\};$ 
9:       end if
10:      end for
11:    end while
12:     $\mathcal{E} \leftarrow \mathcal{E} \setminus \mathcal{S}_t$ , 更新  $\mathcal{V}$ 、 $\mathcal{V}_L$ ,  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\}$ ,  $\mathcal{M} \leftarrow \emptyset$ ,  $t \leftarrow t + 1$ ;
13: end while

```

▷ 干扰判定子算法(SFGS-IC)

输入: 调度链路集合 \mathcal{S} , 待调度链路 l_u , 最短链路 l_v

输出: 如果链路 l_u 破坏集合 \mathcal{S} 中的干扰规则, 则返回TRUE; 否则, 返回FALSE;

```

1:  $\lambda \leftarrow \frac{2^{\alpha}-2}{72 \cdot \beta \cdot 2^\alpha};$ 
2: if  $RI(l_u, l_v) \geq \lambda \cdot c(l_v)$  then
3:   return TRUE
4: else
5:   for 集合  $ALS(\mathcal{S}, l_u) \cup \{l_u\}$  中的每一条链路  $l_w$  do
6:     if  $RI(l_w, l_u) < \frac{3\beta}{2+3\beta} \cdot RI(\mathcal{S} \setminus ALS(\mathcal{S}, l_u, l_w), l_u) + \frac{3c(l_u)-3\beta}{3+3\beta}$  then
7:       return TRUE
8:     end if
9:   end for
10: end if
11: return FALSE

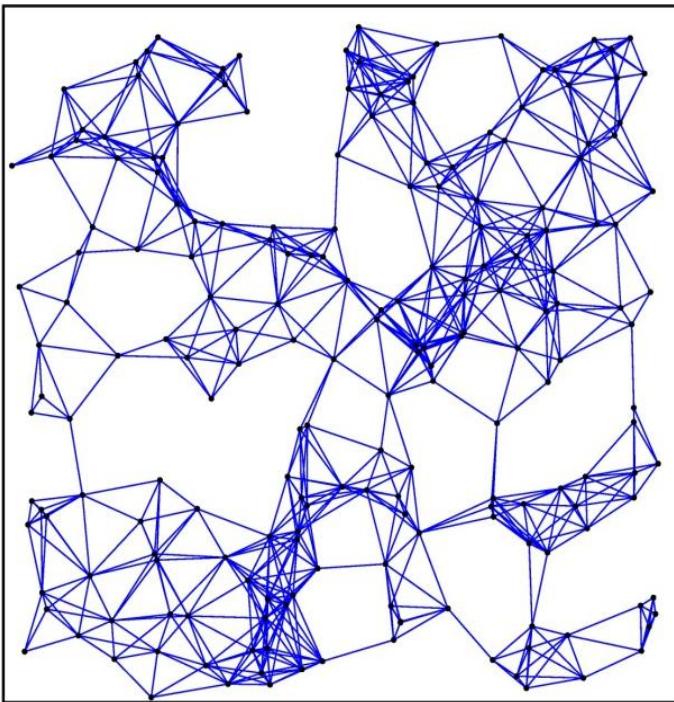
```

与 **RFGS** 随机地选择链路进行调度不同, **SFGS** 算法总是从当前候选链路集合中选择长度最短的链路进行调度。

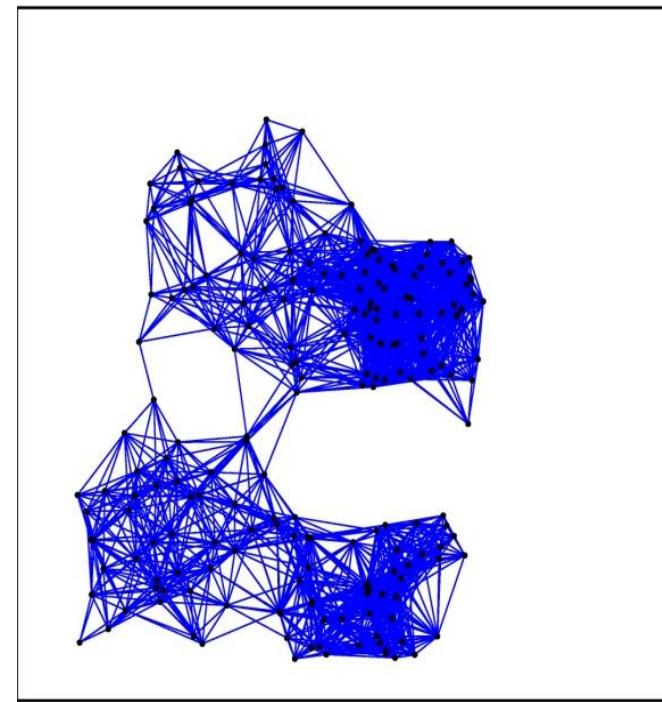
$$\mathcal{O}(|\mathcal{E}|^4)$$

定理 4.5: 在 SIC 模型下, 算法 SFGS 产生的调度是可行的

实验设置



均匀随机网络



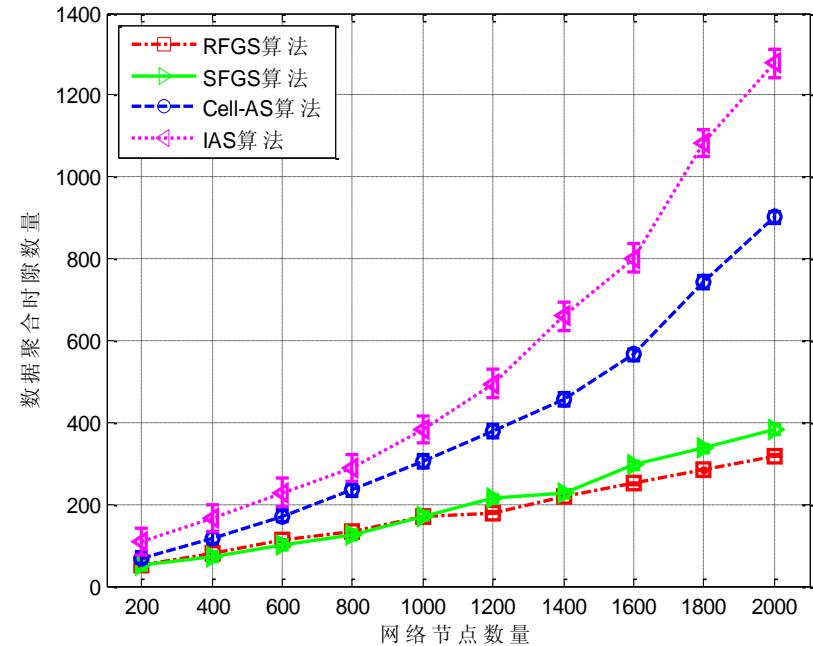
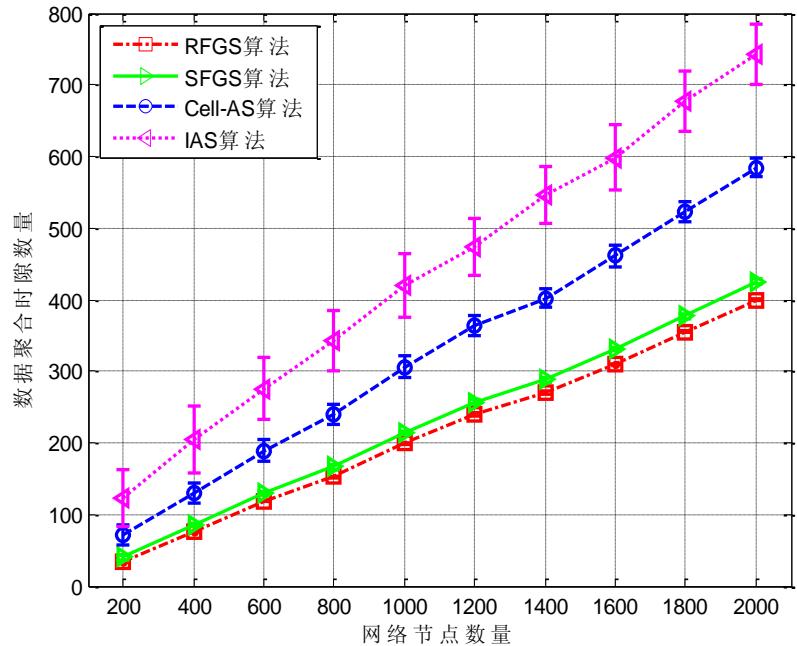
分簇网络

对比算法: **RFGS** **SFGS** **IAS^[3]** **Cell-AS^[4]**

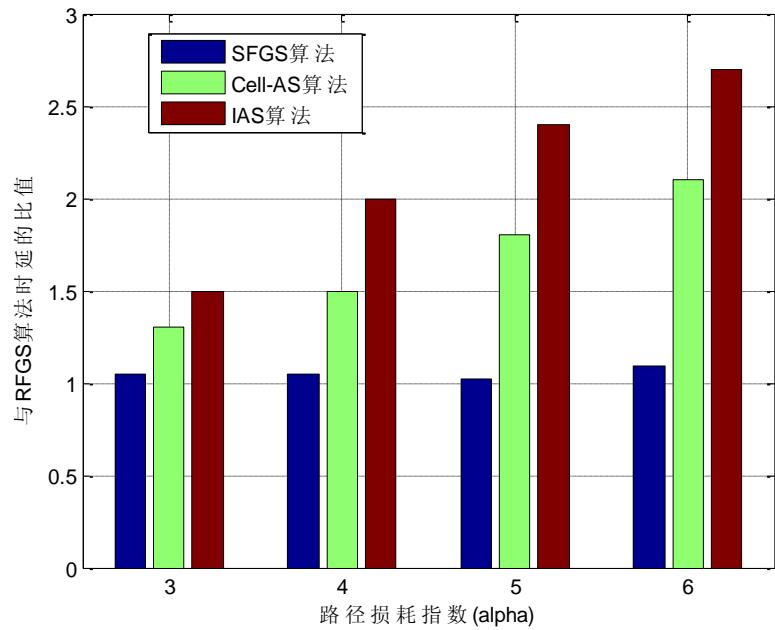
[3] X. Xu, X.-Y. Li, and M. Song, “Efficient aggregation scheduling in multihop wireless sensor networks with sinr constraints,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 12, pp. 2518–2528, 2013.

[4] H. Li, C. Wu, Q.-S. Hua, and F. Lau, “Latency-minimizing data aggregation in wireless sensor networks under physical interference model,” *Ad Hoc Networks*, vol. 23, no. 2, pp. 123–135, 2011.

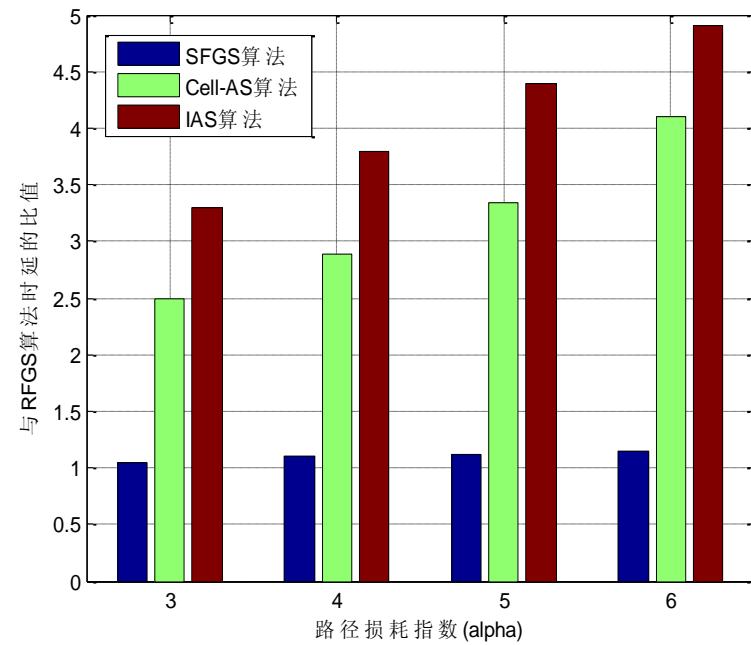
实验结果(1): 时延 vs. 节点数量



实验结果(2): 时延 vs. 路径损耗指数

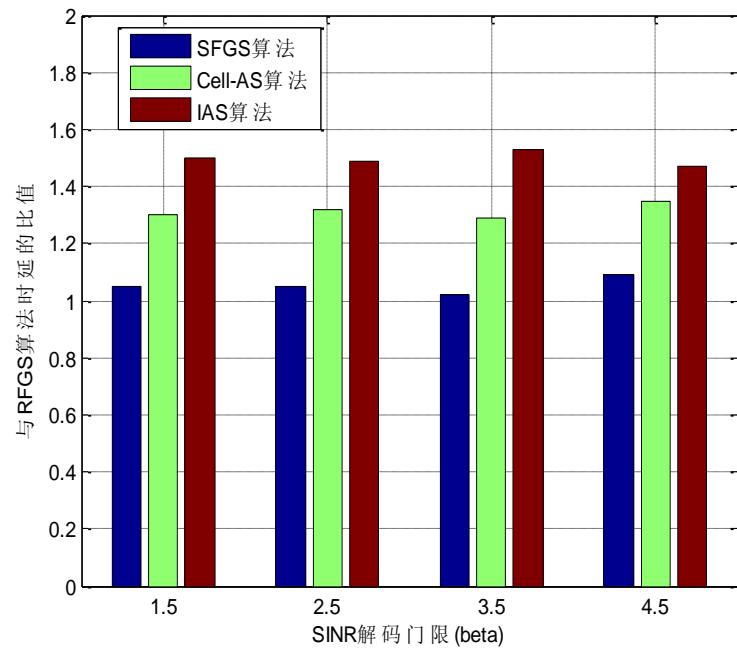


均匀网络

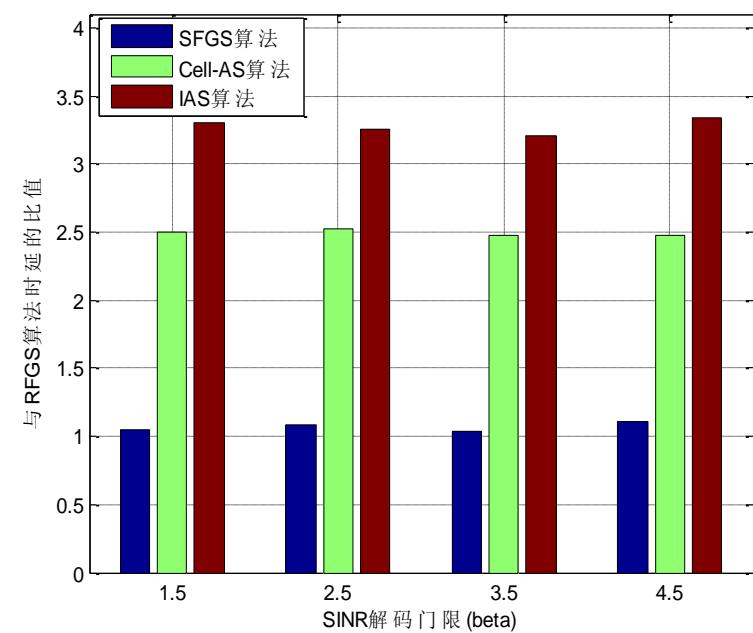


分簇网络

实验结果(3): 时延 vs. SINR解码门限

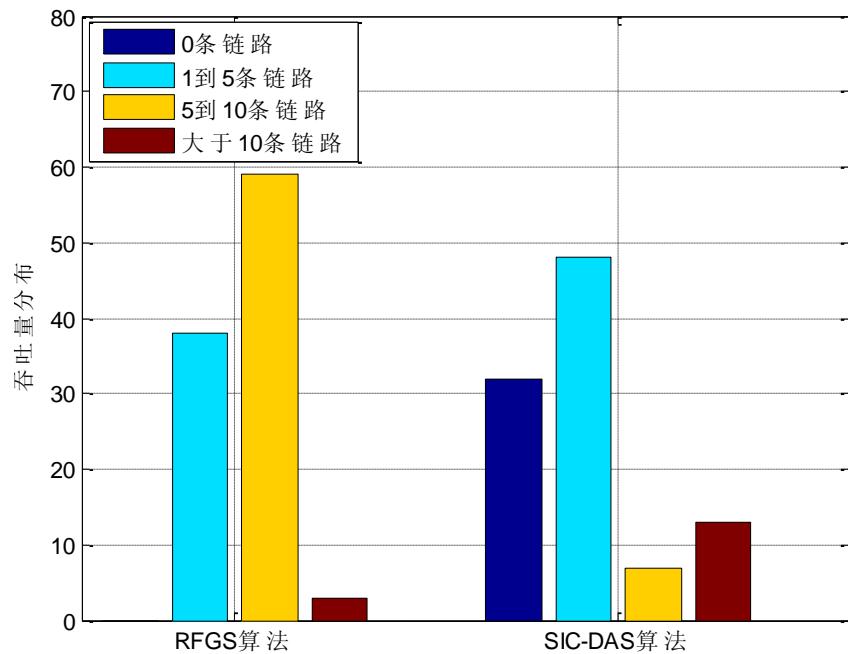
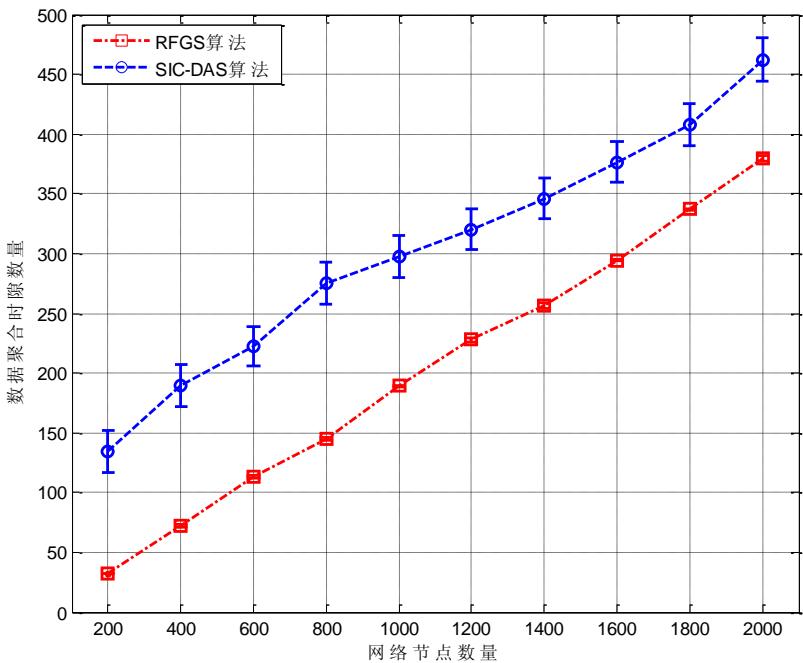


均匀网络



分簇网络

实验结果(4): 与文献^[5]对比



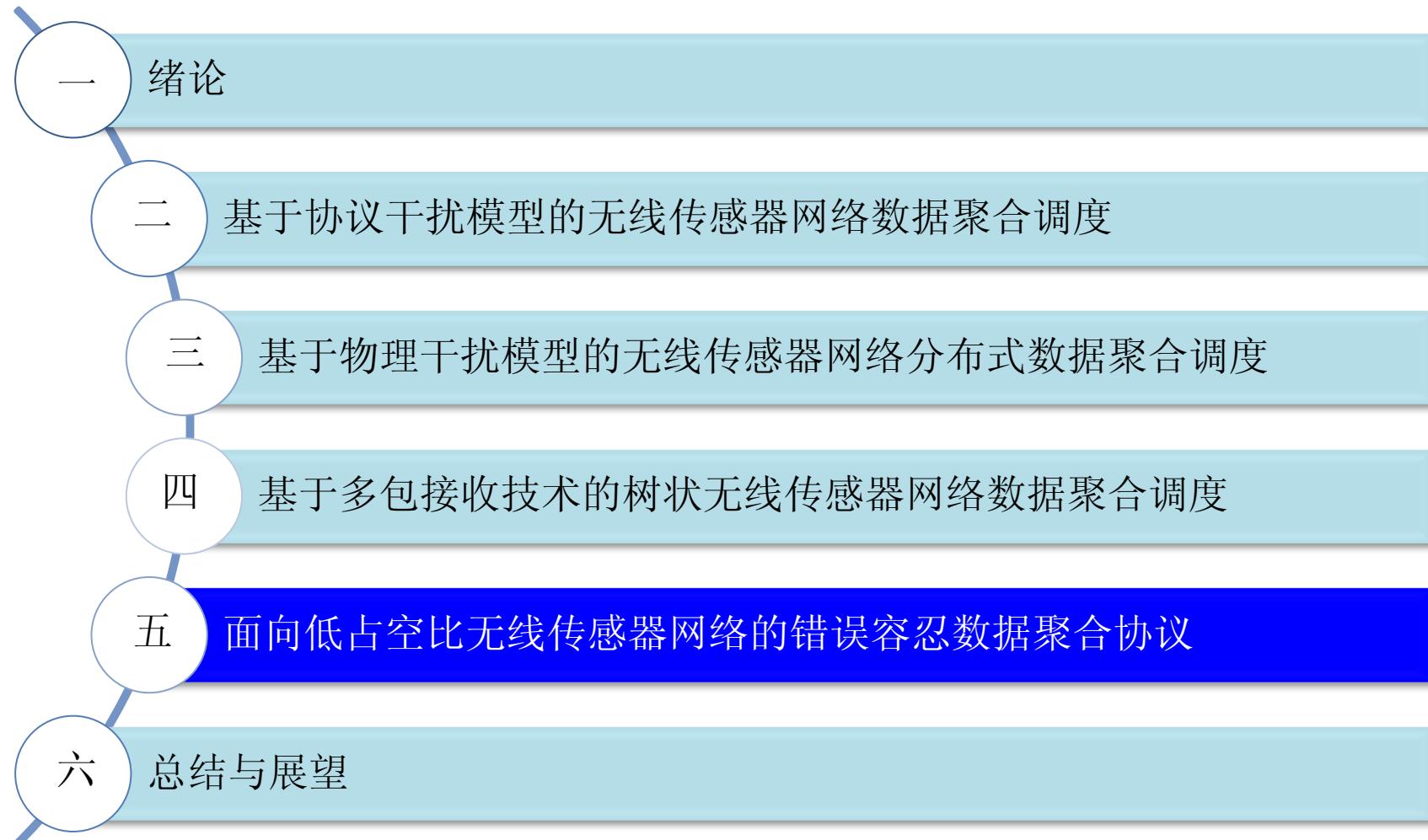
SIC-DAS^[5]

[5] H. Li, C. Wu, D. Yu, Q.-S. Hua, and F. C. Lau, “Aggregation latency-energy tradeoff in wireless sensor networks with successive interference cancellation,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 11, pp. 2160–2170, 2013.

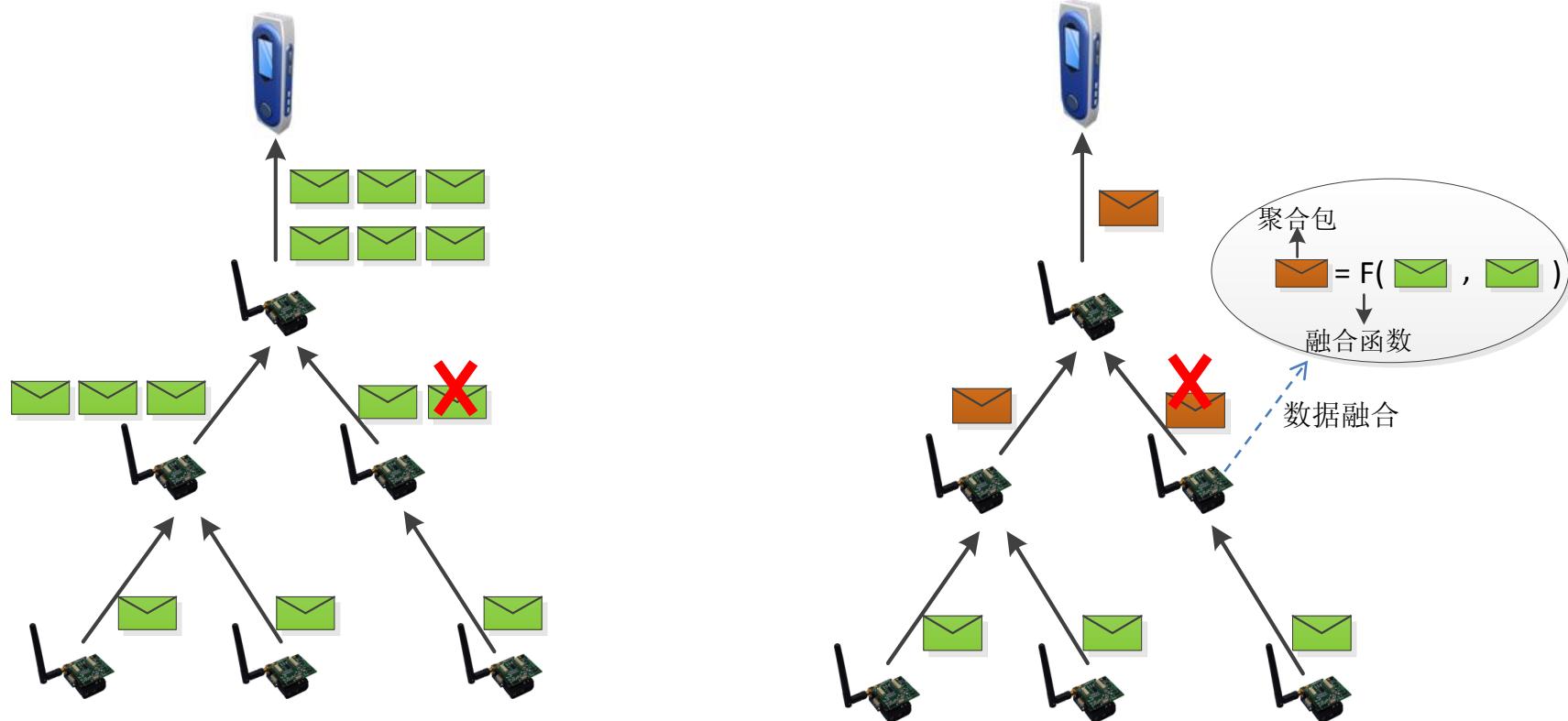
基于主动的干扰消除理念，将物理层多包接收模型引入了无线传感器网络数据聚合中，并研究了该模型下的聚合调度问题，取得了以下创新点：

1. 分别基于随机链路优先策略和最短链路优先策略，提出了两种时延高效的启发式聚合调度算法**RFGS**和**SFGS**，以充分利用干扰消除和多包接收能力所带来的链路并发传输机会。
2. 论证了两种调度算法的可行性，证明了两种调度算法具有多项式时间计算复杂度。
3. 仿真实验结果表明，相比传统的基于单包接收模型的调度方法，本文算法能够显著地降低数据聚合平均时延，提高网络吞吐量。

本章研究工作揭示了干扰消除技术在调度算法设计时的潜力。也就是说，当物理层具有干扰消除能力时，通过精心设计相应的MAC层与网络层算法/协议，无线网络的性能有望得到显著提升。

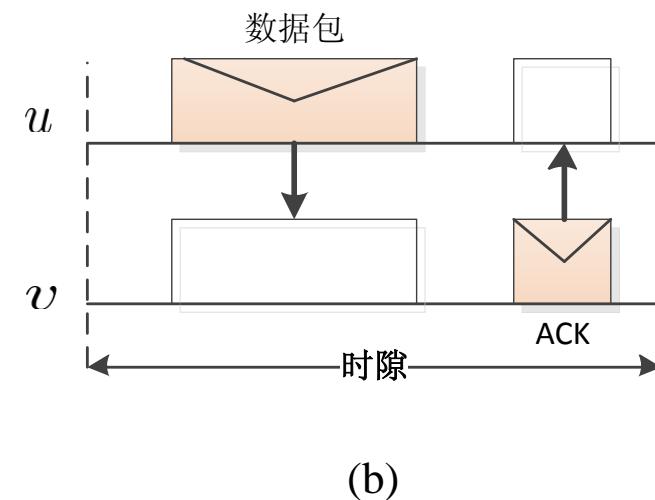
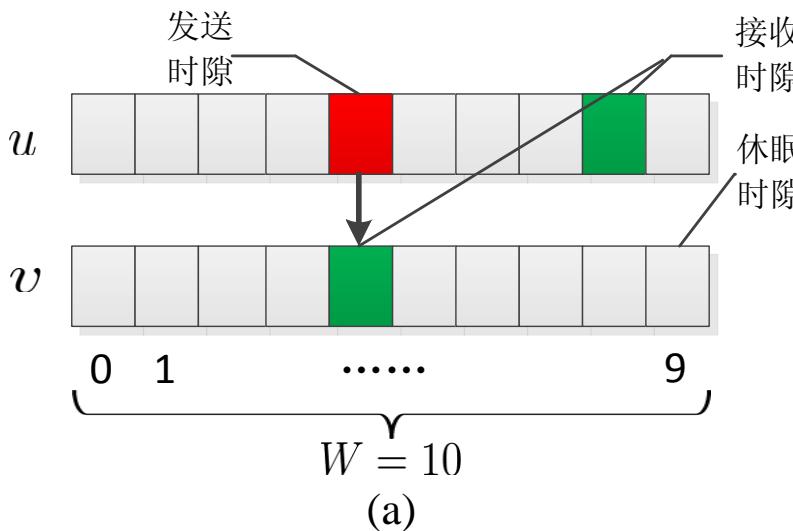


- 当网络中出现由传输错误导致的**数据包丢失**时，传统的数据聚合调度协议的性能将急剧降低——此时，sink节点收到的聚合包中的**有效信息量**大大减少
- 为了克服网络丢包对数据聚合性能的损害，迫切需要设计一种错误容忍的数据聚合协议，以改善链路传输错误(丢包)情况下的**数据聚合信息质量**



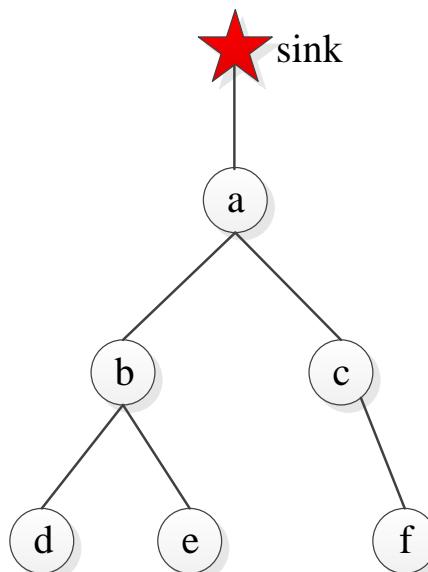
数据聚合中，丢包给网络性能造成的损失更为严重

低占空比(low-duty-cycle)网络



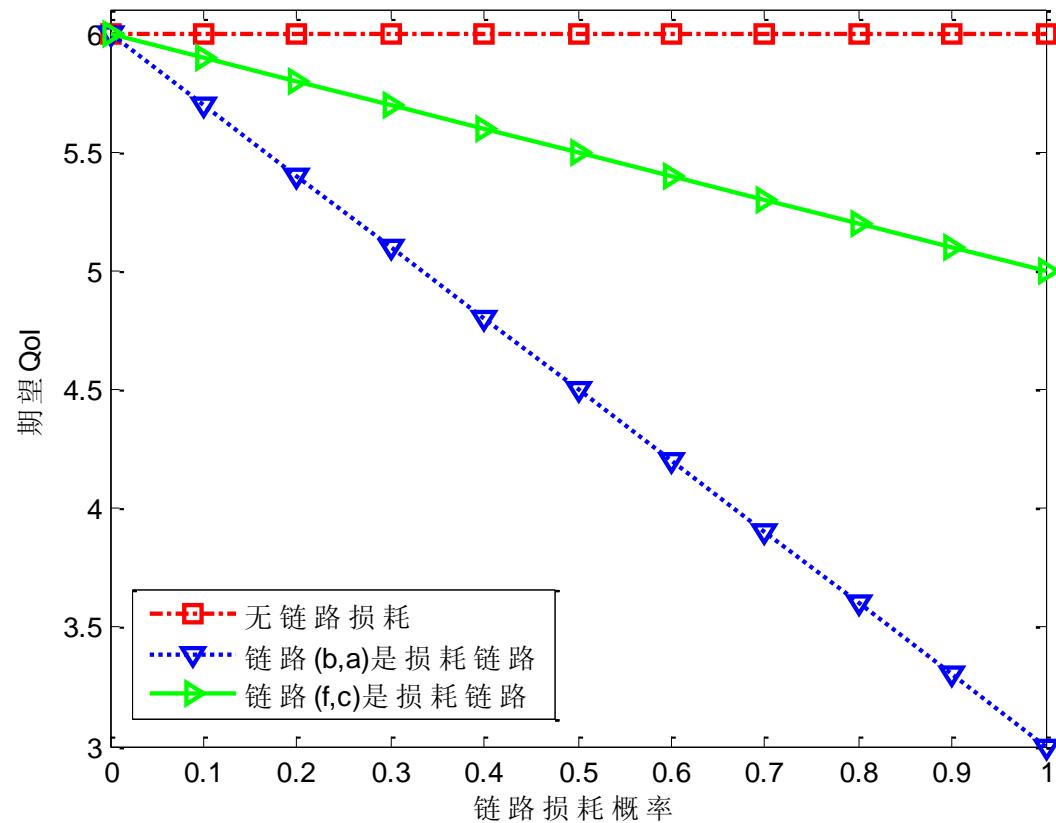
- 睡眠调度(sleep scheduling): 决定节点在每个时隙是处于接收状态(receiving state)还是休眠状态(dominant state)
- MAC调度(MAC scheduling): 决定节点何时进行发送数据(即处于发送状态 transmitting state); 由于睡眠调度的约束, 节点只有在邻居节点处于接收状态时才能向其发送数据。

链路可靠性对QoI的影响

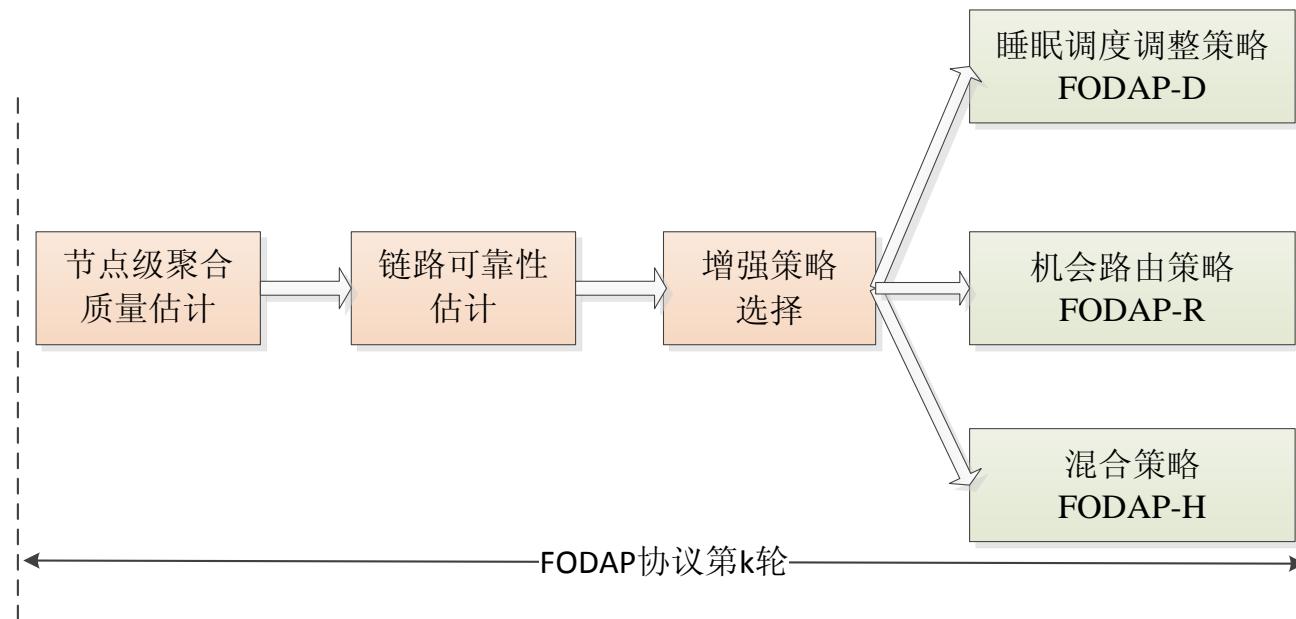


数据聚合信息质量(Quality-of-Information, QoI)

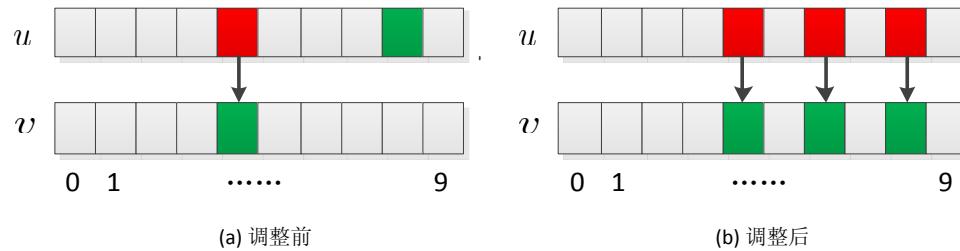
$$\mathbb{E}(QoI) = \sum_{v \in \mathcal{V} \setminus \{v_s\}} \prod_{u \in PATH_v} (1 - \rho(u, P_u)).$$



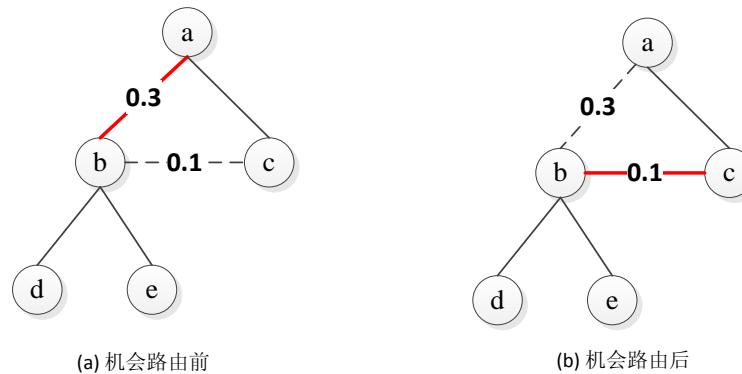
- 设计传输错误(transmission failure)容忍的数据聚合协议，其采用一定的链路可靠性增强机制，来提高聚合包的传输成功概率，从而提高无线传感器网络数据聚合的信息质量QoI。



□ 动态睡眠调度调整(Dynamic Sleep Scheduling Adjustment, DSSA)策略



□ 机会路由(Opportunistic Routing, OR)策略

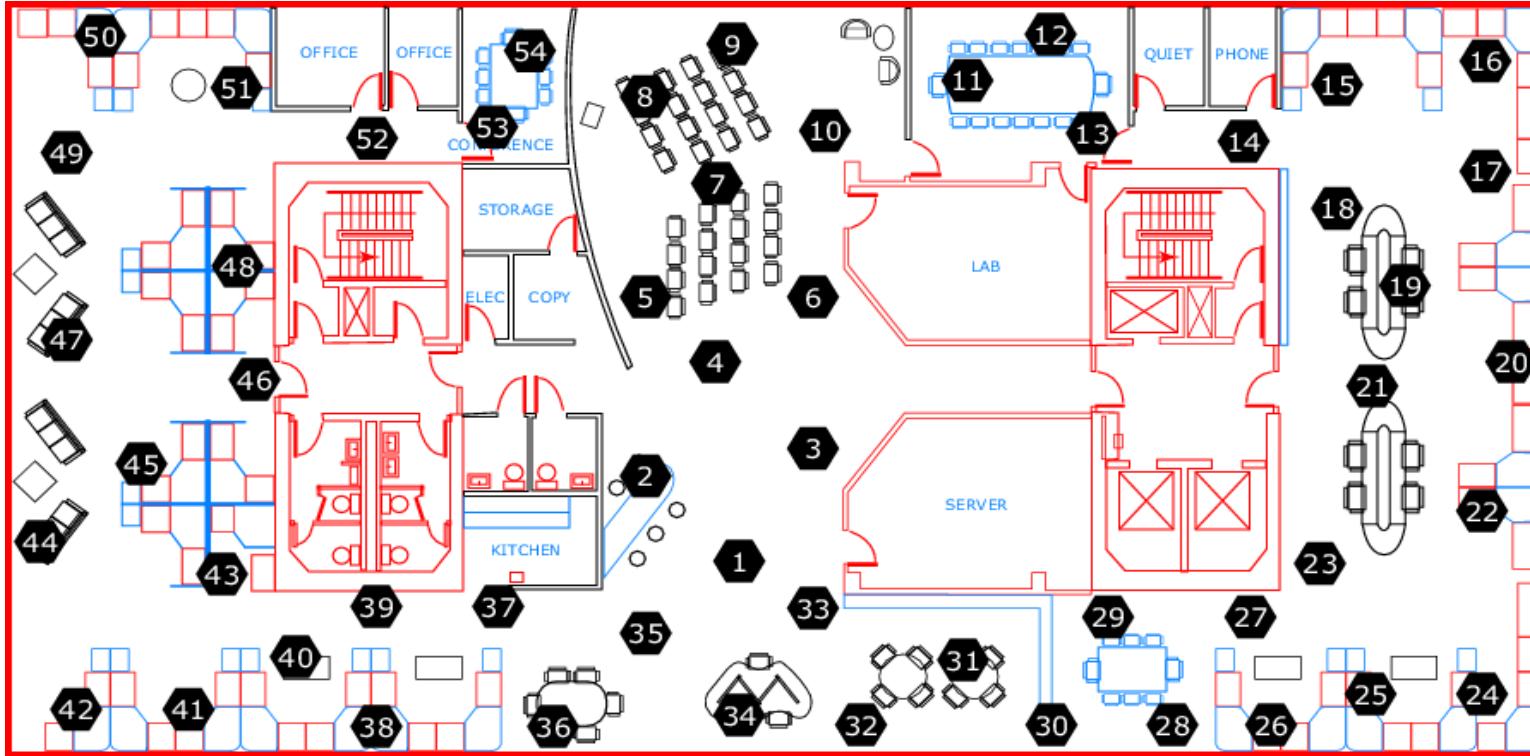


$$P_v = \arg \max_{u \in \mathcal{F}_v} \rho(v, u)$$

□ 混合(hybrid, HYB)策略

将采用了动态睡眠周期调整策略、机会路由策略和混合策略的**FODAP**分别表示为**FODAP-D**, **FODAP-R**和**FODAP-H**。

实验设置

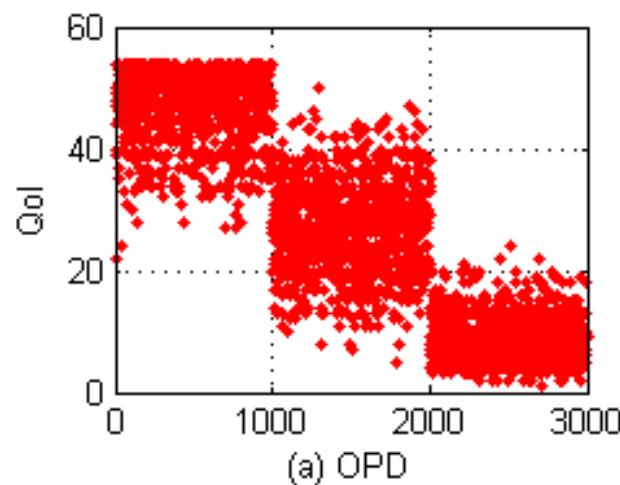


Intel berkeley research lab data: <http://berkeley.intel-research.net/labdata.html>

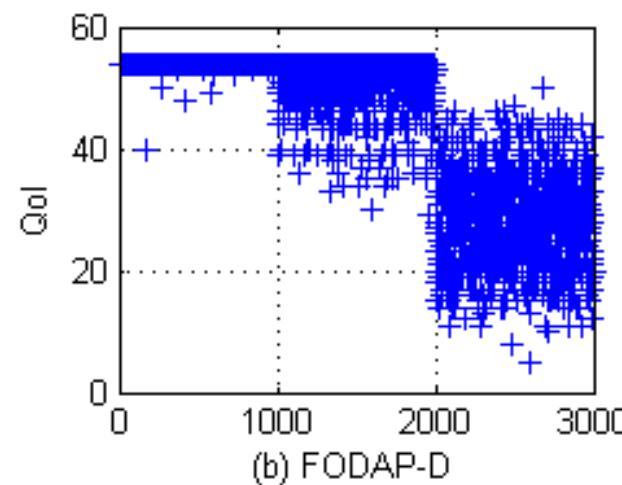
- 1号节点为根节点，最短路径树(shorest path tree, SPT)作为数据聚合的路由
- 统计成功参与数据聚合过程的节点数量作为数据聚合的信息质量QoI
- 与文献中的**OPD**协议进行性能对比。**OPD**协议没有考虑链路损耗，也就是说，无论传输成功与否，每个数据包均传输且只传输一次。

W. Qu, K. Li, et al, “An efficient method for improving data collection precision in lifetime-adaptive wireless sensor networks,” *ICC’07*, 2007, pp. 3161–3166.

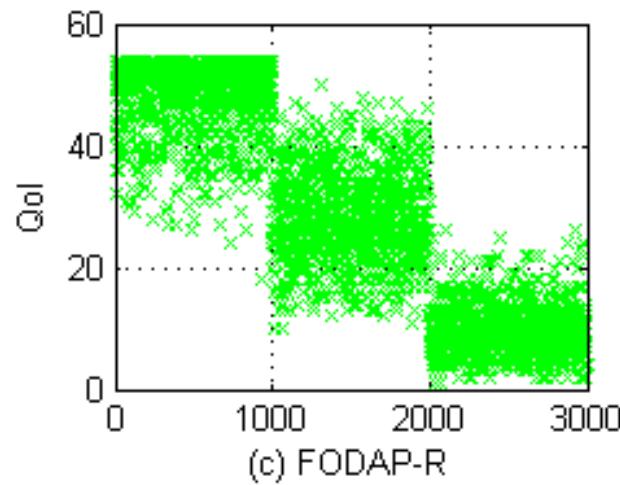
实验结果(1): 3000轮数据聚合快照(snapshot)图



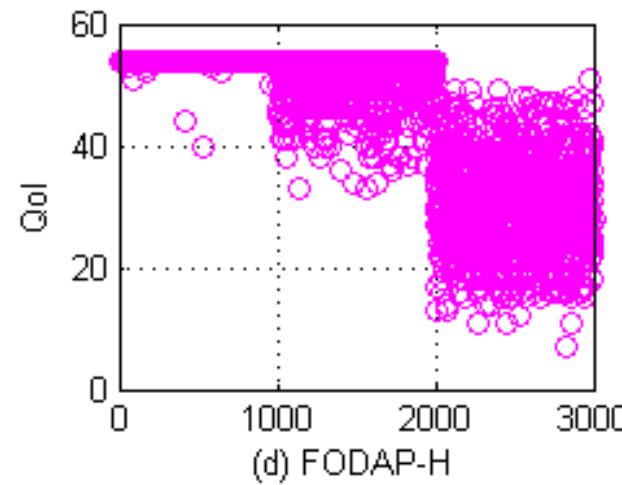
(a) OPD



(b) FODAP-D

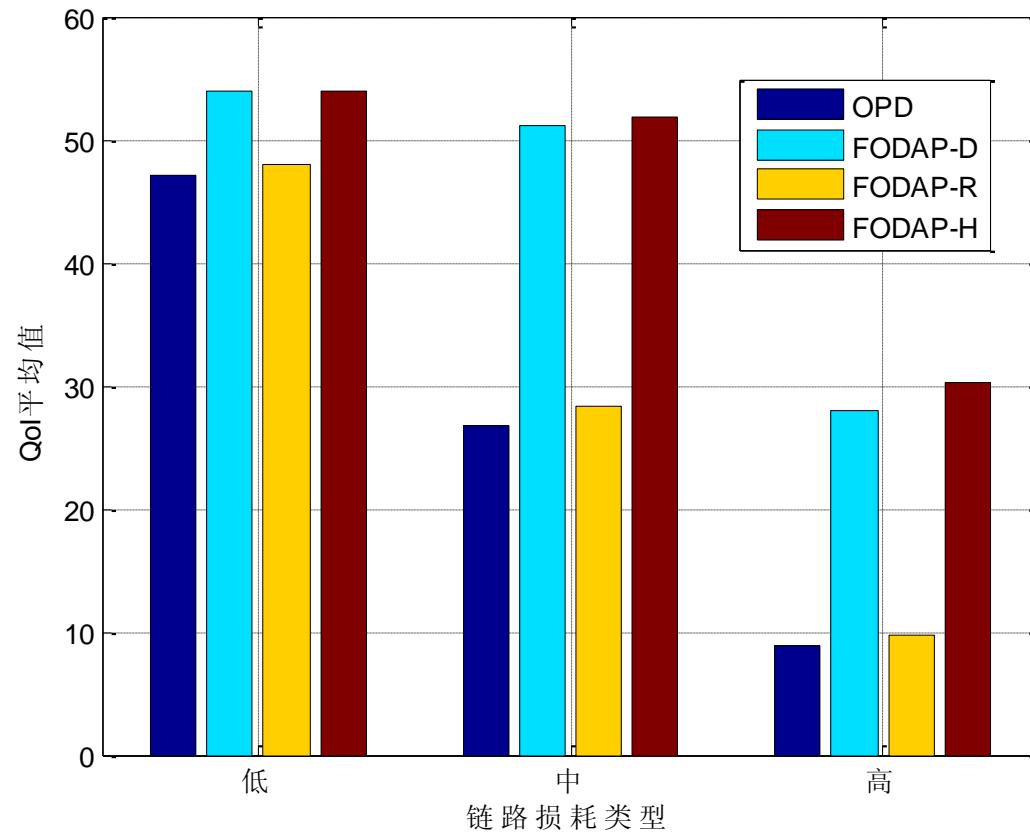


(c) FODAP-R

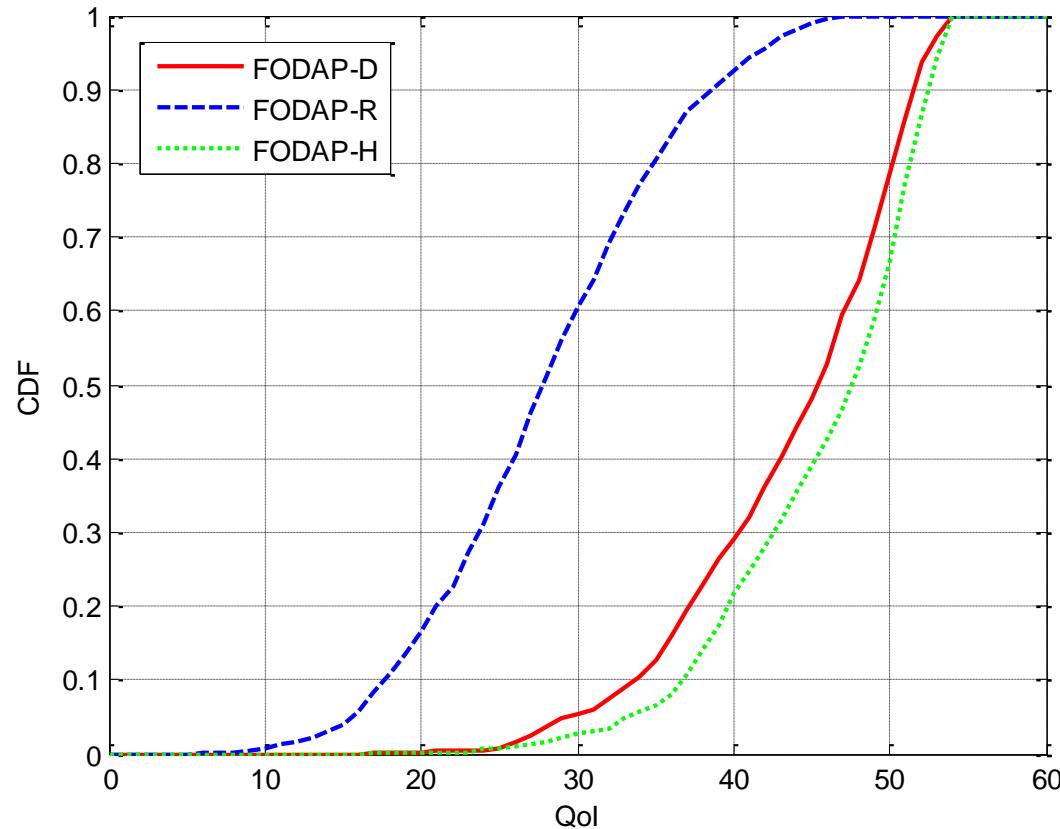


(d) FODAP-H

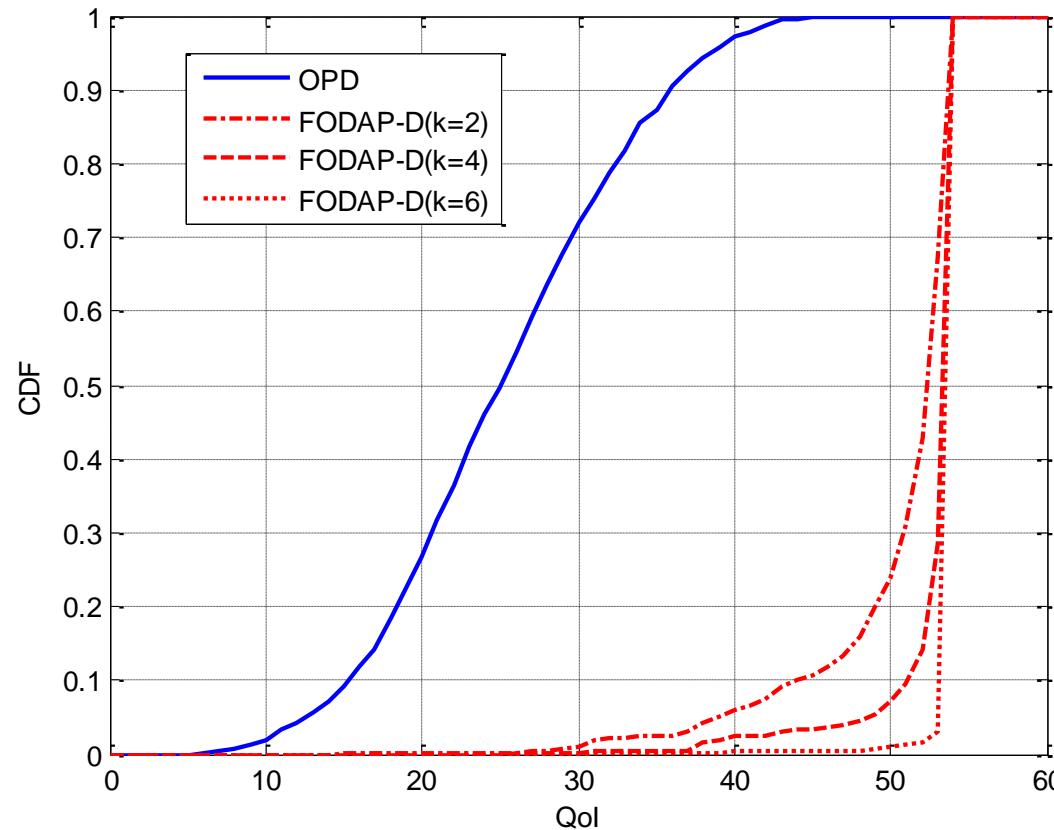
实验结果(2): 四种协议的平均QoI



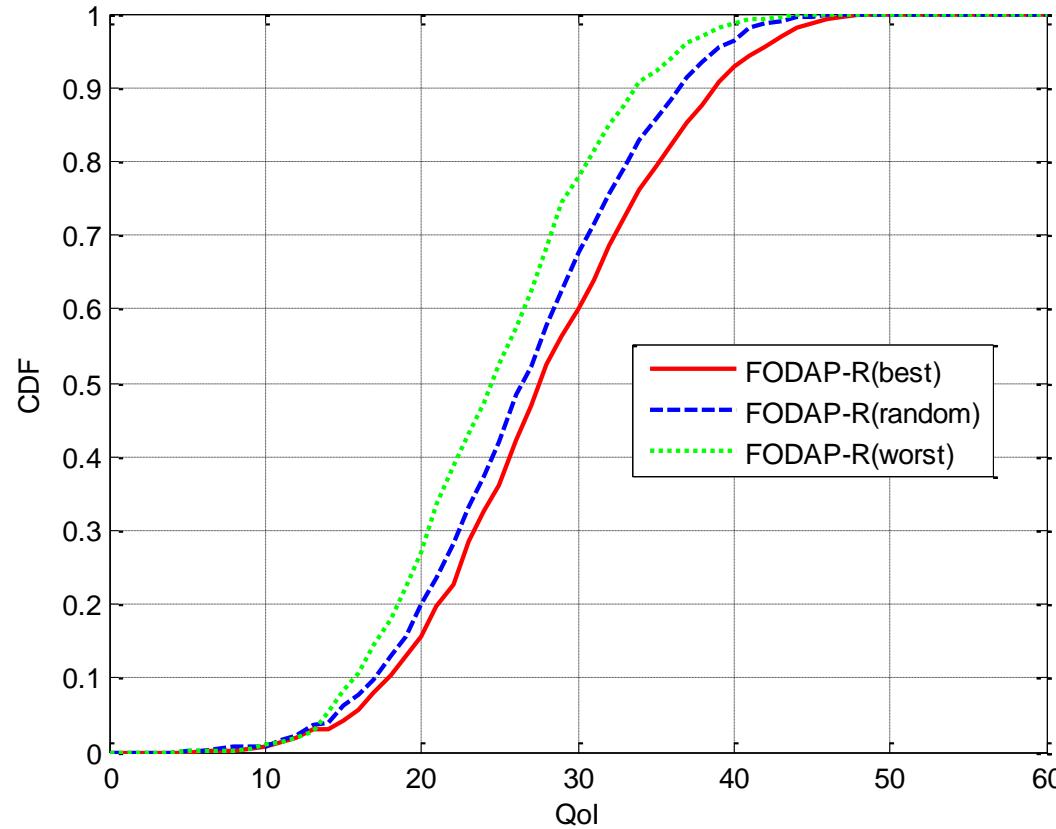
实验结果(3): 三种协议的累积分布CDF对比



实验结果(4): FODAP-D QoI vs. 最大重传次数

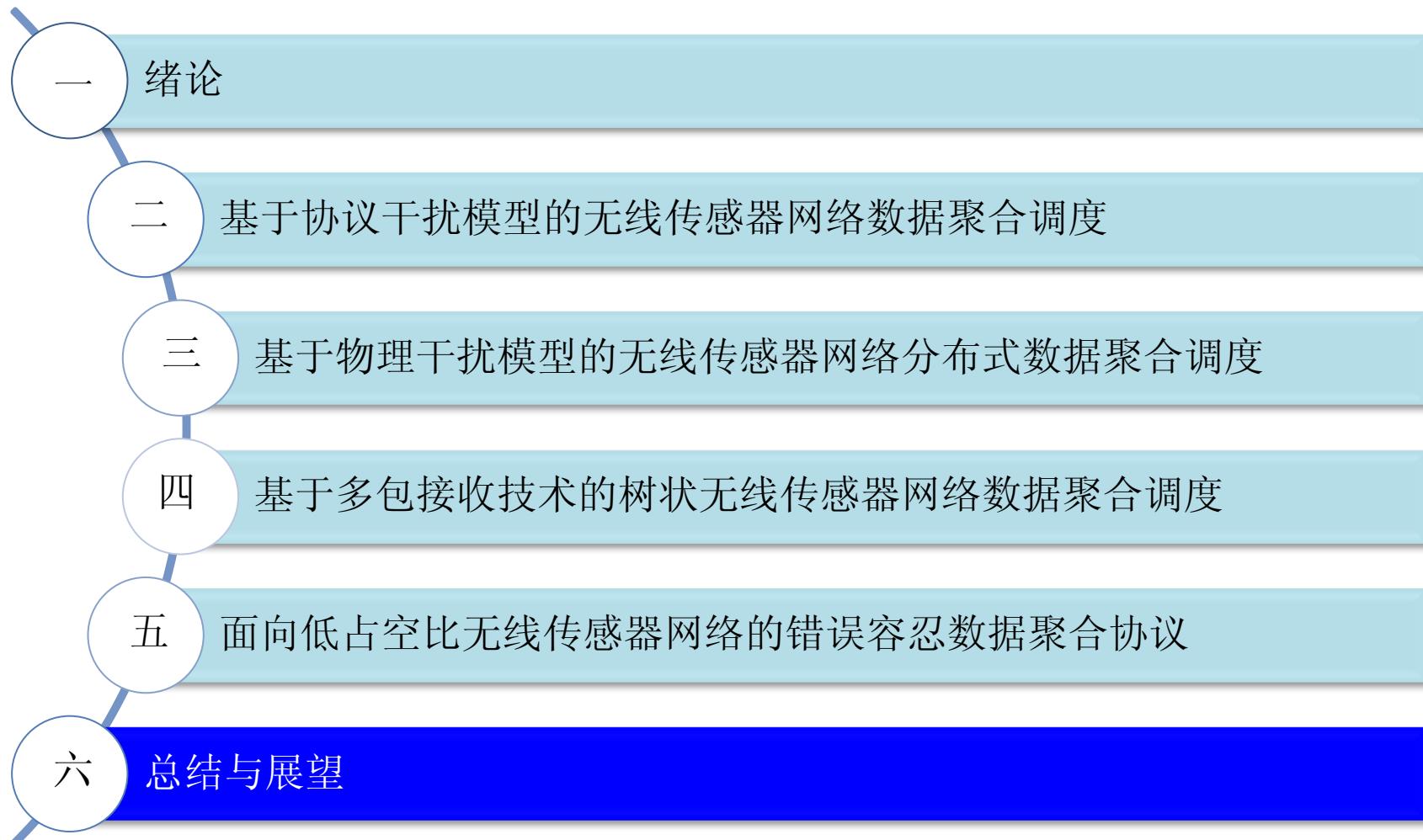


实验结果(5): FODAP-R QoI vs. 路径选择策略

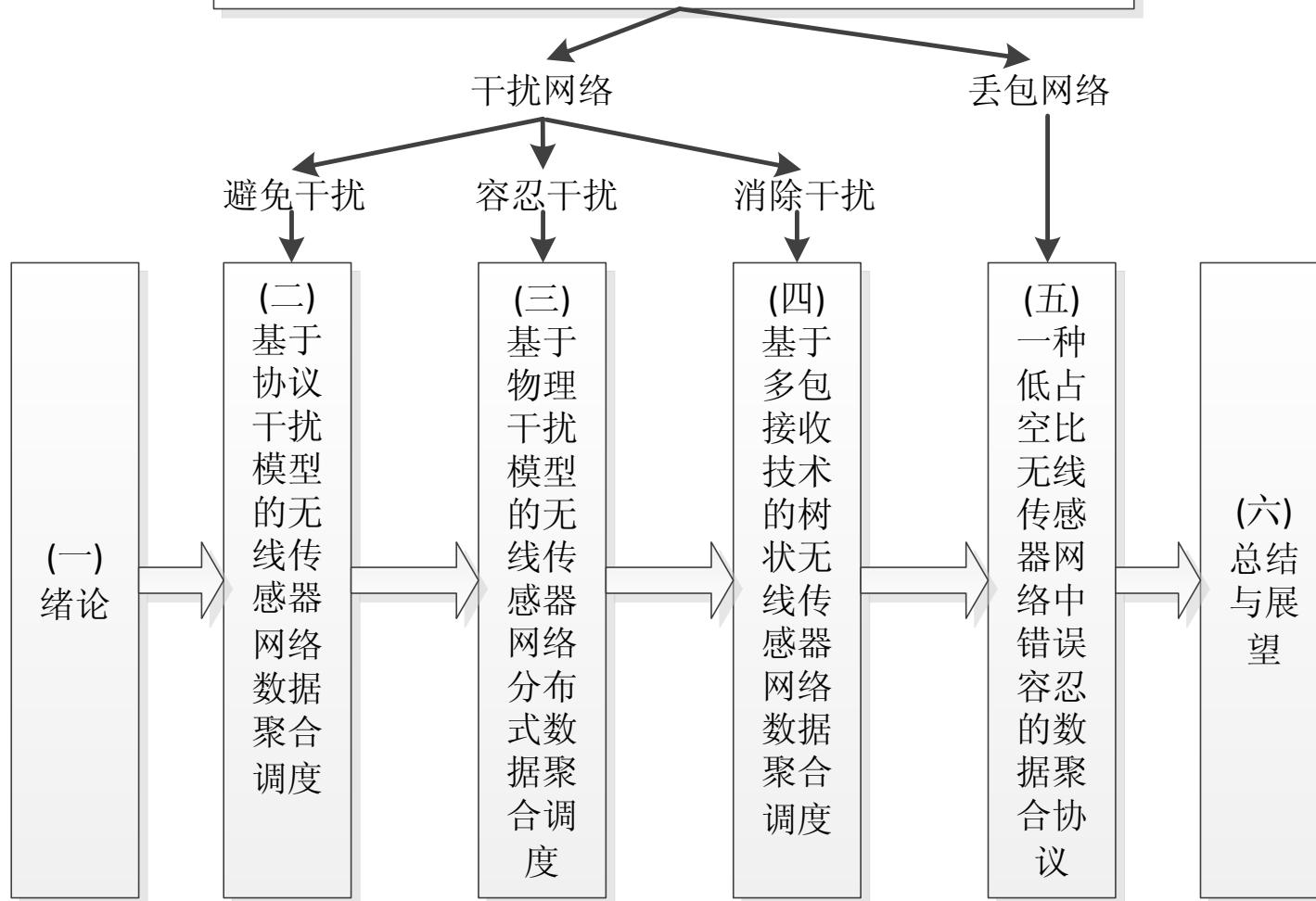


本章研究了低占空比且链路不可靠无线传感器网络中的数据聚合问题，取得了以下创新点：

1. 提出了一种错误容忍的数据聚合协议**FODAP**，以改善链路损耗状况下的数据聚合信息质量。依据采用的链路可靠性增强技术的不同，我们分别基于动态睡眠调度调整策略，机会路由策略和混合策略，为该协议设计了三个变种，即：**FODAP-D**, **FODAP-R**和**FODAP-H**。
2. 仿真实验表明，在链路不可靠的网络场景下，相比文献中传统的未采用链路可靠性增强技术的数据聚合协议，上述三种协议均能够不同程度地提升数据聚合过程中汇聚节点所接收聚合包的信息质量。其中，基于混合策略的**FODAP-H**较另外两种协议对QoI性能的提升更为显著。



面向数据聚合的无线传感器网络传输调度方法研究



- 进一步改善已有数据聚合调度方法的理论上界
- 更为复杂的干扰模型下的数据聚合调度问题
- 低复杂度的分布式数据聚合调度方法
- 满足多种性能指标的数据聚合调度方法

一作论文：

- **Shiliang Xiao***, Baoqing Li, and Xiaobing Yuan. “Maximizing Precision for Energy-Efficient Data Aggregation in Wireless Sensor Networks with Lossy Links”, **Ad Hoc Networks**, vol. 26, no. 3, pp. 103-113, 2015 (SCI, IF=1.943)
- **Shiliang Xiao**, Lebing Pan, Yunzhou Qiu, and Xiaobing Yuan. “Minimum Latency Aggregation Scheduling in Wireless Sensor Networks with Successive Interference Cancellation”, **Ad Hoc & Sensor Wireless Networks**, vol. 26, no. 1, pp. 233-257, 2015 (SCI, IF=0.478)
- **Shiliang Xiao**, Jingchang Huang, et al. “On centralized and distributed algorithms for minimizing data aggregation time in duty-cycled wireless sensor networks”, **ACM/Springer Wireless Networks**, vol. 20, no. 7, pp. 1729-1741, 2014 (SCI, IF=1.055)
- **Shiliang Xiao**, Jun Pei, et al. “Minimum Latency Broadcast in the SINR model: A Parallel Routing and Scheduling Approach”, **IEEE Communications Letters**, vol. 18, no. 6, pp. 1027-1030, 2014 (SCI, IF=1.463)
- **Shiliang Xiao**, Lebing Pan, Jianpo. Liu, et al. “Distributed Broadcast with Minimum Latency in Asynchronous Wireless Sensor Networks under SINR-based Interference”, **International Journal of Distributed Sensor Networks**, vol. 2013, pp.1-12, 2013 (SCI, IF=0.923)

发明专利：

- 肖世良、程小六、吴光等, “一种面向无线传感器网络数据收集的半结构化传输调度方法” ,
发明专利, CN104301966A

谢谢
欢迎各位专家批评指正！

肖世良 博士生
中科院上海微系统所
shliangxiao@gmail.com