



无线传感器网络

——路由协议与拓扑控制

重庆邮电大学

主要内容

- WSN路由协议概述
- 按需发现路由协议
- 基于簇的路由协议
- 以数据为中心的路由协议
- 能量感知路由协议
- 地址位置路由协议
- WSN路由协议的发展

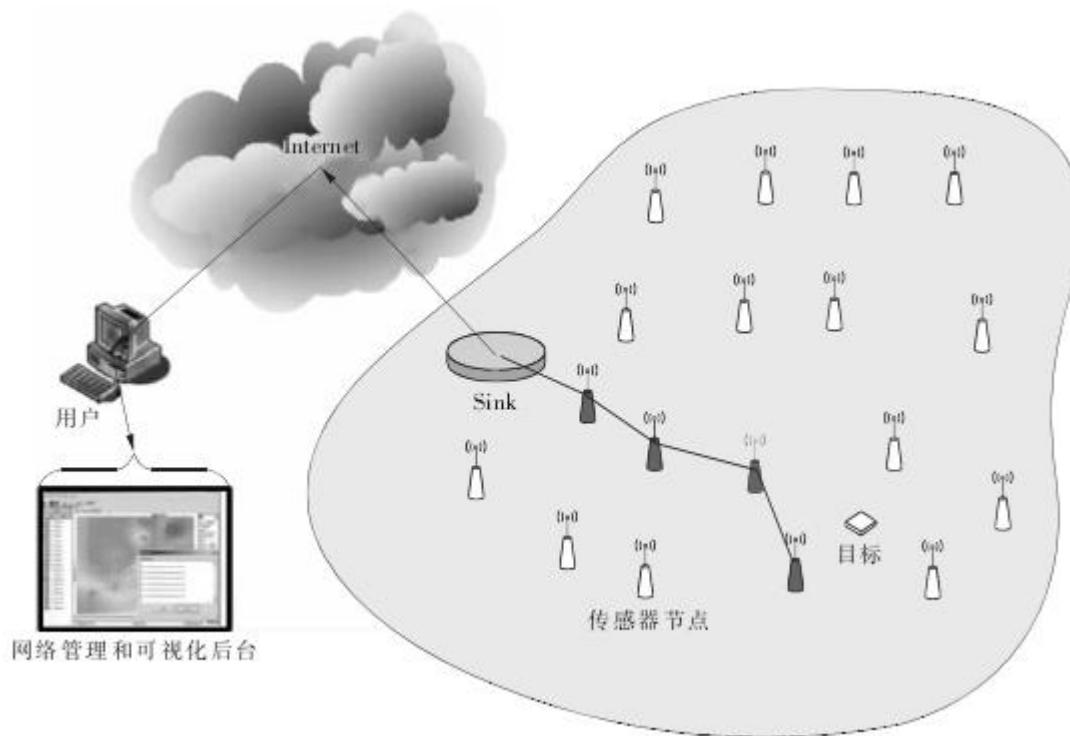


WSN路由协议概述

WSN路由用来做什么？

- 受制于节点的无线传输距离，无线传感器网络往往以多跳（**hop**）的形式进行部署。
- 在多跳网络中，需要处理中间节点的转发问题。
- 路由协议属于网络层的主要功能之一，其主要目的是将数据分组从源节点通过网络转发到目标节点，它主要包括两个方面的功能：
 - 寻找源节点和目标节点间的优化路径
 - 将数据分组沿着优化路径正确转发

WSN路由功能示例



- 如何找到符合需求的路径？ 一路由机制
- 找到如果找到多条符合需求的路径，按什么准则选择最优路径进行转发？ 一路由算法

路由协议基础知识

- 路由协议的运转需要三类节点：源节点、目标节点、中间转发节点（又称为**路由器**）
- 路由器维护一个路由表，表中显示了到达各目标节点所需传送的下一跳节点。路由器收到分组后，通过查询路由表决定将其下一步转发给哪个节点。
- 路由表的填写与更新由路由协议负责。
- 由于无线传感器网络具有能量受限、应用相关等特征，传统有线、无线网络的路由协议一般不适应于无线传感器网络。



WSN路由协议的特点

- 能量优先
- 基于局部拓扑信息
- 以数据为中心
- 应用相关



WSN路由机制需满足的要求

- 能量高效
- 可扩展性
- 鲁棒性
- 快速收敛性

WSN路由协议的分类

- 按需发现路由协议
- 基于簇的路由协议
- 基于查询的路由协议
- 能量感知路由协议
- 地理位置路由协议

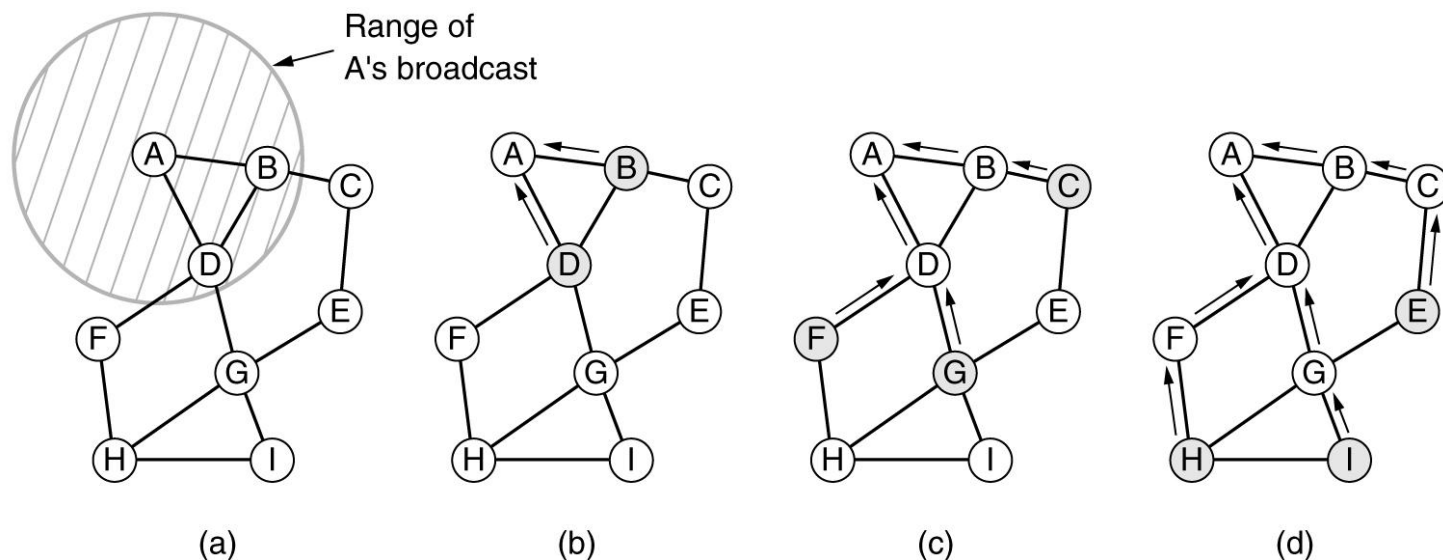


按需发现路由协议 **AODV**协议

AODV路由协议简介（8.1.2节）

- AODV（Ad hoc On-demand Distance Vector）协议
- 适用于拓扑结构动态变化的无线网络（Ad hoc网络）。
- 考虑了速度受限和能量受限的情况。
- 是一种按需分配的路由协议，即只有当有节点想给某一个目标节点发送数据时，才会找出一条通往目标节点的路径。
- 被从Ad hoc网络引入到无线传感器网络，成为无线传感器网络最重要的路由协议之一。

路由发现图示例I



- (a) A在它的辐射范围内广播路由请求帧（ROUTE REQUEST）。
- (b) B和D收到A的广播帧。
- (c) C、F、G收到A的广播帧。
- (d) E、H、I收到A的广播帧。

阴影节点表示新的接收节点，箭头表示反向路由。

关于广播的讨论（1）

- 利用不停的广播来寻找路由的方法，也称为扩散法。（flooding，有的也翻译成泛洪）。
 - 泛洪法非常类似于日常生活中的“谣言”传递过程。
- 思考扩散法路由带来的两个问题：
- 问题1：扩散法会产生大量的重复分组，若不抑制扩散过程，会产生覆盖全网的分组。采用何种方法解决该问题？
- 问题2：在扩散过程中，A广播给B的分组，很有可能在B下一次广播中，再次广播回A，如此反复... ...。如何避免该问题？

关于广播的讨论（2）

- 如何抑制扩散过程在合理范围内？
一般通过“跳数”来控制，声明超过多少跳之后就不再转发。
- 为了抑制扩散，还需要记录下哪些分组已经被扩散过了，以避免再次发送这些分组。
一般通过ID号来控制，如果发现已经转发过该ID的分组，就拒绝掉它。
- 泛洪法在互联网中并不实用，但在无线传感器网络中，网络拓扑经常变化，路由器可能随时停止工作或恢复，为了保证网络的可靠性，泛洪法反而成了一个不错的选择。

源节点的动作

- 源节点发出路由请求帧（ROUTE REQUEST）

源地址	请求ID	目标地址	源序列号	目标序列号	跳计数
-----	------	------	------	-------	-----

路由请求帧（ROUTE REQUEST）

- 请求ID：每当源节点广播一个新的请求时，该ID值加1。源地址和请求ID两个域合起来唯一标识了一个ROUTE REQUEST帧。
- 源序列号：功能类似于A内部的时钟，随着时间其值不断增加。用于区分新的和旧的路由路径。该序列号值越大，说明到该节点的路由越新；最新的值总是存在节点的内存中。
- 目标序列号：A所看到过的最近的I节点的序列号值。如果A之前没有寻找过到达I节点的路由，就置0。
- 跳（hop）计数：记录该帧已经经历了多少跳，其初始值为0。每被转发一次，跳计数值加1。

中间节点的动作

- 当路由请求帧到达某个节点时，该节点将按如下步骤处理：
 - 查询本地历史记录，如果发现已经处理过该帧，则丢弃该帧；否则将一对信息（源地址，请求ID）写入历史记录中（用于识别将来可能重复的帧）。
 - 查询自己路由表中是否有到达目标地址的路径，如果查到一条较新的路径（即路由表中目标序列号大于或等于ROUTE REQUEST 帧中的目标序列号），则给源节点返回一个ROUTE REPLY帧。
 - 如果该节点不知道到达目标地址的较新的路径，则增加跳计数域，并且重新广播ROUTE REQUEST 帧。

目标节点的动作

- 当目标节点I最终收到ROUTE REQUEST 帧后，它发出一个ROUTE REPLY 帧作为应答。

源地址	目标地址	目标序列号	跳计数	生存时间
-----	------	-------	-----	------

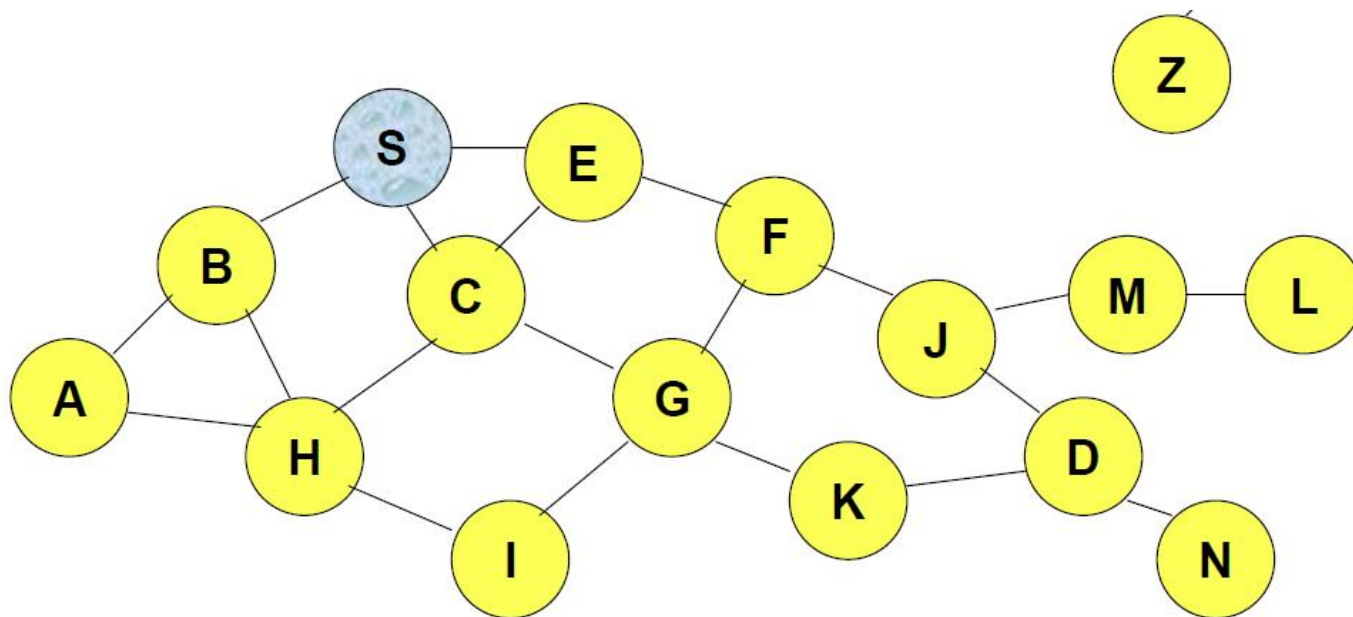
ROUTE REPLY帧

- 源地址和目标地址直接复制自接收到的路由请求帧。
- 目标序列号取自当前内存中的计数器。
- 跳计数域置为0。
- 生存时间控制该路径在多长时间有效期内。

反向路由的构建

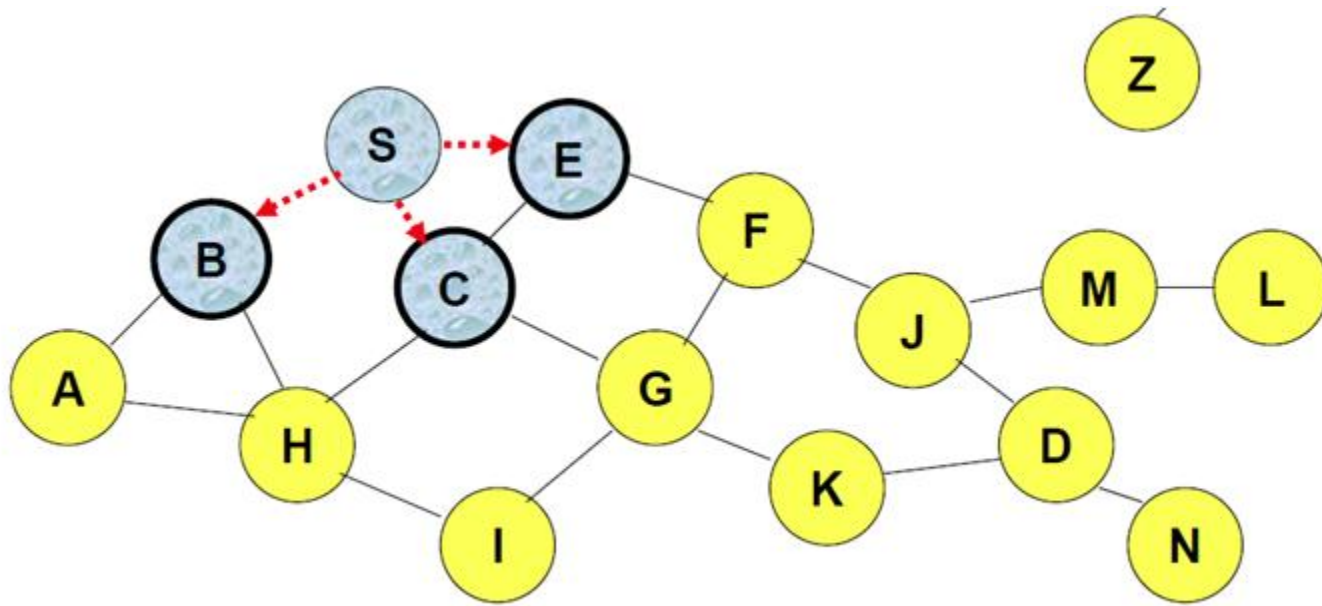
- ROUTE REPLY帧由目标节点I发出后，沿着逆向路径返回到源节点A，在每一个节点上，跳计数域都加1，这样每个节点都可以看到自己距离目标I有多远。
- 回程中每个节点都检查ROUTE REPLY帧，并更新自己到达目标I的路由信息，相当于所有节点都能捎带的得到了通往I的路由路径。

路由发现示例II (1)



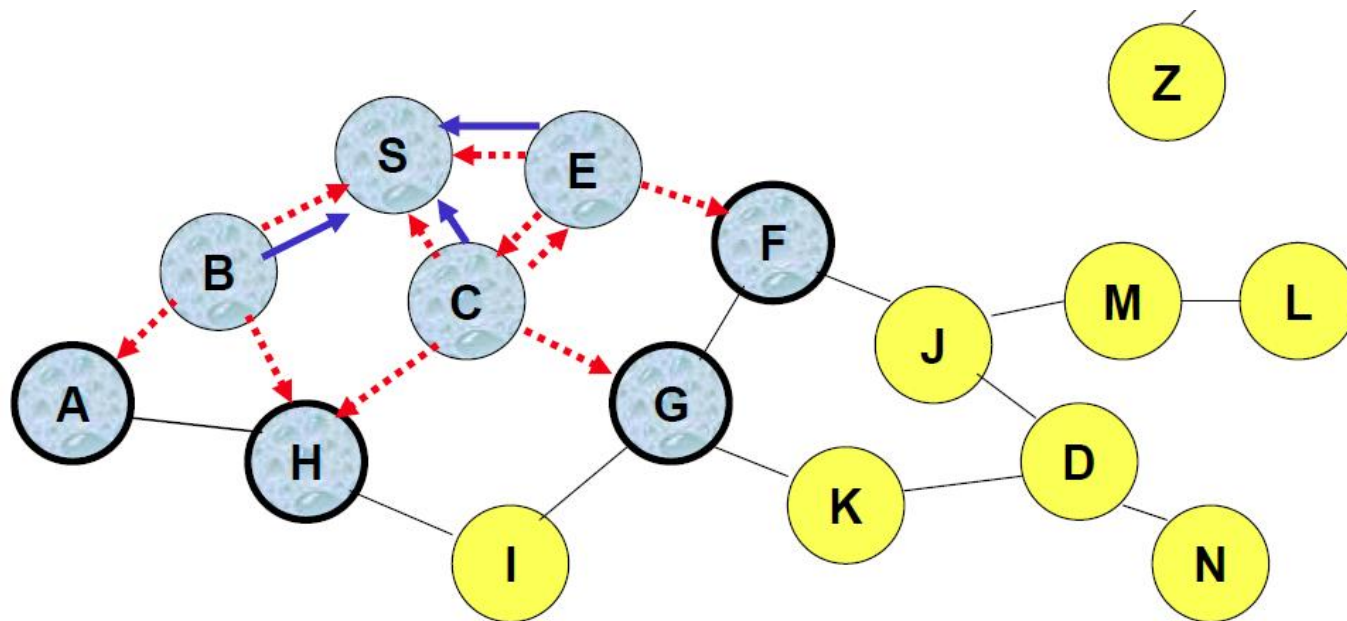
- S想给D发数据，在发送之前需先找到到D的路径。
- 蓝色节点表明该节点已收到S发出的路由请求帧。

路由发现示例II (2)



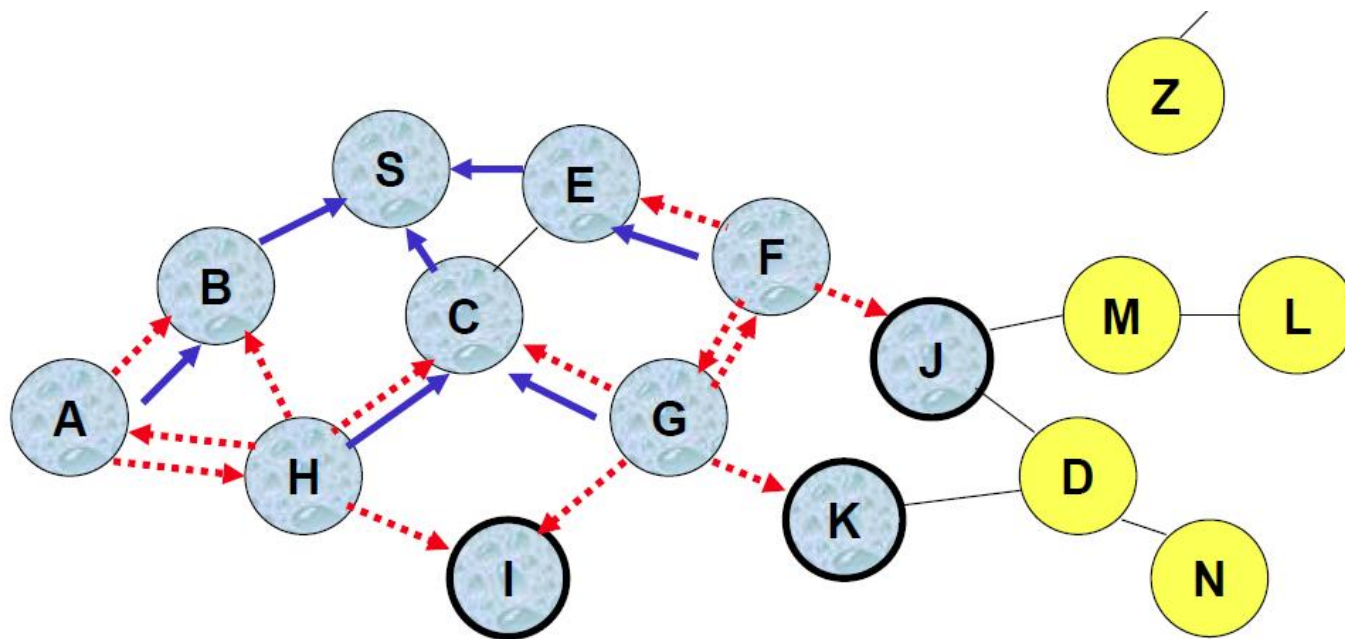
- 红色虚线表示路由请求帧的传输。

路由发现示例II（3）



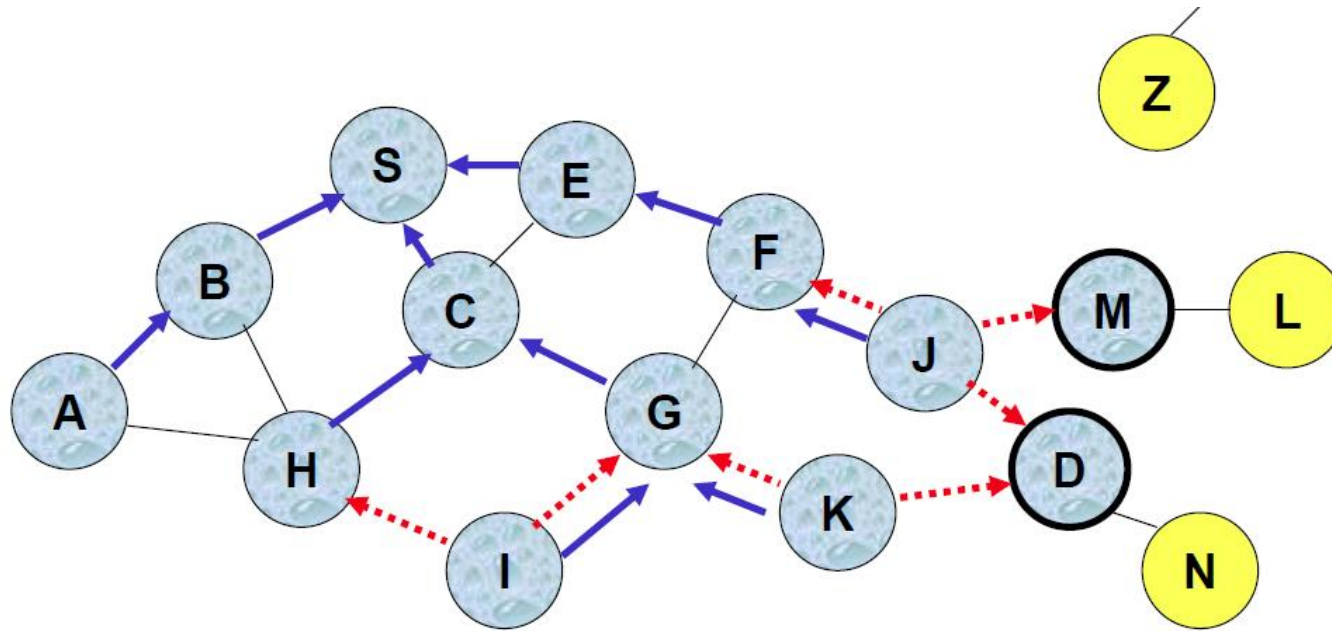
- 紫色实线表示反向路径的建立。

路由发现示例II（4）

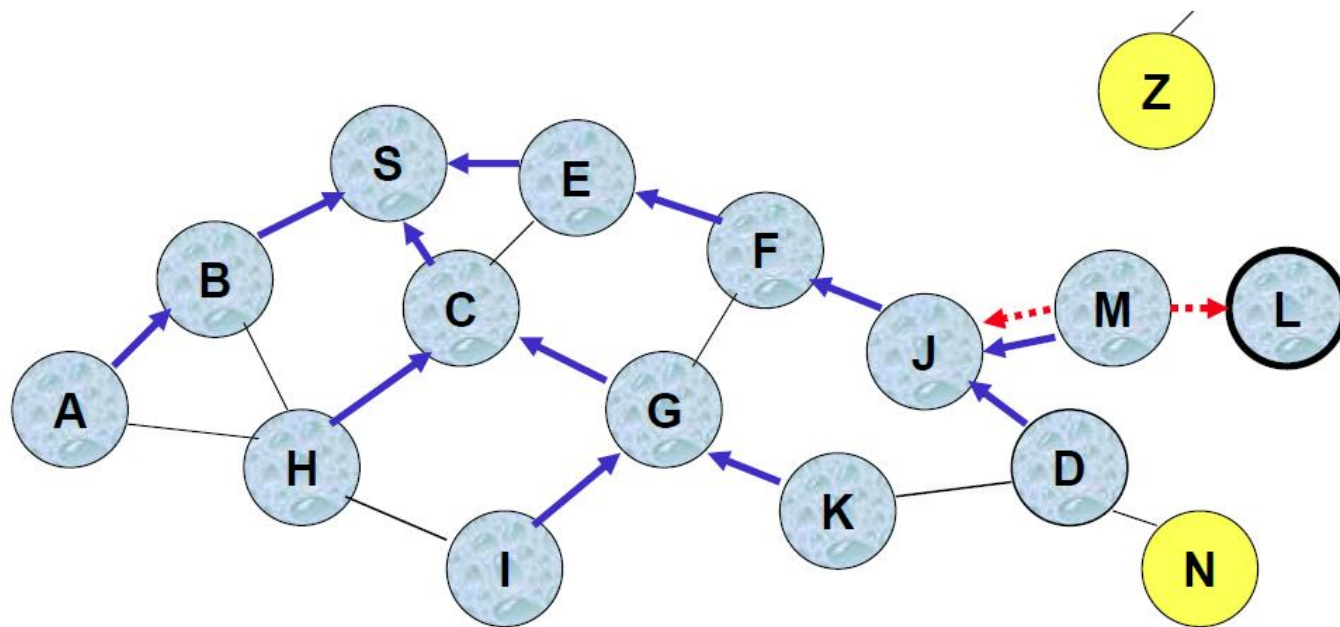


- C也会收到H和G广播来的路由请求帧，但不会再次转发该帧，因为它已经收到过一次（通过判定源地址和请求ID是否匹配）。

路由发现示例II（5）

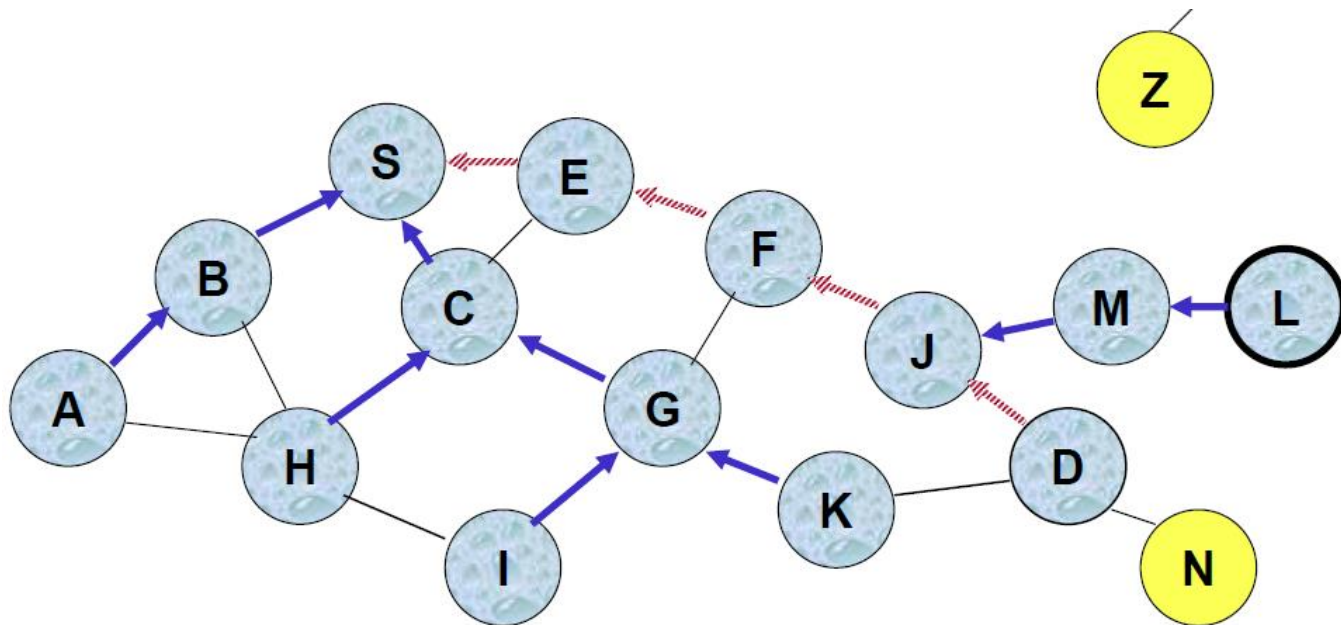



路由发现示例II（6）



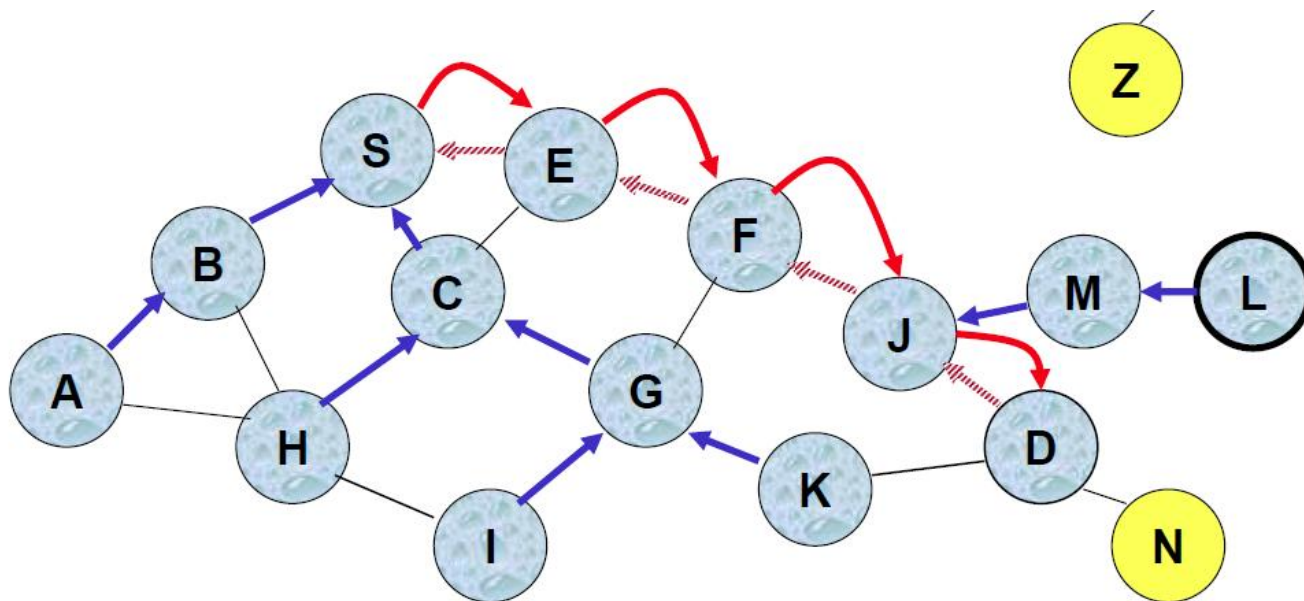
- D不会转发路由请求帧，因为D是该路由请求帧的目标节点。

路由发现示例II（7）




 表示路由响应帧传输路径

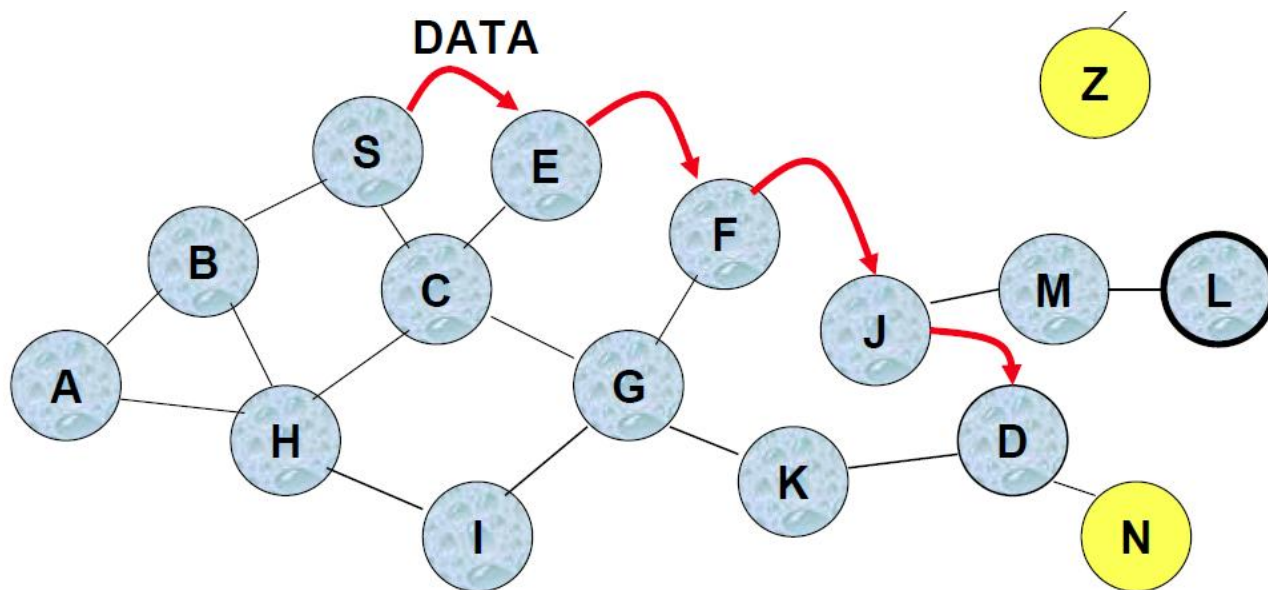
路由发现示例II（8）



- 当路由响应帧沿回程传输时，正向路径也同时建立起来。

 表示正向路径。

路由发现示例II（9）



- 通过路由请求帧和路由响应帧，就能更新路径中各个节点的路由表，从而建立起一条路由。
- 路由建立后，节点才发送实际的数据。AODV不再需要在数据帧的头部携带与路由有关的信息。



基于簇的路由协议 (拓扑控制协议)

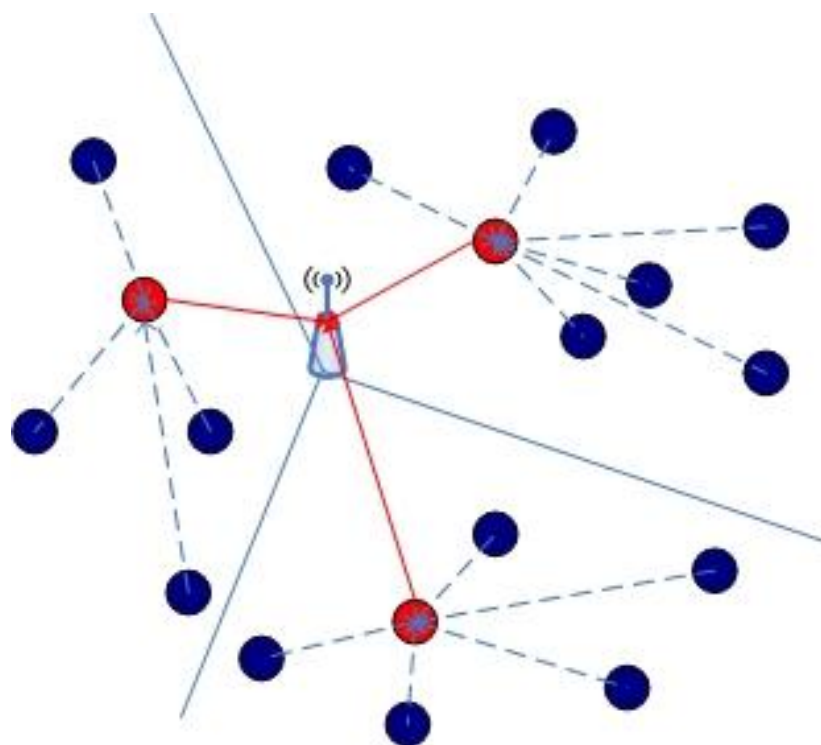
LEACH协议

LEACH协议的提出

- **LEACH(Low-Energy Adaptive Clustering Hierarchy)**是最早提出的分层路由算法，出发点主要是考虑一簇内节点的能量消耗问题，目的是为了延长节点的工作时间，并且实现节点的能耗平衡。
- **LEACH**协议是典型的分簇式路由协议（即集群结构路由协议）。
- **LEACH**协议因为用到了分簇，从簇形成的角度看，也可视为一种拓扑控制算法。
- **LEACH**是的基本思想是网络周期性地随机选择簇头节点，其他的非簇头节点以就近原则加入相应的簇头，形成虚拟簇。簇内节点将感知到的数据直接发送给簇头，由簇头转发给**Sink**节点，簇头节点可以将本簇内的数据进行融合处理以减少网络传输的数据量。

网络结构

- 网络中的所有传感器节点都是同构的，并且距离**Sink**节点较远。
- **Sink**节点处于固定位置。
- 每个节点可以直接同**Sink**节点通信。
- 由于**Sink**和节点的距离较大，节点直接和**Sink**通信会消耗大量的能量，使用**LEACH**后形成一个两级的星型网络。簇内节点直接和簇头通信，由簇头节点和**Sink**通信。



簇头的功能

- 簇头需一直工作，不能休眠。
- 簇头需把所收到的整簇内节点的数据转发给sink，非常消耗能量。
- 如果选择固定的节点来担任簇头，则该节点将很快消耗完能量，簇内节点也将变成“群龙无首”状态。
- 一个比较好的办法是大家轮流担任“簇头”，共同分担能量消耗。

簇头如何选举？

- 如果采取**sink**或其它设备来指定簇头的方式，则需要耗费额外的交互开销。
- 如果通过节点之间相互协商、相互沟通来选举簇头，也会增加很多开销。
- **LEACH**提出了一种“隐式竞争”的簇头选举策略，每个节点通过一定的规则独立的决定自己是否需要担任簇头，无须他人指定或相互协商，并且能够满足每个节点交替担任簇头的设计目标。

簇头选举方法

- 簇头节点的选择依据网络中所需的簇头节点数和迄今为止每个节点已成为簇头的次数来决定。
- 具体的选择办法是:每个传感器节点选择 $[0,1]$ 之间的一个随机数, 如果选定的值小于某一个阈值 $T(n)$, 那么这个节点成为簇头节点。
- $T(n)$ 可以理解为当选为簇头的几率, 例如当 $T(n)=0.6$ 时, 意味着节点有**60%**的可能当选为簇头。

T(n)的计算方法

$$T(n) = \begin{cases} \frac{P}{1 - P * (r \bmod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

- 式中，**P**是每一个回合中簇头在所有节点中所期望占的百分比。若设**N**为网络中传感器节点的总个数，**k**为在进行网络设计时在每一回合所期望的簇头节点数，则：**P = k/N**。**r**为当前的回合数，**r**的范围是第**0**回合到第**(1/P) - 1**回合，**G**为在一个大轮次内（包含**1/P**个回合）尚未成为簇头的节点的集合。
- 网络有很多轮次组成，每个轮次又由**1/P**个回合组成，每个回合都选举一定百分比的簇头出来，然后进行数据传输，本回合进行完毕后，在下一回合再选出一定百分比的簇头出来，直到第**(1/P) - 1**回合。在这**1/P**个回合里，每个节点都会当一遍簇头。第**(1/P) - 1**回合进行完毕后，则进入下一轮次的第**0**回合，如此一直重复。

T(n)计算方法示例

- 设网络中有**10**个节点，每个周期（即回合）想让**20%**的节点成为簇头，**LEACH**协议如何选举簇头呢？
- **P=0.2**。
- 设节点号分别为**1,2,3, ...,10**，带入公式，得到下表。

T(n)演进示意图

回合数， 即r的值	已当过簇头的 节点的T(n) n不属于G	尚未当过簇头 的节点的T(n) n属于G	本轮选举的 簇头（随机 假定）	剩余的还未担任 簇头的节点，即 集合G
第0回合 r=0	0	20%	5	1,2,3,4,6,7,8,9,10
第1回合 r=1	0	25%	3,4	1,2,6,7,8,9,10
第2回合 r=2	0	33%	6,10	1,2,7,8,9
第3回合 r=3	0	50%	1,2,8	7,9
第4回合 r=4	0	100%	7,9	空集

第5回合相当于下一轮次的第0回合（r可以一直增加，不断进入新的轮次），本例中每轮最大为 $1/P=5$ 个回合。在下一轮次中又会按照上表中的概率变化按回合轮流选举簇头。

簇头选举策略的深度分析

- LEACH把一个轮次分为 $1/P$ 个回合，并且总是能够保证在 $1/P$ 个回合内每个节点能够当选且只能当选一次簇头。
- “能够当选”的原因在于，在第 $(1/P)-1$ 回合时，任何没当选过的节点，其当选概率 $T(n)$ 都等于1，即在最后一个回合会“兜底”。
- “只能当选一次”的原因在于：一个节点在一个轮次内已经当选过一次簇头后，其在接下来的回合内当选概率直接为0。
- 随着回合数的增加，节点当选的概率逐渐增加（非线性增加），促使没有当选过的节点“抢簇头”的意愿增强。
- LEACH只是给出了每个回合当选的概率，至于节点能否真正当选，完全是一种概率；此外，每个回合里簇头数并不一定总等于期望个数，忽大忽小，有随机性。
- 虽然是概率，但LEACH保证了每个节点都能当选一次，而且总体上看，多个回合多个轮次的簇头平均个数趋近于期望个数，这是统计意义上的结果。

计算题

某传感器网络运行LEACH协议，节点总数为30个。回合数从 $r=0$ 开始计数，已知在回合数 $r=2$ 簇头选举中，尚未当选过簇头的节点，其被选为簇头的概率（即阈值）为25%。回答下列问题：

（1）在每一回合簇头选举中，网络设计者所期望的平均簇头数量是多少个？

（2）在回合数 $r=4$ 簇头选举中，对于尚未当选过簇头的节点，其被选为簇头的概率将变为多少？对于已经当选过簇头的节点，其再次被选为簇头的概率是多少？

答案

(1) 将 $r=2$, $N=30$, $T(n)=0.25$ 代入簇头选举公式, 得:

$$\frac{k}{30 - k * (2 \bmod \frac{30}{k})} = 0.25$$

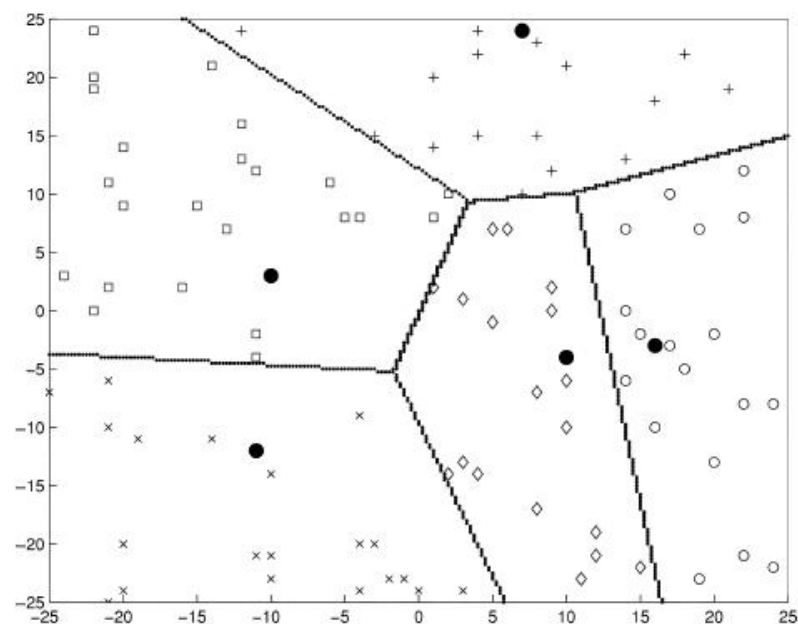
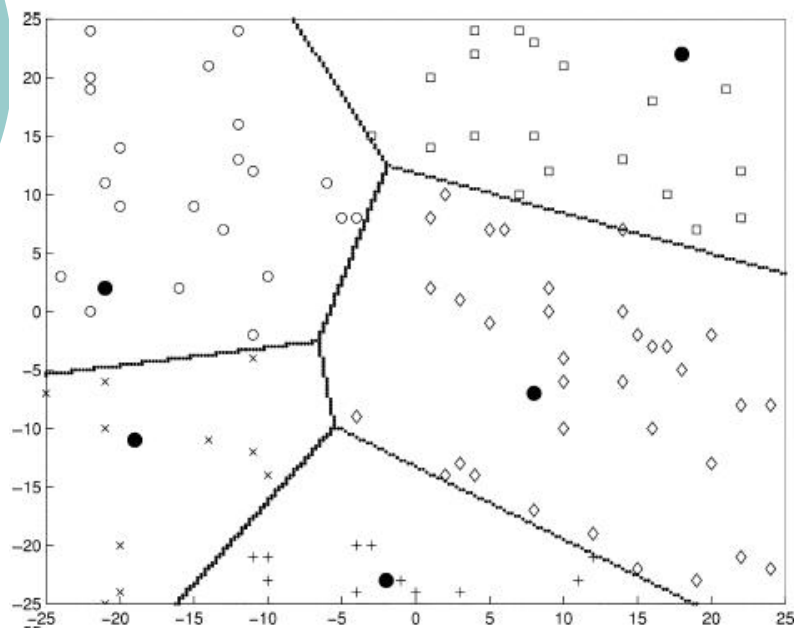
因为有取模运算, 所以分两种情况讨论: 当 $30/k \leq 2$ 时, 即 $15 \leq k \leq 30$ 时, k 无论取何值均不满足上式。当 $30/k > 2$ 时, 可化为:

$$\frac{k}{30 - k * 2} = 0.25$$

可得 $k=5$ 。即平均簇头数量为5。(6分)

(2) 把 $r=4$, $k=5$ 代入公式, 可得: 未当选过簇头的节点的当选概率为50%。已当选簇头的节点再次当选的概率总是为0。(4分)

LEACH簇头选举的仿真



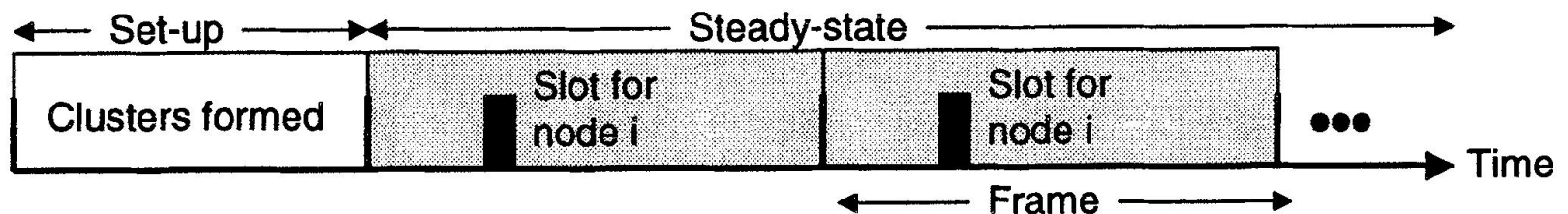
第 i 个回合与第 $i+1$ 个回合所选举的簇头和形成的虚拟簇，黑点为簇头，相同形状的符号为同一簇节点。

簇头选出来之后... ..

- 簇头选举的每个回合，可称为一个周期。
- 在每个周期里，除了选举簇头外，还要干实事，就是传输传感器数据。
- 每个周期分为两个阶段：簇的建立和稳定的数据传输阶段，稳定传输阶段的持续时间要大于簇建立所需要的时间。
- **簇的建立阶段：**选定簇头以后，簇头节点通过广播告知整个网络自己成为簇头的事实，网络中的非簇头节点根据接收信号的强度决定从属的簇，并通知相关的簇头，从而建立虚拟簇。上述通信过程通过**CSMA**来完成。

LEACH的TDMA传输机制

- **稳定传输阶段**：簇头节点采用**TDMA**方式为簇中每个节点分配传输数据的时间片，簇节点在自己的时间片内将数据传输给簇头。
- **LEACH**在簇内的传输可看做是基于调度的**MAC**协议，而加上簇头到**Sink**的传输后，又称为两跳的路由协议。所以有些书上将**LEACH**也归为**MAC**协议，有的归为拓扑控制协议。



LEACH的评价

LEACH协议从传输数据的能量和数量上进行了优化，提高了网络的生存时间。

LEACH有哪些缺点？

LEACH的评价

- LEACH协议从传输数据的能量和数量上进行了优化，提高了网络的生存时间。
- **LEACH**有哪些缺点？（请大家思考五分钟，请同学来讨论）
 - 簇头的产生过程时间上不均匀
 - 簇头产生的位置是随机的，有可能出现一片区域有多个簇头，而另外一片区域一个也没有的情况。
 - 数据流量没考虑
 - 每个节点都能与**sink**通信的假定不适合大规模**WSN**网络
 - 不适合异构网络，即节点的资源和能力不相同的网络。
 - 簇的大小影响能量消耗
 -



以数据为中心的路由协议

DD协议、SPIN协议

DD路由协议

- 定向扩散协议(**DD, Directed Diffusion**)是一种基于查询的路由方法，这和传统路由算法的概念不一样。**DD**算法是一种基于数据相关的路由算法，**Sink**节点周期地通过泛洪的方式广播一种称为“兴趣”的数据包，告诉网络中的节点它需要收集什么样的信息。“兴趣”在网络中扩散的时候同时也建立了路由路径，采集到和“兴趣”相关的数据的节点通过“兴趣”扩散阶段建立的路径将采集到的“兴趣”数据传送到**Sink**节点。
- **1.基本思想**
- 定向扩散协议是一种基于查询的路由协议，该协议中引入几个基本概念：**兴趣**、**梯度**和**路径加强**。整个过程可以分为**兴趣扩散**、**梯度建立**以及**路径加强**三个阶段。

定向扩散路由机制

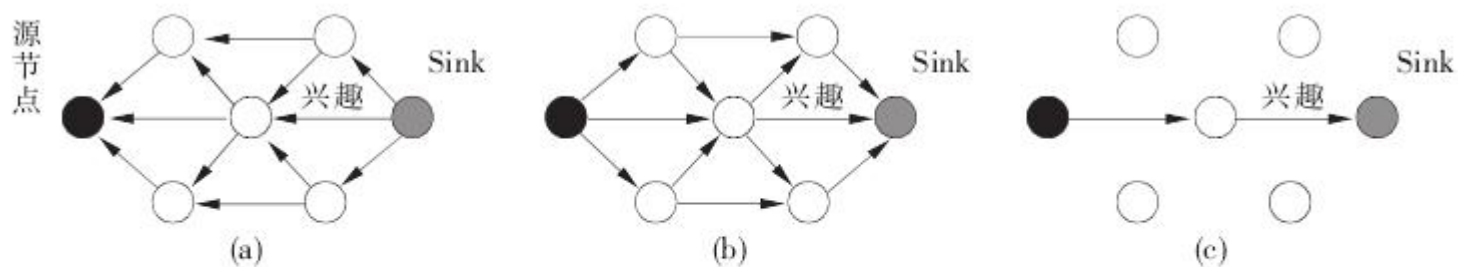


图 2-6 定向扩散路由机制

(a) 兴趣扩散; (b) 梯度建立; (c) 加强路径

DD路由协议

- 2.关键问题
- **DD**路由协议核心的问题是解决兴趣扩散阶段的**梯度建立**过程、**强化路径**的选择和建立过程以及**路由的维护**过程。
- 在兴趣扩散阶段，**Sink**节点周期性地向邻居节点广播兴趣消息。**兴趣消息中包含有任务类型、目标区域、数据传输率、时间戳等参数**。每个节点都有一个**兴趣Cache**，**兴趣Cache**中的何项都对应着不同的兴趣。

DD路由协议

- 一个兴趣表项可能有多个梯度域，每个梯度域对应一个和该“兴趣”消息有联系的邻居节点。节点接收到一个“兴趣”消息之后，再把该“兴趣”消息发送给与自己相邻的节点。兴趣扩散完成后，对于某个事件的梯度也就在网络中建立起来了。
- 当传感器节点采集到与兴趣匹配的数据时，将数据发送给对应梯度域中的邻居节点。
- 同一个数据包会经过多条路径到达**Sink**节点，**Sink**节点通过一定的标准，比如最小延时，来选择一条最优的路径作为强化路径。

DD路由协议

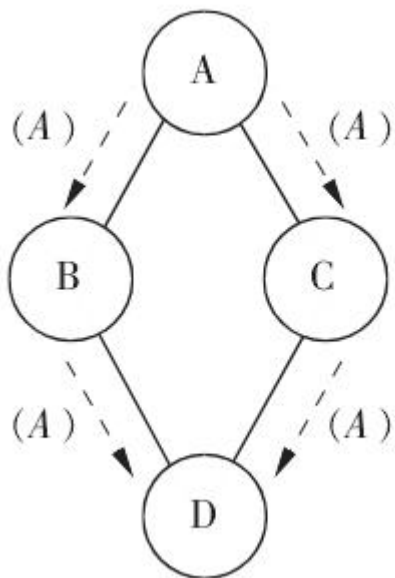
- **3.扩展分析**
- **DD**算法是一个以数据为中心的经典路由算法，该方法的显著特点是通过**Sink**节点发起兴趣查询消息，通过泛洪的方法建立传感器节点到**Sink**节点的路由。**DD**路由协议需要通过**Sink**节点完成对节点的查询，因此不能用于大规模的网络。该算法主要用于具有大量查询而只有少量事件的应用场景，如果网络拓扑结构频繁变动，该算法性能将大幅下降。

SPIN协议

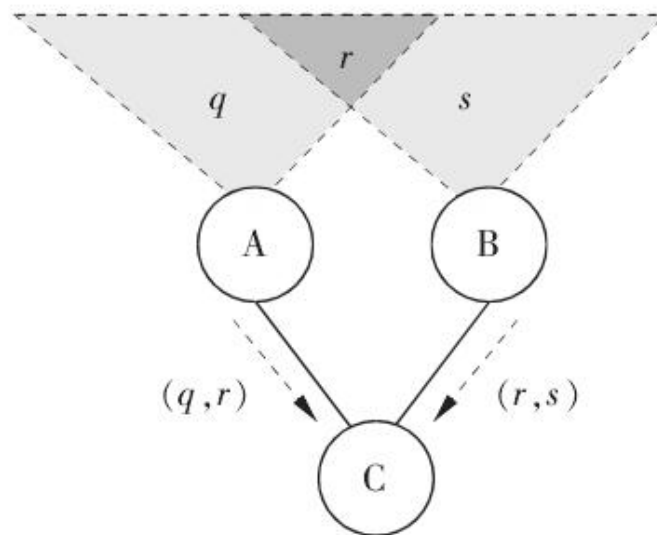
- SPIN (sensor protocol for information via negotiation) 协议是一种基于泛洪路由改进的以数据为中心的自适应路由协议。
- SPIN协议考虑到了WSN中的数据冗余问题：邻近的节点所感知的数据具有相似性，通过节点间协商的方式减少网络中数据的传输的数据量。节点只广播其他节点所没有的数据以减少冗余数据，从而有效减少能量消耗。
- 在扩散过程中不再传播重复和冗余的信息，减少扩散的通信量。

SPIN协议所解决的问题

- SPIN协议通过节点之间的协商，解决了Flooding协议和Gossiping协议的**内爆(Impllosion)**和**交叠(Overlap)**问题。

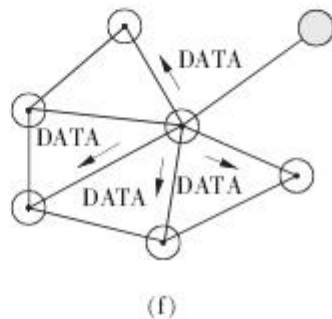
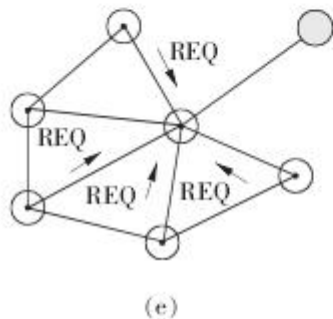
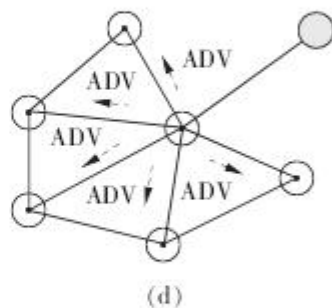
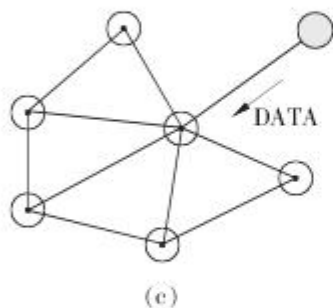
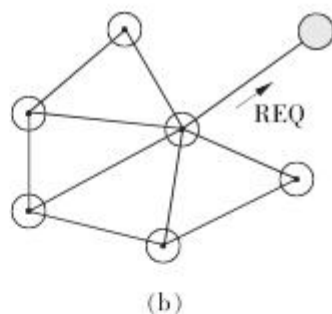
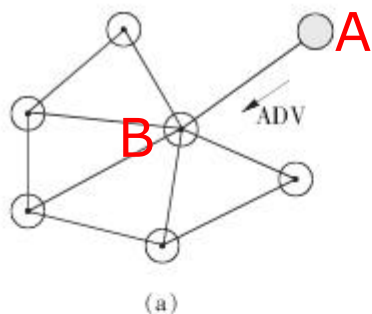


Impllosion: 同一数据包多次转发给同一节点



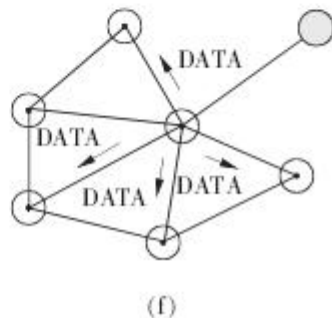
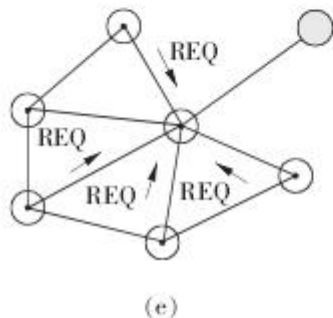
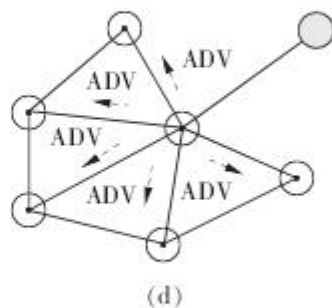
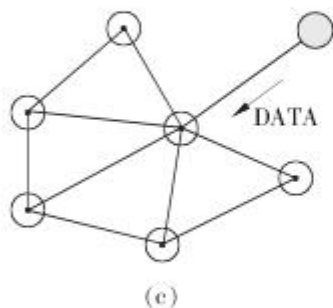
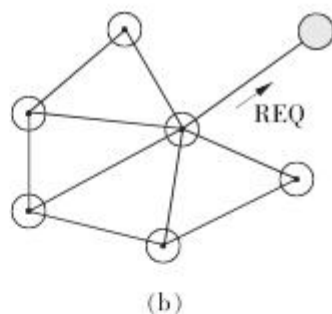
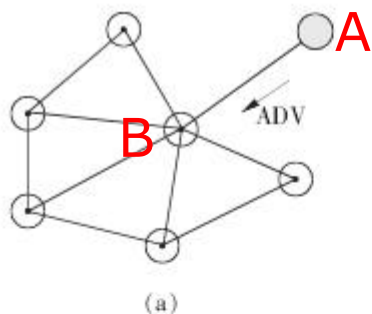
Overlap: 重叠区域的事件被两个传感器检测到，并传输给共同的邻居。虽然C收到的是不同的数据包，但因为内在数据一致，所以也属于浪费。

SPIN协议工作流程



- 当节点A感知到新事件之后，主动给其邻居节点广播描述该事件的元数据ADV报文，
- 收到该报文的节点B检查自己是否拥有ADV报文中所描述的数据，如图(a)。如果没有的话，节点B就向A发送REQ报文，在REQ报文中列出需要A节点给出的数据列表，如图2-(b)。
- 当节点A收到了REQ请求报文，它就将相关的数据发送给节点B，如图(c)。

SPIN协议工作流程



- 节点B发送ADV报文通知其邻居节点自己有新的消息，如图(d)，由于A节点中保存有ADV的内容，A节点不会响应B节点的ADV消息。
- 协议按照这样的方式进行，将数据扩散到网络中那些希望得到数据的节点。如果收到ADV报文的节点发现自己已经拥有了ADV报文中描述的数据，那么它不发送REQ报文，图(e)中有一个节点没有发送REQ报文。

SPIN协议的类型

- SPIN协议的最初版本称为**SPIN-PP**，节点之间采用单播通信，既不考虑节点剩余能量，也不考虑信道丢包。
- SPIN协议簇还包含了3种不同的形式：
 - **SPIN-EC**：在**SPIN-PP**的基础上增加了能量自适应策略，只有能量不低于设定阈值的节点才能参与数据交换。
 - **SPIN-BC**：采用了广播信道，并且为了防止产生重复的REQ请求，节点在听到ADV消息后，设定了一个随机定时器来控制REQ请求的发送。
 - **SPIN-RL**：在**SPIN-BC**基础上，进一步考虑如何恢复无线链路引入的分组差值与丢包。当发送REQ后一段时间内没有接收到数据，则重传请求，直至达到最大重传次数。

SPIN协议总结

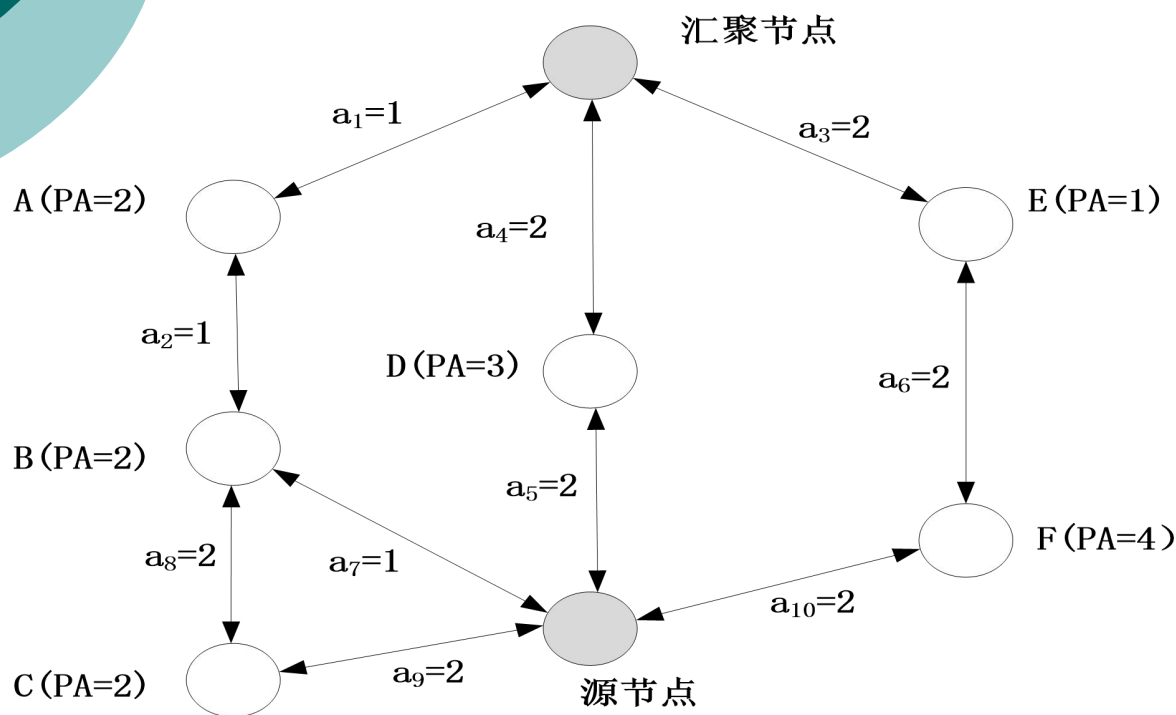
- SPIN协议充分体现了“以数据为中心”的思想，节点间的数据传输以内容为依据。
- SPIN通过协商机制有效的避免了节点间不必要的数据传输，减少了数据传输的不确定性。
- SPIN也存在潜在的问题，如果对数据感兴趣的节点不是携带数据节点的邻居节点，那么信息将无法到达这些节点。



能量感知路由协议

能量路由

能量路由，根据节点的剩余可用能量（**PA, Power Available**）或传输路径上的能量需求，来选择数据的传输路径。



路径1：源节点-B-A-汇聚节点

PA之和为**4**，能量之和为**3**；

路径2：源节点-C-B-A-汇聚节点

PA之和为**6**，能量之和为**6**；

路径3：源节点-D-汇聚节点

PA之和为**3**，在能量之和为**4**；

路径4：源节点-F-E-汇聚节点

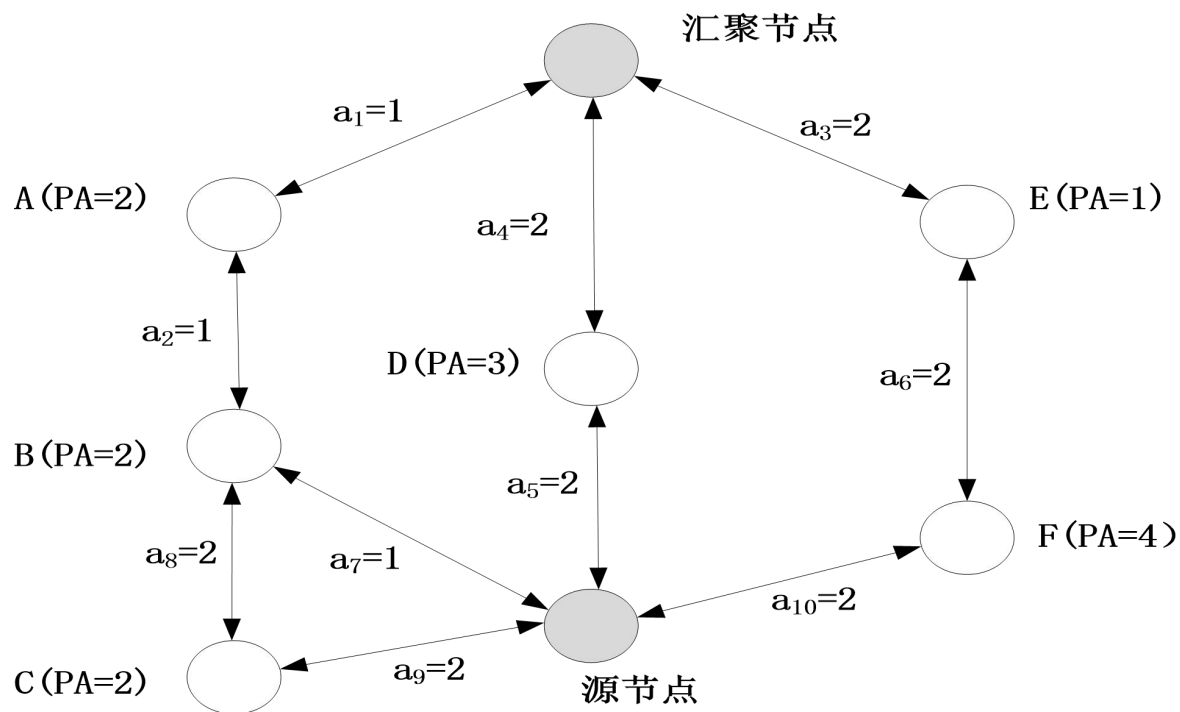
PA之和为**5**，能量之和为**6**。

能量路由选择策略

- **最小能量消耗路由**：从数据源到汇聚节点的所有路径中选取节点**耗能之和最少**的路径。
- **最少跳数路由**：选取从数据源到汇聚节点**跳数最少**的路径。
- **最大PA路由**：从数据源到汇聚节点的所有路径中选取节点**PA之和最大**的路径。（注意：需要将存在绕路的路径排除）
- **最大最小PA节点路由**：每条路径上有多个节点，且节点的可用能量不同，从中选取每条路径中可用能量最小的节点来表示这条路径的可用能量。如路径4中节点E的可用能量最小为1，所以该路径的可用能量是1。最大最小PA节点路由策略就是选择路径可用能量最大的路径。

下列路径中，满足最小能量消耗路由的是（）

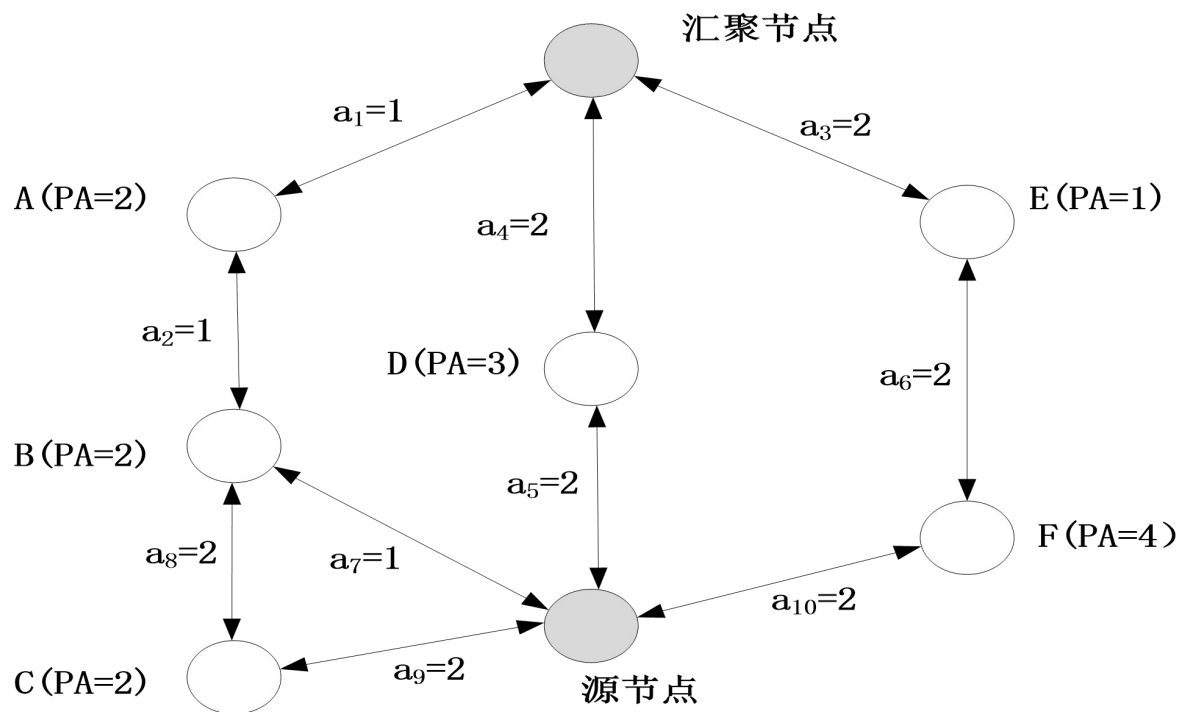
- ☒ A 路径1
- ☐ B 路径2
- ☐ C 路径3
- ☐ D 路径4



提交

下列路径中，满足最大PA路由的是（）

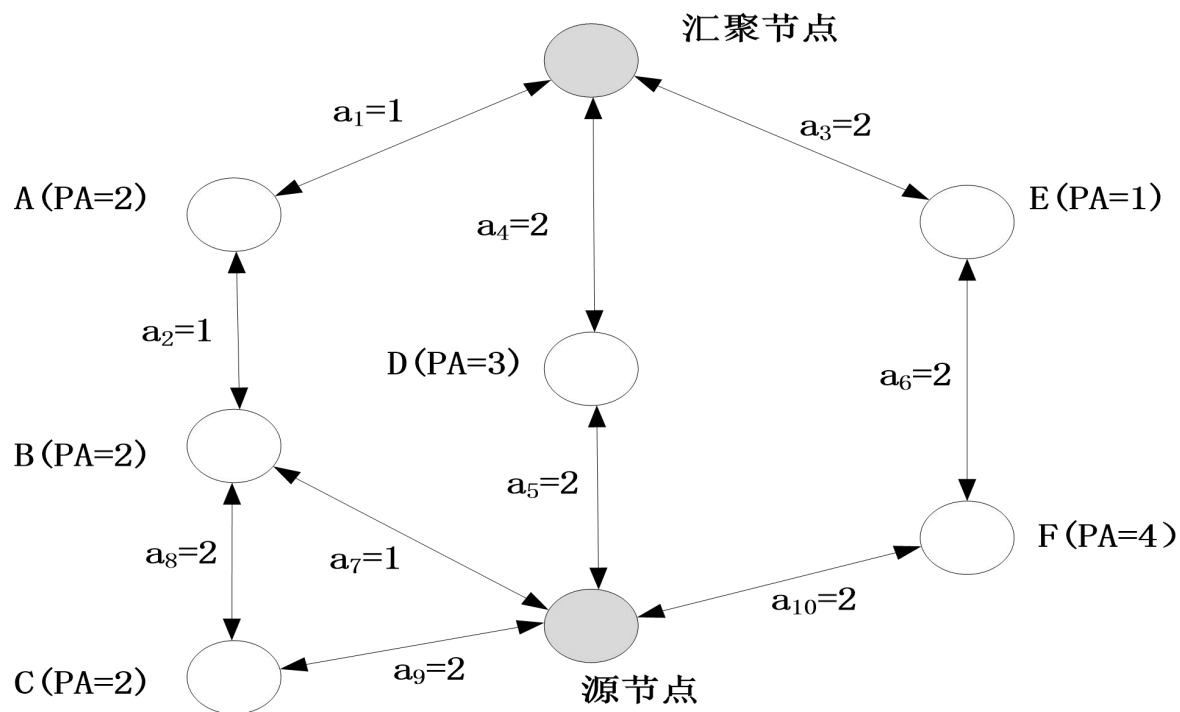
- ☐ A 路径1
- ☐ B 路径2
- ☐ C 路径3
- ☒ D 路径4



提交

下列路径中，满足最大最小PA节点路由的是（）

- ☐ A 路径1
- ☐ B 路径2
- ☒ C 路径3
- ☐ D 路径4



提交

能量多路径路由协议概述

○ 问题的提出

- 如果根据某种路由选择准则（例如跳数最小）来传输数据，会造成频繁使用同一路径，使该路径上的节点能量消耗过快而过早失效，缩短全网的生存时间。

○ 能量多路径路由协议的原理

- 在源节点和目标节点之间建立多跳路径，根据路径上的节点的**传输能量消耗**和**剩余能量**情况，给每条路径赋予一定的选择概率，使数据传输均衡的消耗整个网络的能量，延长全网的生存时间。

能量多路径路由协议基础

- 能量多路径路由协议包括**路径建立、数据传播和路由维护**三个过程。
- 路径建立过程是目标节点主动发起的。（类似于**DD**路由协议，目标节点来发起找到源节点，然后源节点再传输数据）
- 每个转发节点需要知道**到达目的节点的所有下一跳节点**，并计算选择每个下一跳节点传输数据的概率。概率的选择是根据节点到目的节点的通信代价来计算。（**通信代价**越小，表明链路质量越优）

路径建立过程（1）

- （1）目的节点向邻居节点广播路径建立消息，启动路径建立过程。路径建立消息中包含一个**代价域**，表示发出该消息的节点到目的节点路径上的能量代价信息，初始值设置为零。

$$Cost(N_D)=0$$

- （2）当节点收到邻居节点发送的路径建立消息时，相对发送该消息的邻居节点，**只有当自己距源节点更近，而且距目的节点更远的情况下，才需要转发该消息**，否则将丢弃该消息。

路径建立过程（2）

- （3）如果节点决定转发路径建立消息，需要计算新的代价值来替换原来的代价值。当路径建立消息从节点 N_i 发送到节点 N_j 时，该路径的通信代价值为节点 i 的代价值加上两个节点间的通信能量消耗，即：

$$C_{N_j, N_i} = \text{Cost}(N_i) + \text{Metric}(N_j, N_i)$$

- 其中

$$\text{Metric}(N_j, N_i) = e_{ij}^{\alpha} R_i^{\beta}$$

- 这里 e_{ij}^{α} 表示节点 N_j 和 N_i **直接通信的能量消耗**， R_i^{β} 表示节点 N_i 的**归一化剩余能量**（书上P253页将该变量描述成“剩余能量”是错误的）， α 、 β 是常量，根据系统需求自由选定，这个度量标准综合考虑了节点的能量消耗以及节点的剩余能量。
- **归一化剩余能量 = 节点的初始能量 / 节点当前剩余能量**。链路传输消耗的能量越大，**Metric**就越大；节点剩余能量越小，**Metric**越大。**Metric**越小，链路越优。

路径建立过程（3）

- （4）节点要放弃代价太大的路径，节点 j 将节点 i 加入本地路由表 FT_j 中的条件是：

$$FT_j = \{i \mid C_{N_j, N_i} \leq \alpha(\min_k(C_{N_j, N_k}))\}$$

其中， α 为大于1的系统参数。

- （5）节点为路由表中每个下一跳节点计算选择概率，节点选择概率与能量消耗成反比。节点 N_j 使用如下公式计算选择节点 N_i 的概率：

$$P_{N_j, N_i} = \frac{\frac{1}{C_{N_j, N_i}}}{\sum_{k \in FT_j} \frac{1}{C_{N_j, N_k}}}$$

路径建立过程（4）

- （6）节点根据路由表中每项的能量代价和下一跳节点选择概率计算本身到目的节点的代价 $\text{Cost}(N_j)$ 。 $\text{Cost}(N_j)$ 定义为经由路由表中节点到达目的节点的代价的平均值，即：

$$\text{Cost}(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i}$$

- 节点 N_j 将用 $\text{Cost}(N_j)$ 值替换消息中原有的代价值，然后向邻居节点广播该路由建立消息。
- 在数据传播阶段，对于接收的每个数据分组，节点根据概率从多个下一跳节点中选择一个节点，并将数据分组转发给该节点。路由的维护是通过周期性地从目的节点到源节点实施洪泛查询来维持所有路径中的活动性。
- **总结：能量多路径协议根据概率选择下一跳路径，从而能够将能量消耗分散到各个路径和节点上去，实现整个网络平稳的能量使用，延长了整体生存时间。**



地理位置路由协议 **GEAR**协议

基于地理位置信息的路由协议

- 在路由协议中使用地理位置信息，一般有两种用途：
 - 使用地理位置信息作为其他路由算法的辅助。在地理位置信息的支持下，可以限制网络中搜索路由的范围，减少路由控制分组的数量，一般用于基于泛洪的路由协议的改进。例如**GEAR**路由。
 - 直接使用地理位置信息建立路由，节点直接根据位置信息制定数据转发策略。例如**GPSR**路由。

2.5 WSN路由协议的发展

- 路由协议是无线传感器网络当前研究的热点之一，传感器网络由于其自身资源受限的特点，对路由协议的要求非常高，设计个通用的路由协议是一件非常困难的事。一般传感器网络路由协议的设计都专门针对特定的应用场景，传感器网络应用场景的专一性，为设计高效专用的路由协议带来了可能性。尽管如此，**WSN**的路由设计还是存在大量挑战，这些问题包括传统的**Internet**由协议设计中碰到的问题，还包括由**WSN**自身的特点引入的问题。尽管在**WSN**路由算法的研究上取得了很多进展，但还是有一些根本性的问题需要进一步地解决。

2.5 WSN路由协议的发展

- **WSN**路由协议还需进一步解决如下问题：
 - 全局最优路由策略
 - 路由算法的**QoS**支持
 - 路由算法的安全性
 - 能源有效路由策略

作业2

- **P219** **7.22** **7.25** **7.26**
- **P261** **8.12** **8.14**



Thank you !