物联网通信技术

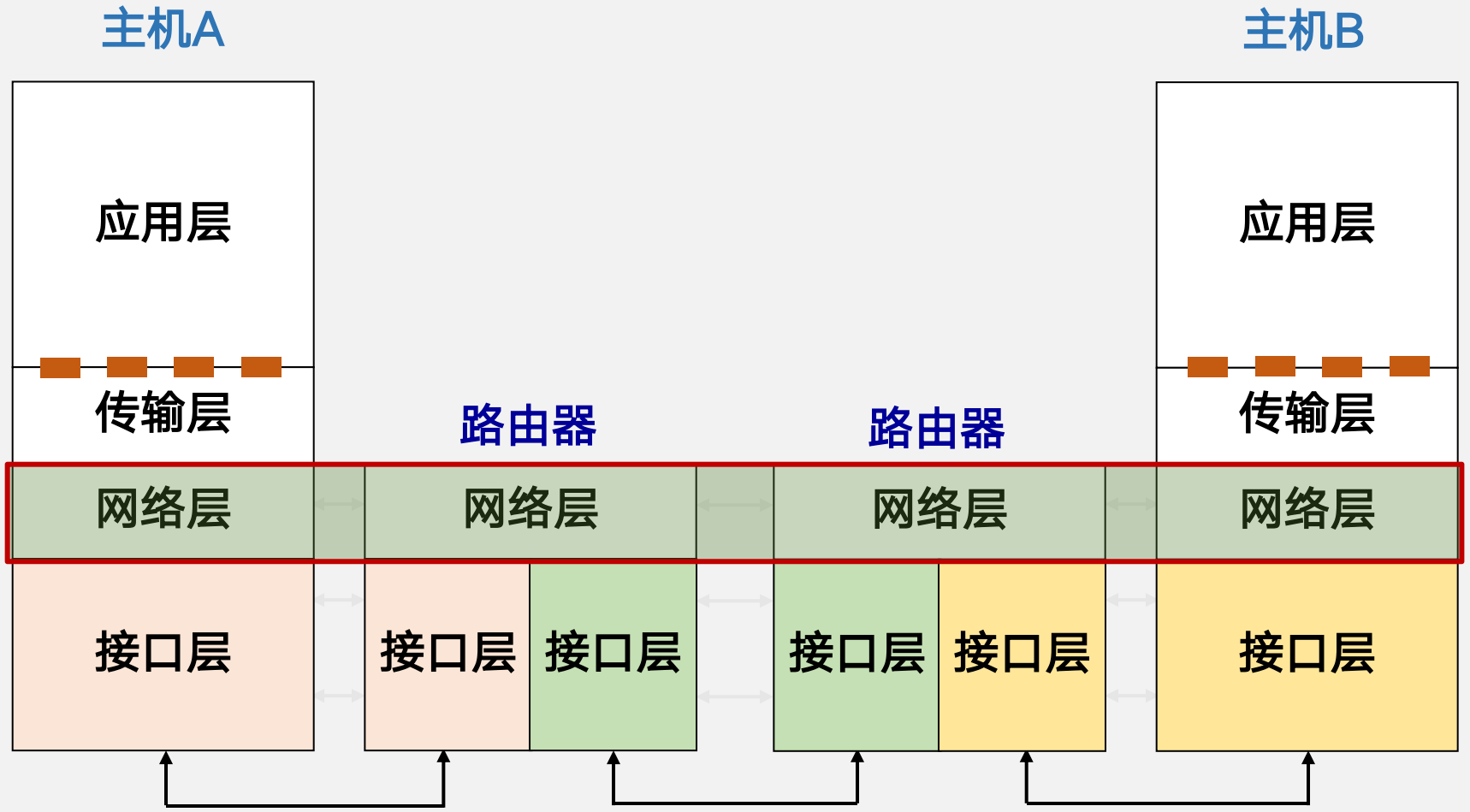
主讲人：宁 磊
Email：ninglei@sztu.edu.cn

目 录

CONTENTS

- 本章主要内容：集中式路由算法、分布式路由算法，自组织网络路由协议，无线传感网络路由协议。
- 本章学习目标
  - 了解集中式路由和分布式路由算法;
  - 熟悉自组织网络路由协议，掌握DSDV和AODV协议;
  - 熟悉无线传感网络路由协议，掌握LEACH协议。

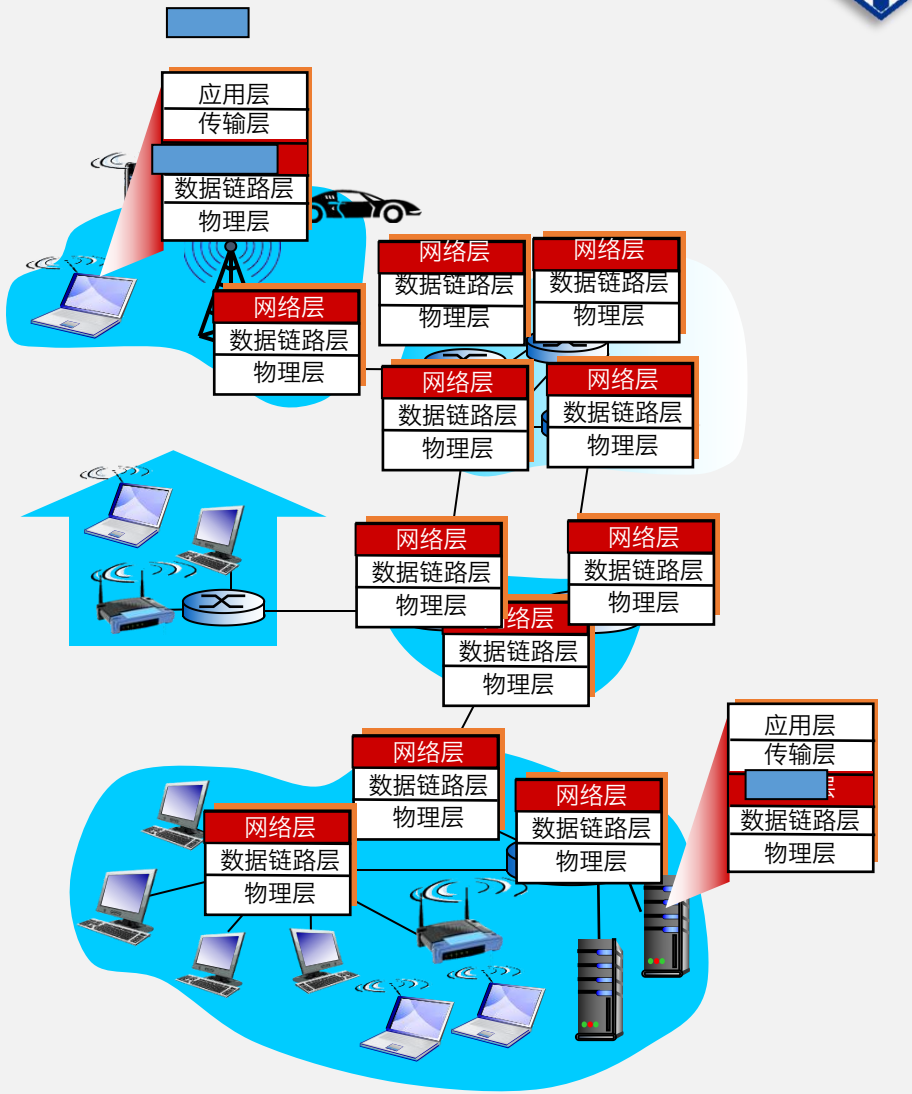
网络层服务概述



接口层通常包括数据链路层和物理层

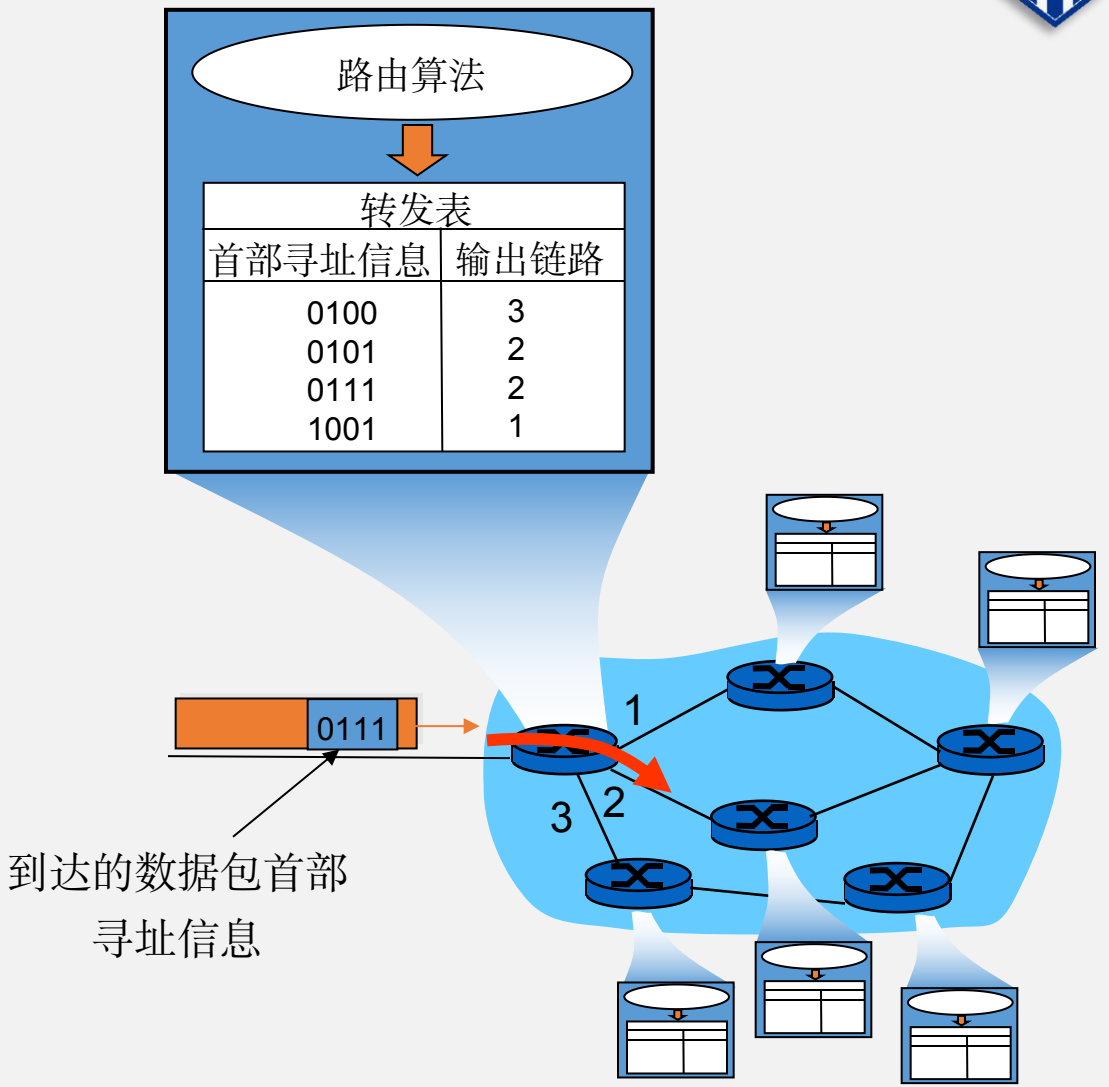
网络层服务的实现

- 网络层实现端系统间多跳传输可达
- 网络层功能存在每台主机和路由器中
  - 发送端：将传输层数据单元封装在数据包中
  - 接收端：解析接收的数据包中，取出传输层数据单元，交付给传输层
  - 路由器：检查数据包首部，转发数据包

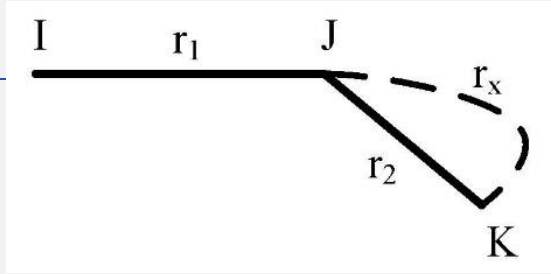


网络层关键功能

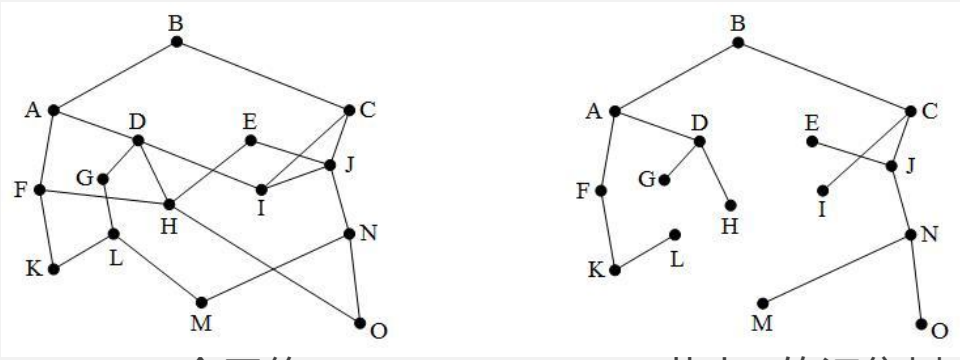
- 路由（控制面）
  - 选择数据报从源端到目的端的路径
  - 核心：路由算法与协议
- 转发（数据面）
  - 将数据报从路由器的输入接口传送到正确的输出接口



算法概述



- 优化原则
  - 如果节点J是在节点I到节点K的最优路径上，那么，从J到K的最优路径也必定沿着同样的路由路径 **最优化原理**
  - 最优化原则的一个直接结果：从所有的源到一个指定目标的最优路径的集合构成了一棵以目标节点为根的汇集树
  - 路由算法的目标：为所有节点找到这样的汇集树，并根据汇集树来转发数据包

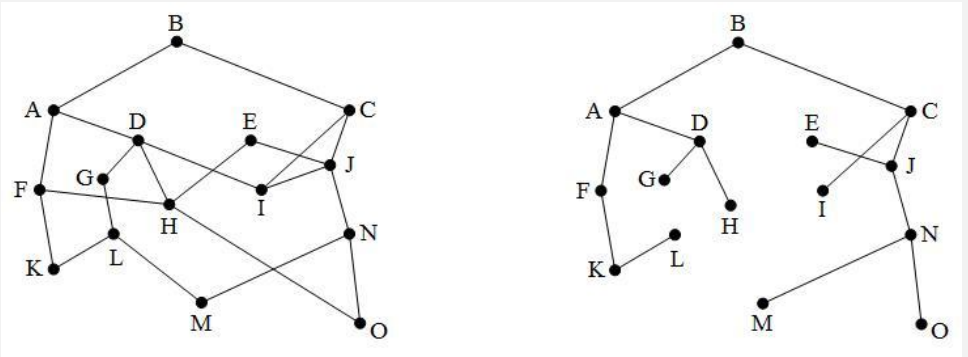


(a) 一个网络　　(b) 节点B的汇集树

汇集树

算法概述

- 最短路由算法
  - 使用广泛（简单且易于理解）
  - 最短取决于对链路长度的定义
  - 链路长度通常是一个正数，它可以是物理距离的长短、时延的大小、各个节点队列长度等等
  - 链路长度随着时间可能是变化的
  - 最短路由算法的理论基础是图论



路由算法分类

- 按路由决策
  - 集中式路由算法、分布式路由算法
- 集中式路由算法基本思路：路由控制中心计算路由，周期性收集各链路的状态，经过路由计算后，周期性地向各网络节点提供路由表
- 分布式路由算法基本思路：网络中所有节点通过相互交换路由信息，独立地计算到达各节点的路由

集中式路由算法

- Bellman-Ford算法（负权边有效）
- Dijkstra算法(复杂度低)

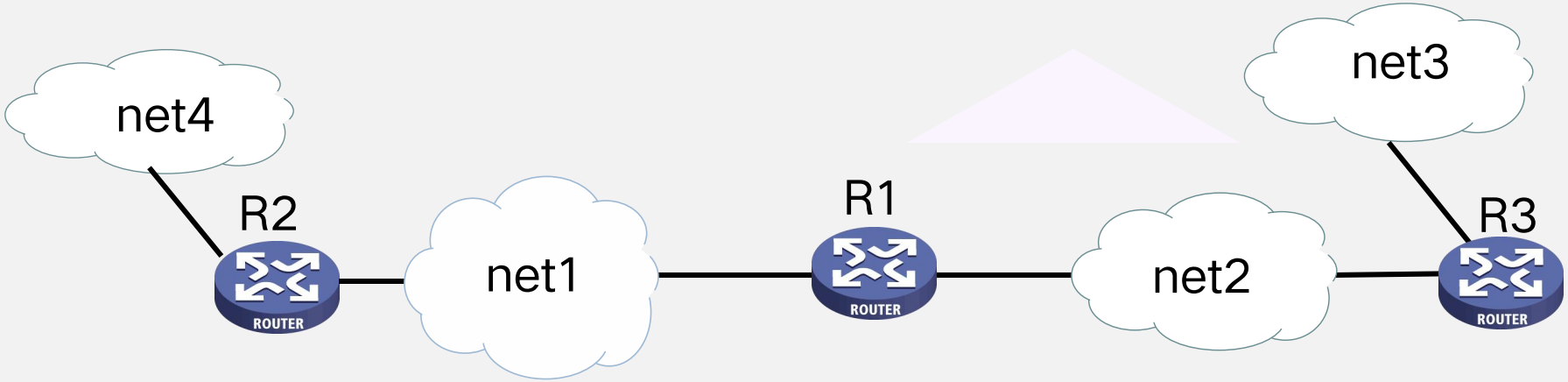
点对多点最短路径算法

分布式最短路径算法

- 分布式路由算法的特点
  - 节点用于存储网络拓扑结构的空间较少
  - 每个节点周期性的从相邻的节点获得网络状态信息，同时也将本节点做出的决定周期性的通知周围的各节点，以使这些节点不断地根据网络新的状态更新其路由选择
  - 各个节点的路由表相互作用
  - 整个网络的路由经常处于一种动态变化的状态。当网络状态发生变化时，会影响到许多节点的路由表。要经过一定的时间以后，各路由表中的数据才能达到稳定的数值
  - 分布式路由选择算法的核心思想是各个节点独立的计算最短路径
- 典型的分布式最短路径选择算法有距离矢量路由算法和链路状态路由算法

距离向量路由

- 路由器启动时初始化自己的路由表
  - 初始路由表包含所有直接相连的网络路径，距离均为0



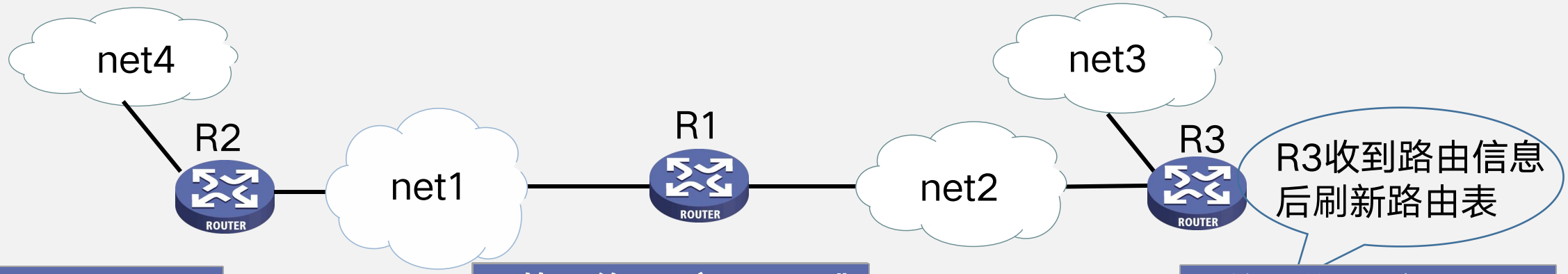
目的网络	距离	下一跳
net1	0	--
net4	0	--

目的网络	距离	下一跳
net1	0	--
net2	0	--

目的网络	距离	下一跳
net2	0	--
net3	0	--

距离向量路由

- 路由器周期性地向其相邻路由器广播自己知道的路由信息
- 相邻路由器可以根据收到的路由信息修改和刷新自己的路由表



目的网络	距离	下一跳
net1	0	--
net4	0	--

①

R2将路由信息发送给R1

目的网络	距离	下一跳
net1	0	--
net2	0	--
net4	1	R2

②

R1收到路由信息后刷新路由表

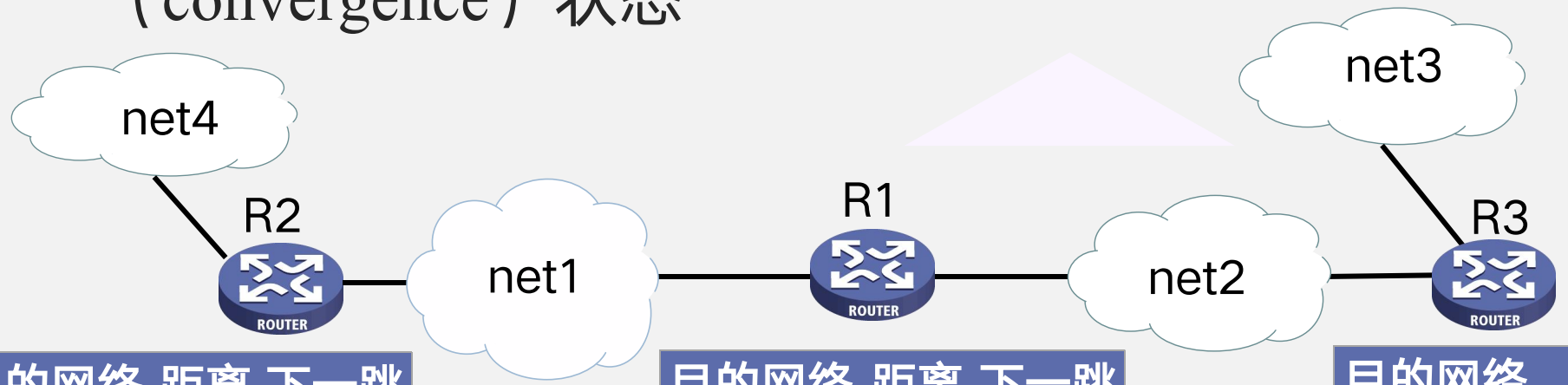
③

R1将路由信息发送给R3

目的网络	距离	下一跳
net2	0	--
net3	0	--
net1	1	R1
net4	2	R1

距离向量路由

- 路由器经过若干次更新后，最终都会知道到达所有网络的最短距离
- 所有的路由器都得到正确的路由选择信息时网络进入“收敛”（convergence）状态



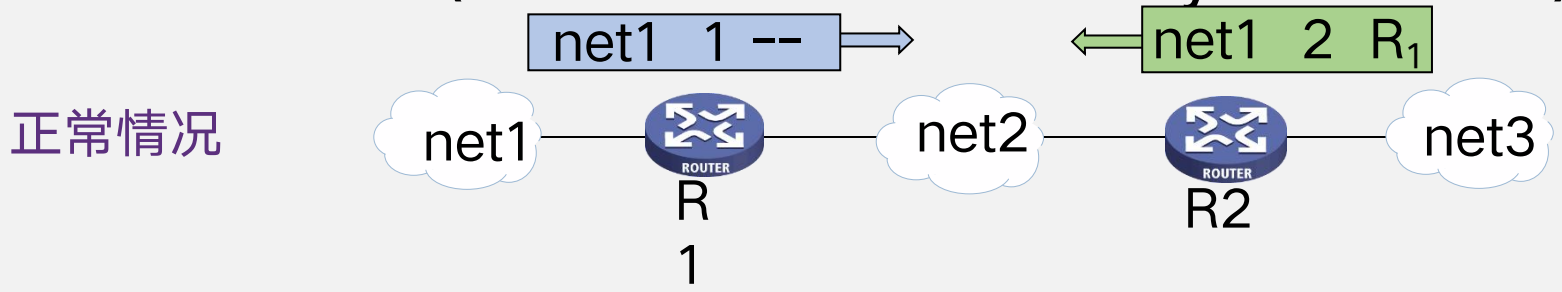
目的网络	距离	下一跳
net1	0	--
net4	0	--
net2	1	R1
net3	2	R1

目的网络	距离	下一跳
net1	0	--
net2	0	--
net4	1	R2
net3	1	R3

目的网络	距离	下一跳
net2	0	--
net3	0	--
net1	1	R1
net4	2	R1

距离向量路由

- 计数到无穷问题（The Count-to-Infinity Problem）

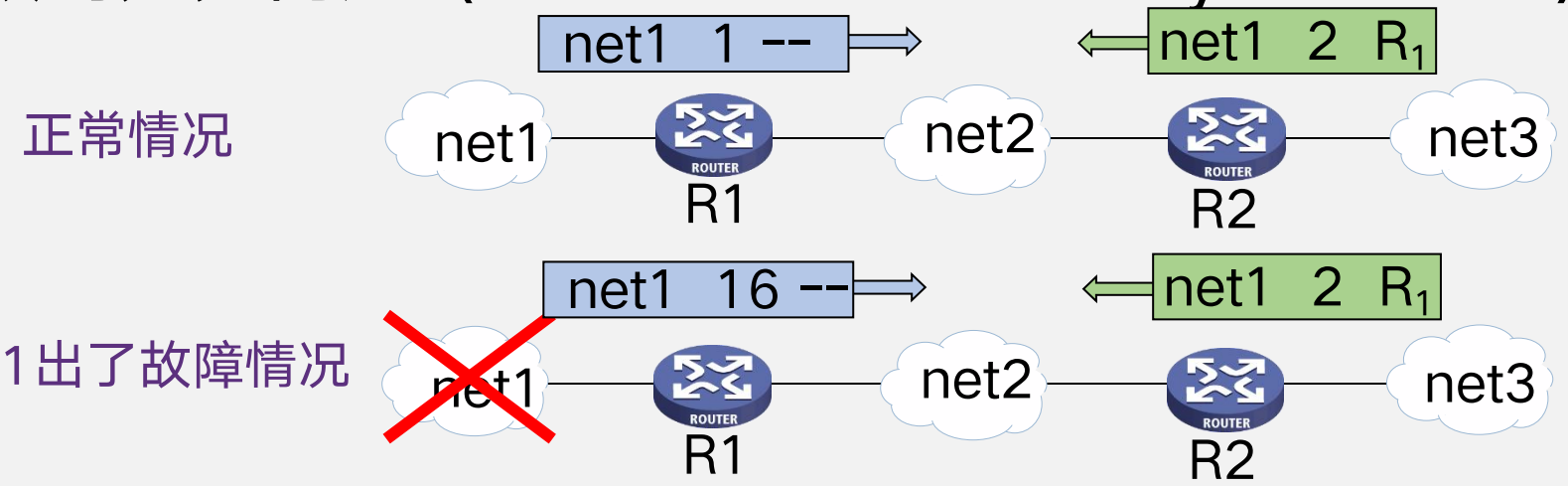


R_1 说：“我到net1 的距离是 1，是直接交付。”

R_2 说：“我到net1 的距离是 2，是经过 R_1。”

距离向量路由

- 计数到无穷问题（The Count-to-Infinity Problem）

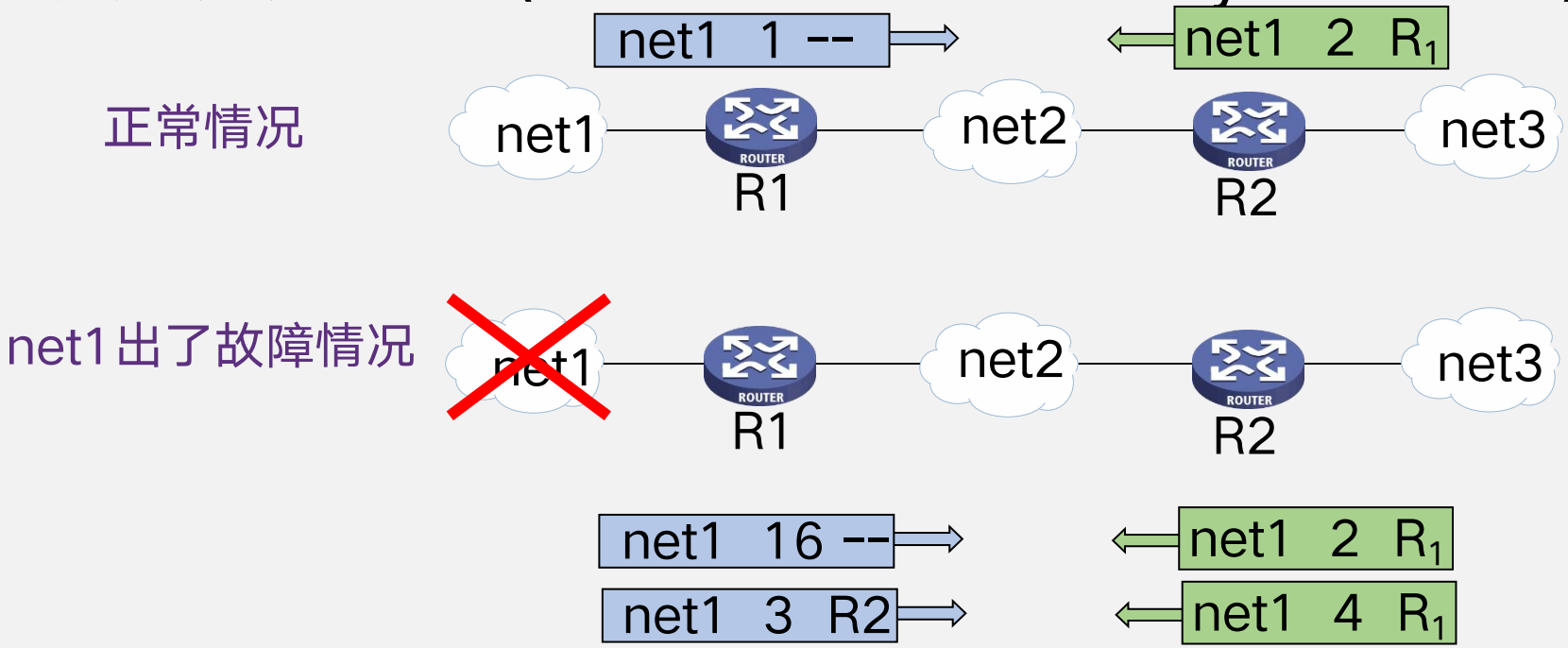


R_1 说：“我到net1 的距离是 16（表示无法到达），是直接交付。”

但 R_2 在收到 R_1 的更新报文之前，还发送原来的报文，因为这时 R_2 并不知道 R_1 出了故障。

距离向量路由

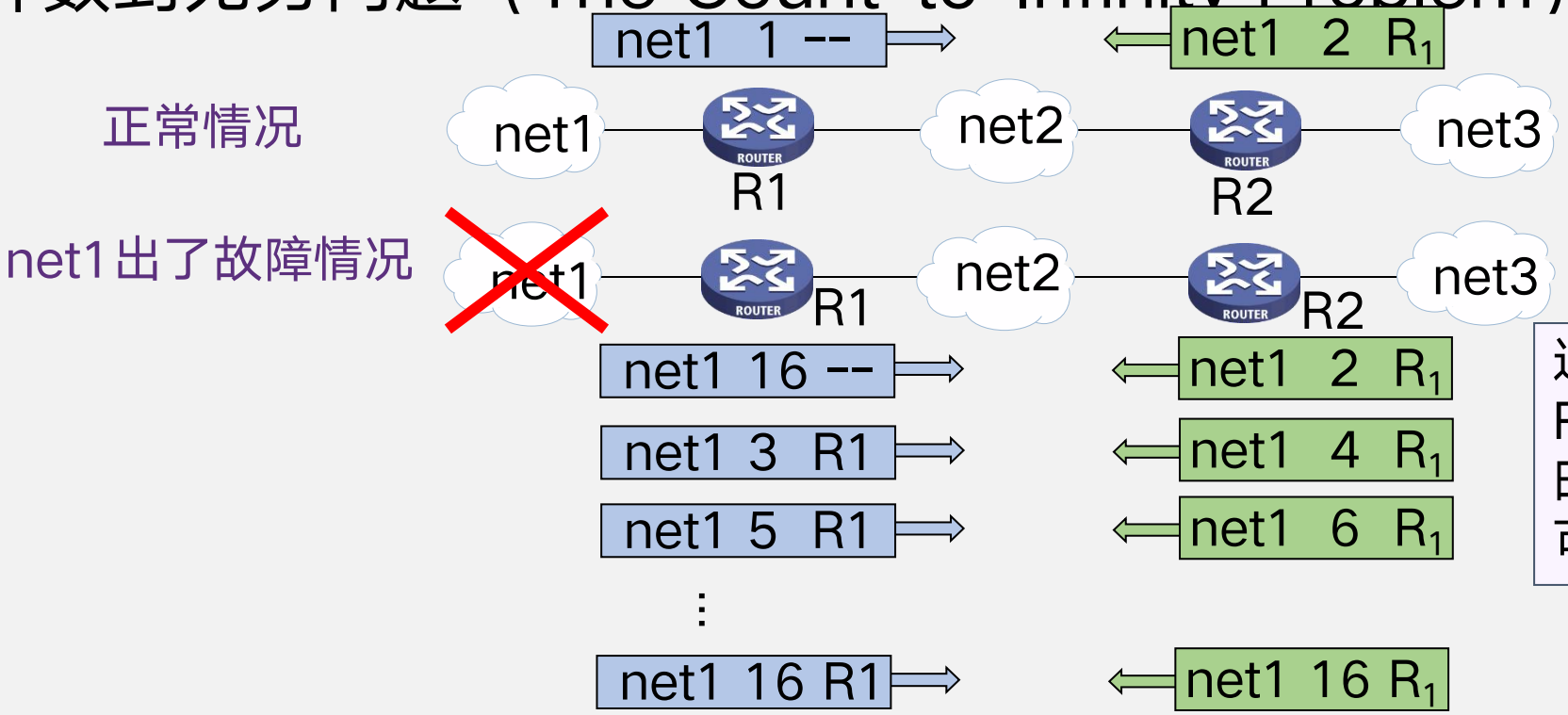
- 计数到无穷问题（The Count-to-Infinity Problem）



R_2 以后又更新自己的路由表为“net1, 4, R_1”，表明 “我到net 1 距离是 4，下一跳经过 R_1”。

距离向量路由

- 计数到无穷问题（The Count-to-Infinity Problem）



这样不断更新下去，直到 R_1 和 R_2 到net 1 的距离都增大到16时， R_1 和 R_2才知道net 1 是不可达的。

➢好消息传播快，坏消息传播慢，是距离向量路由的一个主要缺点

DSDV

- 目的节点序列距离矢量路由协议
  - 路由算法：改进的距离矢量算法
  - 在路由表中加入一个带有目的序列号的项，确保无路由回路问题
  - 改进对拓扑变化的反应速度：
    - 当路由表发生重大变化时立即重新启动路由发现
    - 对于不稳定的路由通告，节点延迟一段时间再发送路由更新信息，减缓路由波动

DSDV

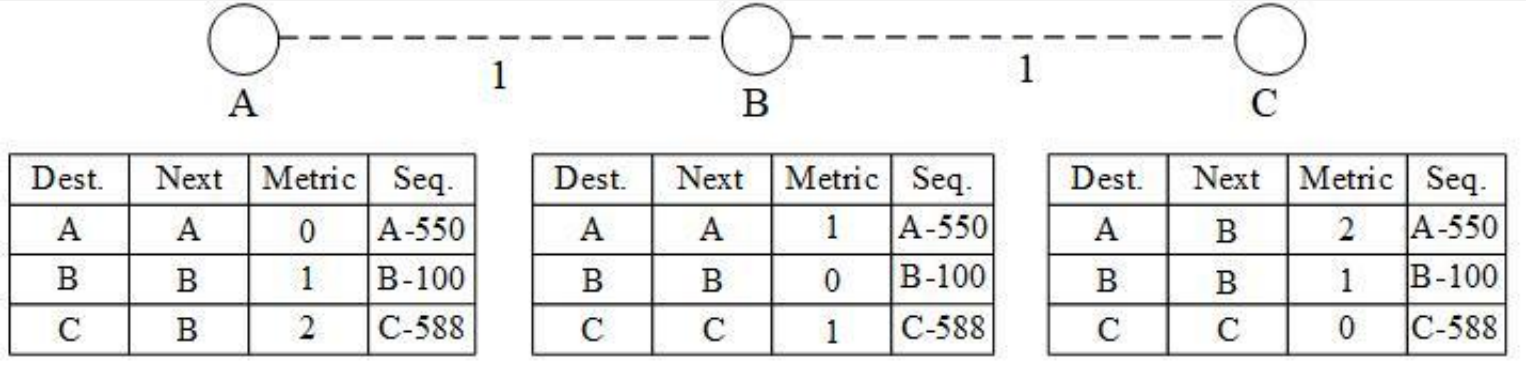
- 目的节点序列距离矢量路由协议
  - DSDV路由算法在距离矢量算法的路由表中增加Sequence Number、Install Time、Stable Data三个表项
    - （1）Sequence Number (Seq.Nr)：目的主机的编号。由目标节点生成，用来保证不出现路由回路
    - （2）Install Time：用来表示该路由表项的创建时间，用来删除表中过时的路由信息
    - （3）Stable Data：指向一张路由表(广播表)的指针，用来保存路线的稳定性

Destination	Next	Metric	Seq.Nr	Install Time	Stable Data
A	A	0	A-550	001000	Ptr_A
B	B	1	B-102	001200	Ptr_B
C	B	3	C-588	001200	Ptr_C
D	B	4	D-312	001200	Ptr_D

DSDV路由表结构

DSDV

- 目的节点序列距离矢量路由协议
  - 初始状态
    - 根据DSDV路由表结构特点，每个节点向邻节点广播自己的路由信息，网络拓扑和DSDV路由表项
    - 广播的路由信息包含：目的地址（Dest.）、下一跳地址（Next）、到达目的地址的跳数（Metric）和目的地址序列号（Seq.）



Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-100
C	B	2	C-588

Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-100
C	C	1	C-588

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-100
C	C	0	C-588

DSDV路由表项

DSDV

- 目的节点序列距离矢量路由协议
  - DSDV路由信息更新
    - 目的地址序列号的设置规则

      ① 每次广播将自己的目的序列号加2

      ② 如果一个节点不可到达，将这个节点的序列号加1并且设置Metric为无穷大

    - 以B节点为例，说明发送路由更新信息的过程

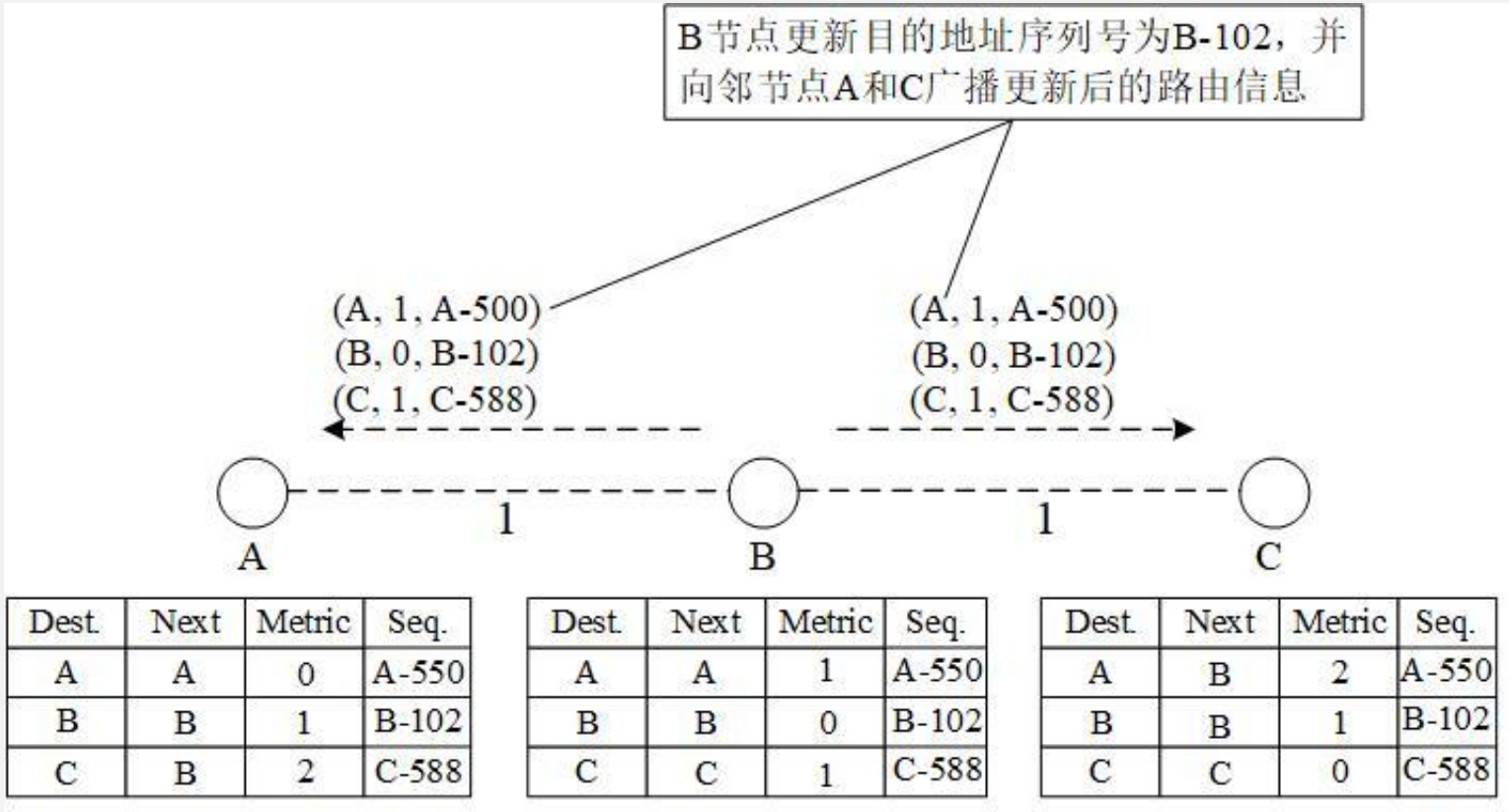
      ① 首先，B节点将自己的序列号从B-100增加到B-102；

      ② 其次，B节点广播自己的路由信息到邻节点A和C

判断节点可达性

DSDV

- 目的节点序列距离矢量路由协议
  - DSDV路由信息更新（续）



Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-102
C	B	2	C-588

Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-102
C	C	1	C-588

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-102
C	C	0	C-588

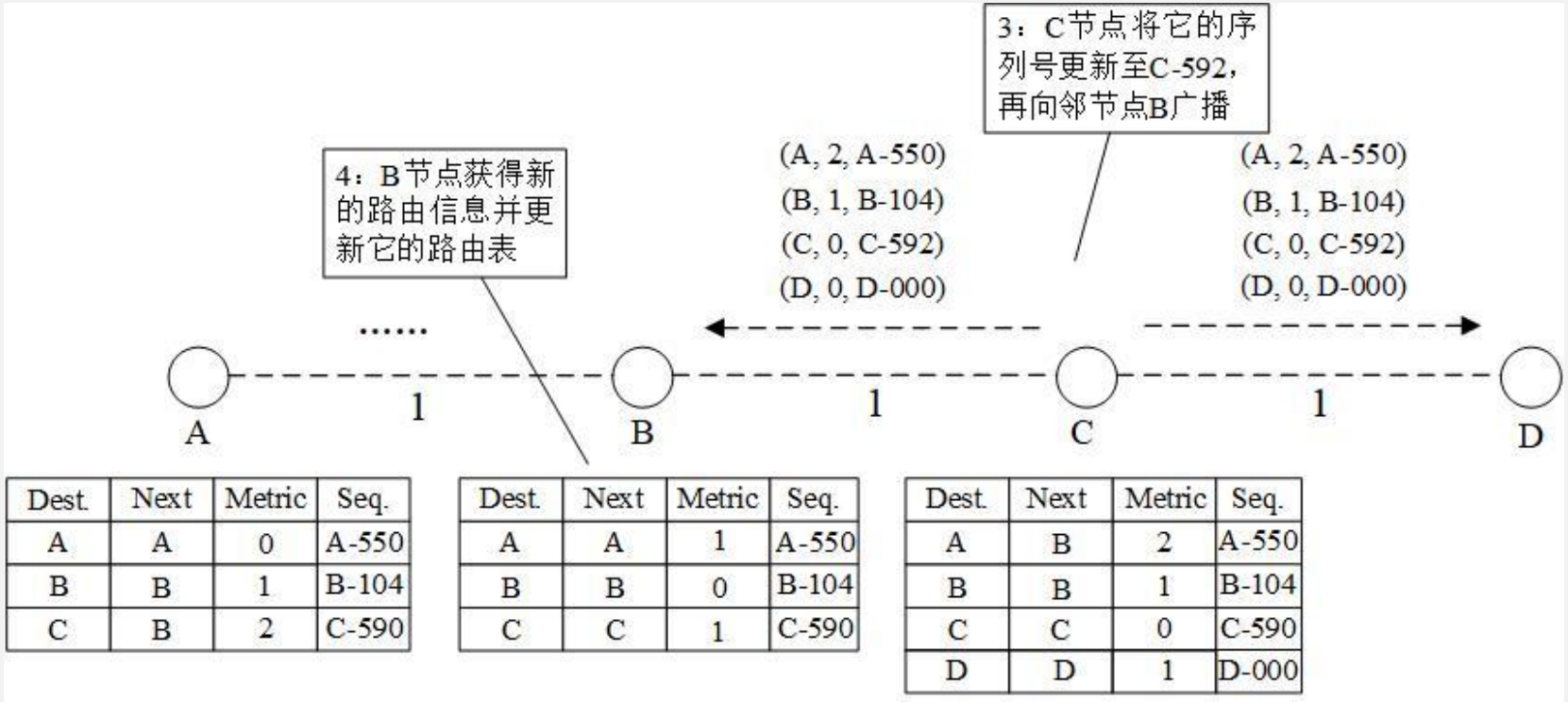
B节点发送DSDV路由更新信息

DSDV

- 目的节点序列距离矢量路由协议
  - DSDV路由信息更新（续）
    - 为了维护路由表的更新，节点周期性地广播路由更新数据包
    - 邻节点收到路由更新数据包后，先与自己的路由表进行比较，选择目的序列号大的路由
    - 如果目标序列号相同，选择Metric中跳数较小的路由

DSDV

- 目的节点序列距离矢量路由协议
  - DSDV路由信息更新（续）
    - 新节点加入时路由信息的更新过程



Dest.	Next	Metric	Seq.
A	A	0	A-550
B	B	1	B-104
C	B	2	C-590

Dest.	Next	Metric	Seq.
A	A	1	A-550
B	B	0	B-104
C	C	1	C-590

Dest.	Next	Metric	Seq.
A	B	2	A-550
B	B	1	B-104
C	C	0	C-590
D	D	1	D-000

新节点D加入网络（问题图，参考课堂讲解的书本勘误）

DSDV

- 目的节点序列距离矢量路由协议小结
  - 优点
    - 简单→由经典的B-F路由算法改进而来
    - 有效防止路由环路的发生
    - 延时性低
  - 缺点
    - 耗能问题严重→节点不能休眠（未考虑电池供电）
    - 产生大量网络开销→路由表中大部分的路由信息是从来不使用的

开销随着网络规模N的变化呈**爆炸式增长**

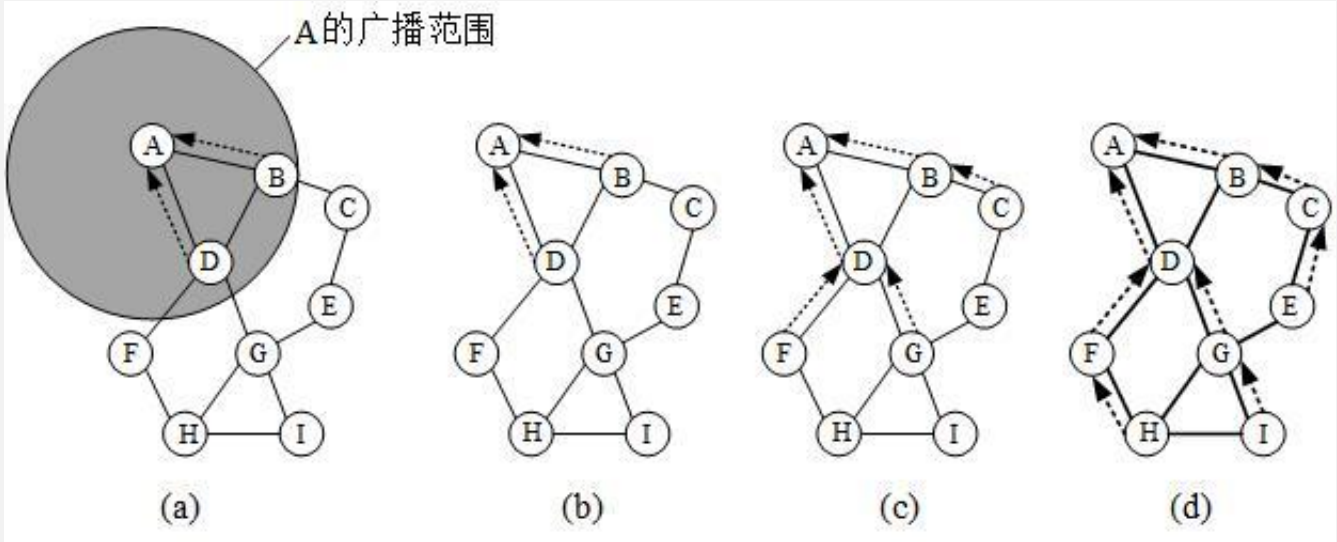
DSDV路由并不适用于节点数目较多能耗要求高的网络

AODV

- 无线自组网按需平面距离向量路由
  - 基于DSDV协议
    - 加入按需机制改进而来：只有向某个目标发送分组时，才计算路由
    - 考虑了带宽有限和电源寿命较短的限制
    - 工作于移动环境
    - 不需要维护全网路由信息
  - AODV算法思路
    - 路径发现
    - 路径维护
    - 路由信息新旧判断
    - 拥塞控制

AODV

- 无线自组网按需平面距离向量路由
  - 路径发现
    - A 要向 I 发送分组。A 检查表项里没有去往 I 的表项，于是按需计算到 I 的路径
    - A 构造 ROUTE REQUEST（RReq）分组（寻找 A→I 的路由）



AODV算法示意图

(a) A的广播范围

(b) B和D接收之后

(c) C、F和G接收后

(d) E、H和I接收后

（阴影节点是新的接收者，虚线表示可能的逆向路由，实线表示发现的路由）

Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

Route Request分组结构

AODV

- 无线自组网按需平面距离向量路由
  - 路径发现（续）
    - A 要向 I 发送分组。 A 检查表项里没有去往 I 的表项，于是按需计算到 I 的路径
    - A 构造 ROUTE REQUEST（RReq）分组（寻找 A→I 的路由）
      - 源地址
      - 请求 ID 。每个节点单独维护的本地计数器。（源地址 + 请求 ID）唯一标识一个RReq 分组
      - 目标地址
      - 源序列号：类似时钟，区分新老路径
      - 目标序列号：源节点所知道的目的节点的最近值
      - 跳计数：分组经过的跳数

AODV

- 无线自组网按需平面距离向量路由
  - 路径发现（续）
    - 当 ROUTE REQUEST 分组到达某节点后
      - 在本地的历史表中查找（源节点，请求 ID）对，如果是重复分组，则丢弃，处理结束；如果不是重复分组，将（源节点，请求 ID）写入历史表，以便将来识别重复分组
      - 在路由表中查找该分组的目标地址，如果找到较新的通往该目标的路径的话（路由表目标序号大于等于 Rreq 分组中的目标序号），给源节点送回一个 ROUTE REPLY 分组

AODV

- 无线自组网按需平面距离向量路由
  - 路径发现（续）
    - 当 ROUTE REQUEST 分组到达某节点后
      - 如果在路由表中没有找到较新的路径，增加跳计数，重新广播 ROUTE REQUEST 分组，同时从分组中提取数据，构造逆向路由表，便于以后应答分组可以回到源节点。同时启动定时器，定时器到期的话，该表项被删除
      - 如果沿途都没有节点知道目标节点的路径，则 ROUTE REQUEST分组 最终到达目标节点，目标结点向源结点发送 ROUTE REPLY 分组作为应答

AODV

- 无线自组网按需平面距离向量路由
  - 路径发现（续）
    - 回程中每个节点检查 Route Reply 分组。如果符合以下条件之一，将分组信息加入本地路由表，作为到达目的地址的一条路由
      - 如果目前还没有通往该目的节点的路由路径
      - 在 Route Reply 中的目标序列号比本地路由器中的序列号大（更新的消息）
      - 序列号相等，但是新的路径更短
    - 将该 Route Reply 分组沿着逆向路径向上游节点转发，直至达到源节点

Source address	Destination address	Destination sequence #	Hop count	Lifetime

Route Reply分组结构

AODV

- 无线自组网按需平面距离向量路由
  - 路径维护：拓扑结构变化时要能处理
    - 节点N定期的向邻居广播HELLO消息，希望邻居做出应答，如果没有应答，则认为该邻居之间的连接已经断开
    - N在它的路由表查找：哪些目标的路由路径用到了该不可达的邻居，对于每一条这样的路径，N都要通知对应的活动邻居，告诉它们经过N的路径不再有效，应该从路由表中清除
    - 递归，直到所有依赖于该失效节点的路径全部从路由表中清除为止

AODV

- 无线自组网按需平面距离向量路由
  - 拥塞控制
    - 源节点在发送RREQ后，在规定的时间内没有收到来自目的节点的RREP时，可以选择再次发送RREQ路由请求帧
    - 在尝试了RREQ_RETRIES次之后，如果依旧收不到RREP，则在路由表中标记该目的节点不可达，并通知应用层
    - 每次在重新发送RREQ请求帧时，等待RREP应答帧的时间要在原来时间的基础上乘以2，避免拥塞

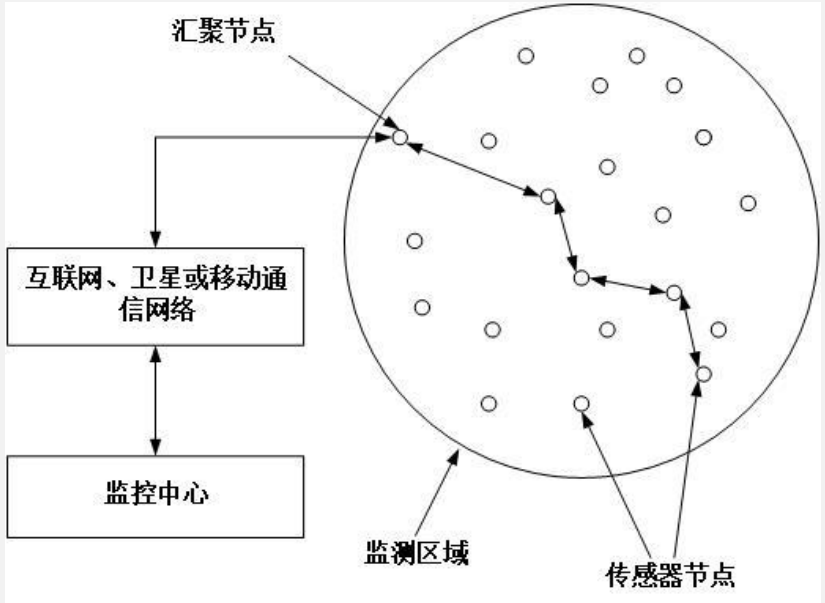
AODV

- 无线自组网按需平面距离向量路由协议小结
  - 优点
    - ① 扩展性能强大
    - ② 每个节点拥有唯一的目的序列号，可以避免路由环路
    - ③ 能够快速修复失效路由
    - ④ 路由协议简单
    - ⑤ 由于中间节点参与路由发现过程，使得源节点向邻节点广播的次数较少

按需协议路由有助于
节省带宽和电池寿命

WSN路由

- WSN概述
  - WSN由部署在检测区域内大量的微型传感器节点组成，通过无线的方式形成一个多跳的自组织网络
  - WSN是一种特殊的Ad hoc网络，由无线传感器节点协同组织起来
- WSN典型体系结构
  - 分布式传感器节点
  - 汇聚节点
  - 互联网
  - 监控中心



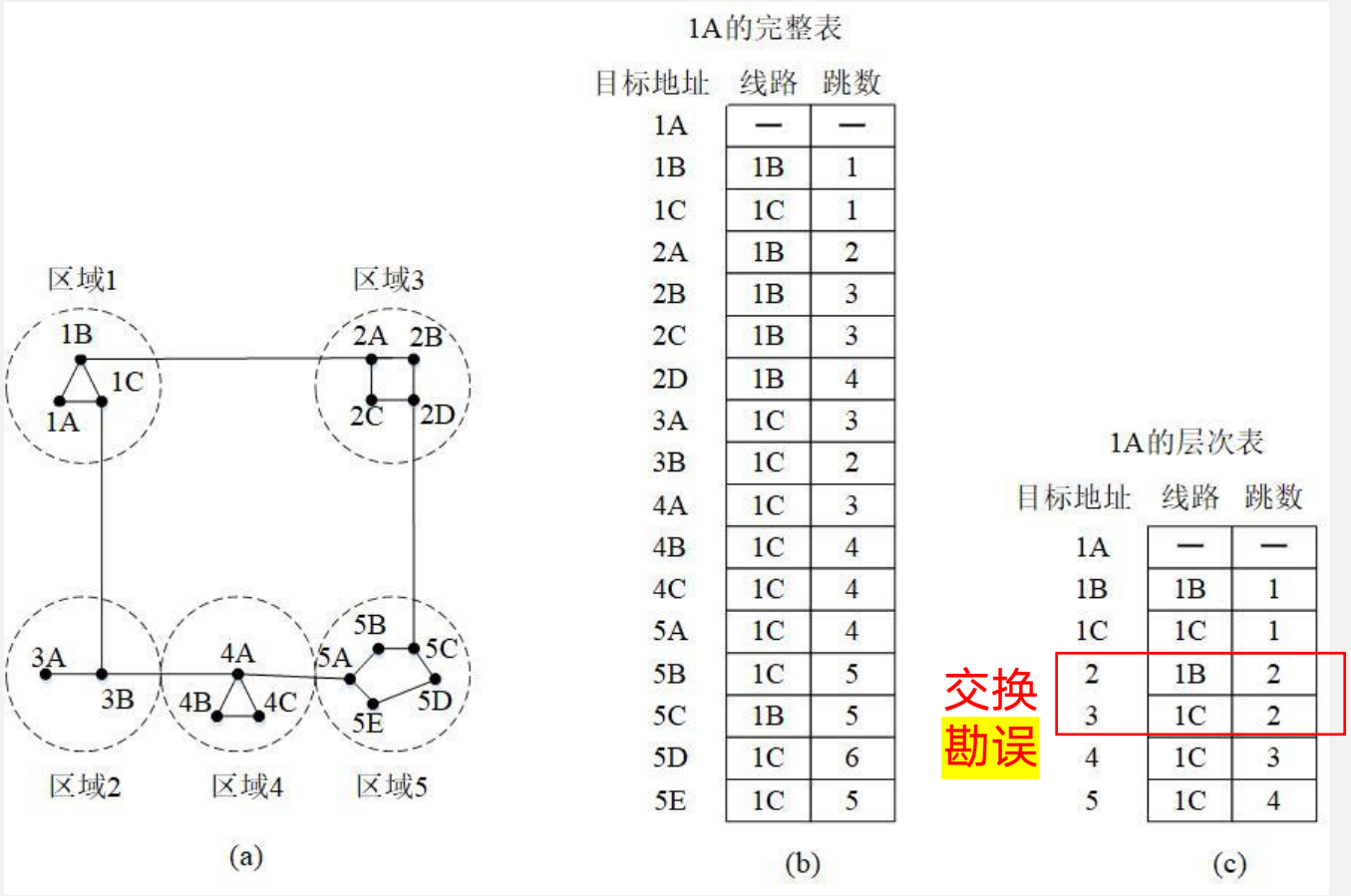
WSN体系结构

WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型协议（Low Energy Adaptive Clustering Hierarchy，LEACH）
    - 分层路由→节点被划分成区域→每个节点只知道如何将数据包路由到自己所在区域内的目标地址→一个网络中的节点不必知道其他网络的拓扑结构
    - 对于大型网络，两级的层次结构可能还不够；可能有必要将区域组织成簇，将簇组织成区，将区组织成群，等等，直到将所有的集合名词用完为止

WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型协议
    - 两级的层次结构
      - 节点1A的完整路由表有17个表项，如图(b)
      - 采用分级路由，路由表如图(c)
    - 分层协议优点
      - 节省路由表空间
    - 分层协议缺点
      - 增加路径长度

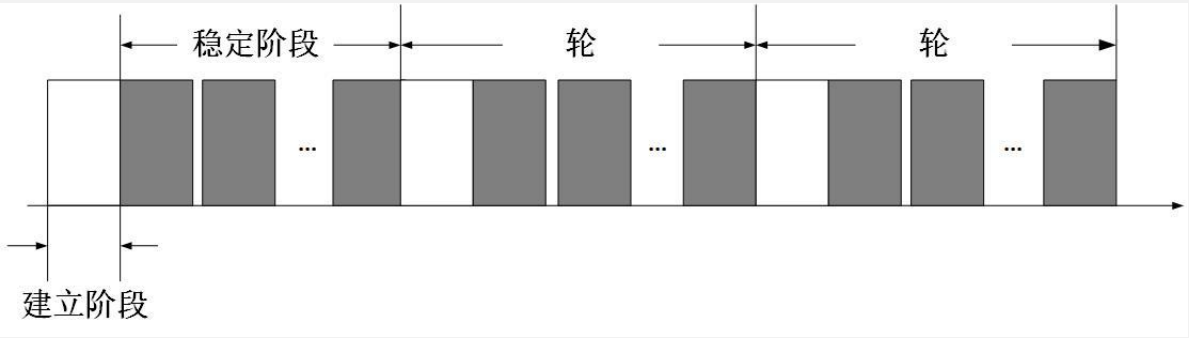


(a) 两级层次网络；(b) 节点1A的完整路由表；(c) 节点1A的层次路由表

层次路由

WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型
    - 网络生命周期延长15%（仿真，与多跳路由算法比）
    - 簇的重构过程
      - 簇建立阶段和稳定运行阶段
      - 簇建立阶段和稳定运行阶段所持续的时间总和为一轮



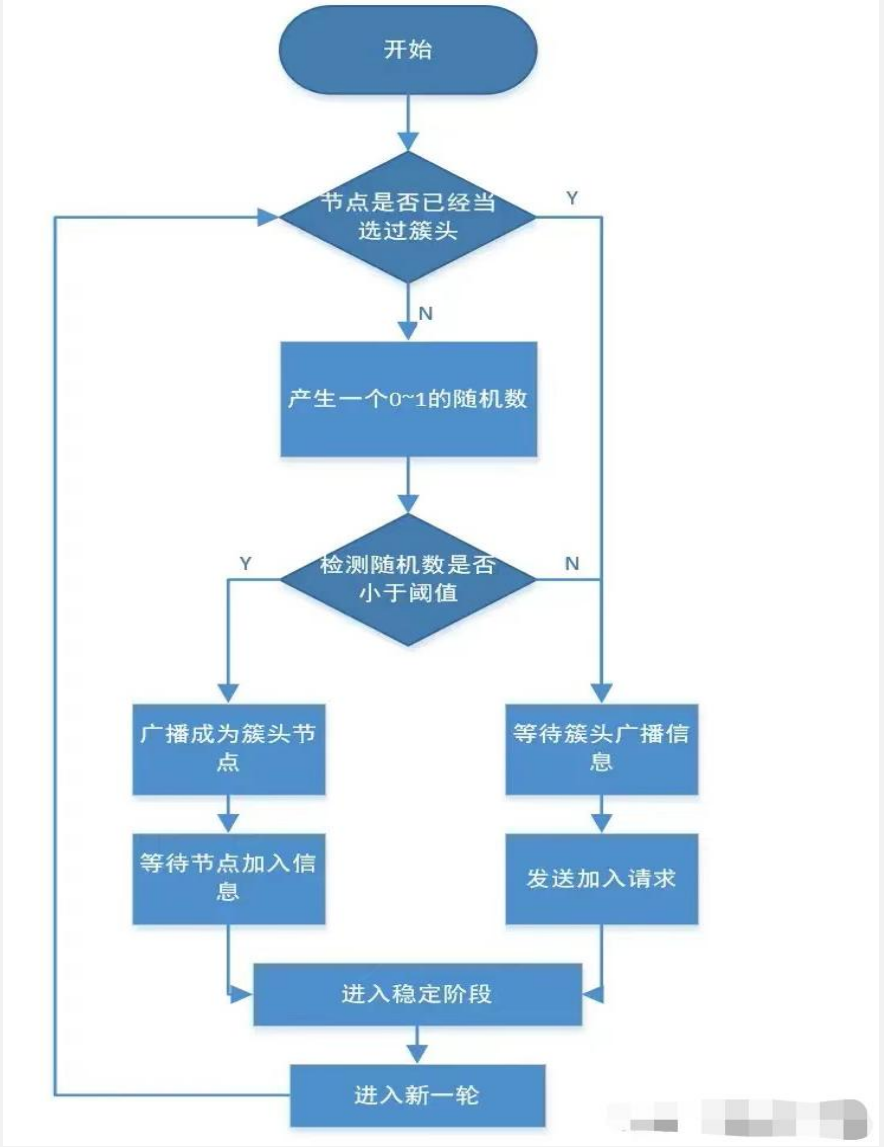
LEACH算法中的工作循环

WSN路由

低功耗自适应集簇分层型

簇建立阶段

- 簇首节点的选择、簇首节点的广播、簇首节点的建立和调度机制的生成
- 簇首节点的选择办法：每个传感器节点随机选择0-1之间的一个数。如果该随机数小于阀值T(n)，那么这个节点成为簇首节点
- 在每轮循环中，若节点已经当选过簇头，则将T(n)设置为0
- 若未当选过簇头的节点，将以T(n)的概率当选
- 随着当选过簇头的节点的数量增多，剩余节点当选簇头的阈值T(n)也随之增大，节点产生小于T(n)的随机数的概率随之增大，所以节点当选为簇头的概率也增大
- 当只剩余一个节点未当选时，T(n)=1，表示该节点一定当选



WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型
    - 簇建立阶段
      - T(n)可表示为

$$T(n)=\begin{cases}\dfrac{P}{1-P\times\left[r \bmod \left(\dfrac{1}{P}\right)\right]}, & n\in G\\ 0, & \text{其他}\end{cases}$$

式中，P为簇头在所有节点中所占的百分比（预设值），r是当前轮数，$r \bmod \left(\frac{1}{p}\right)$表示这一周期循环中当选过簇头的节点个数，$G$是这一周期循环中未当选过簇头的节点的集合

      - 节点当选簇头后，通过广播告知整个网络；信号强度决定从属簇
      - 簇头产生一个TDMA定时信息，为簇中的每个成员分配通信时隙

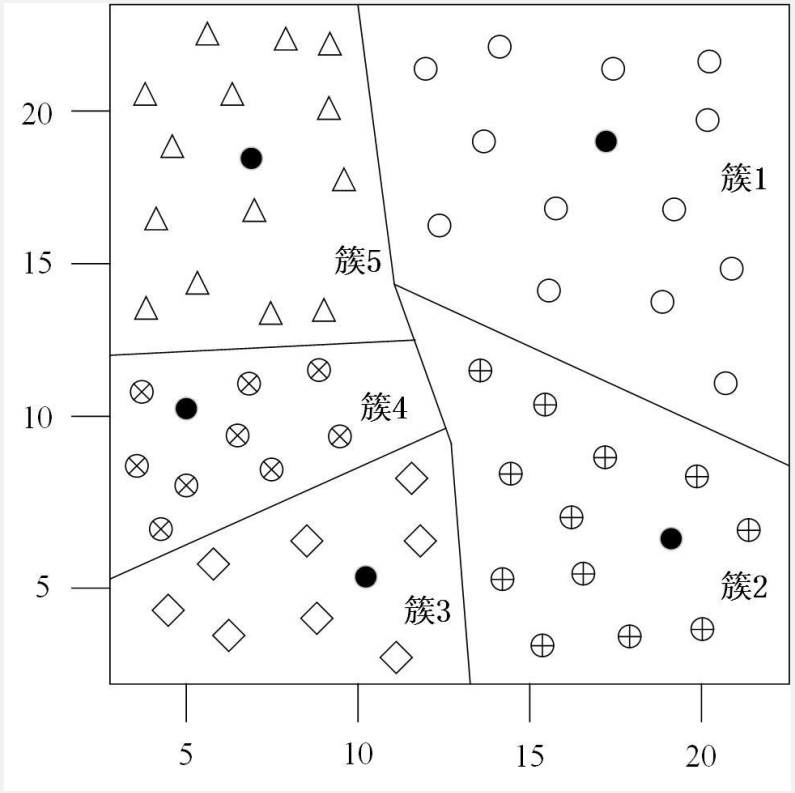
W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," in *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, Oct. 2002, doi: 10.1109/TWC.2002.804190.

WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型
    - 稳定运行阶段
      - 传感器节点将持续采集检测数据并传送到簇首节点
      - 簇首节点对簇中所有节点所采集的数据进行信息融合后再传送给汇聚节点
      - 稳定阶段持续一段时间后，网络重新进入簇的建立阶段，进行下一轮的簇重构
      - 为了避免附近簇的信号干扰，簇头可以决定本簇中所有节点所用的CDMA编码

WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型
    - 稳定运行阶段



LEACH算法中的簇的划分

WSN路由

- WSN常用路由协议
  - 低功耗自适应集簇分层型（Low Energy Adaptive Clustering Hierarchy，LEACH）
    - 优点
      - 利用将区域划分成簇，簇内本地化协调和控制的形式有效的进行数据收集；独特的选簇算法（随机轮换）；首次运用了数据融合的方法
    - 缺点
      - 簇头消耗能量比较大，是网络中的瓶颈
      - 簇头选举是随机循环选举。簇头选举没有根据节点的剩余能量以及位置等因素，会导致有的簇过早耗尽电量
      - 网络扩展性不强

- 本章主要内容：集中式路由算法、分布式路由算法，自组织网络路由协议，无线传感网络路由协议。
- 本章学习目标
  - 了解集中式路由和分布式路由算法;
  - 熟悉自组织网络路由协议，掌握DSDV和AODV协议;
  - 熟悉无线传感网络路由协议，掌握LEACH协议。

  课后题：6,10