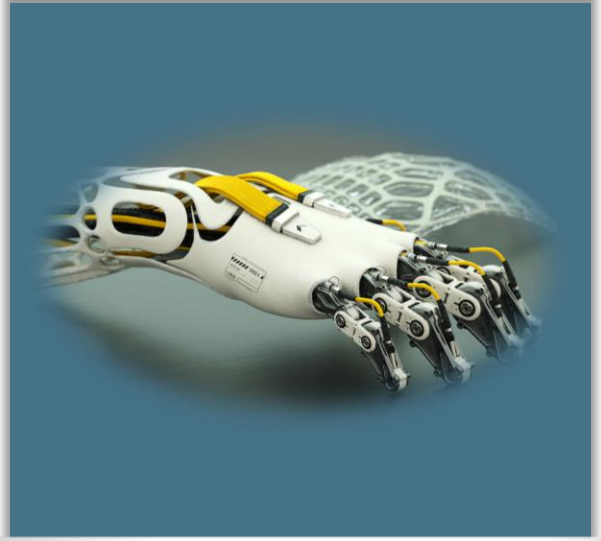
智能感知与物联网

授课人：王闻博

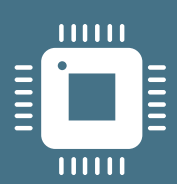
Email: wenbo_wang@kust.edu.cn

昆明理工大学 机电工程学院

2026年3月10-4月10日



第二章
物联网接入与组网




2.1 无线网络基础

2.2 TCP/IP协议简介

2.3-2.4 近距离和中远距离无线通信

回顾：物联网解决方案的功能层划分

数据和服务

资源和任务管理

安全

业务层

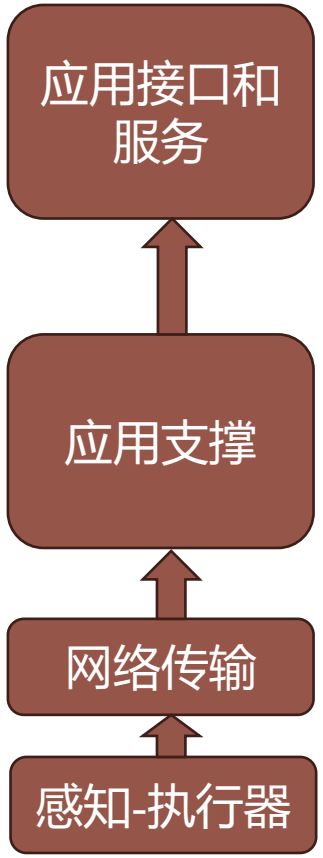
应用层：运营平台、信息中心、内容服务，etc…

数据和信息层：管理系统及数据库、智能处理、云计算，etc…

服务驱动层（中间件）

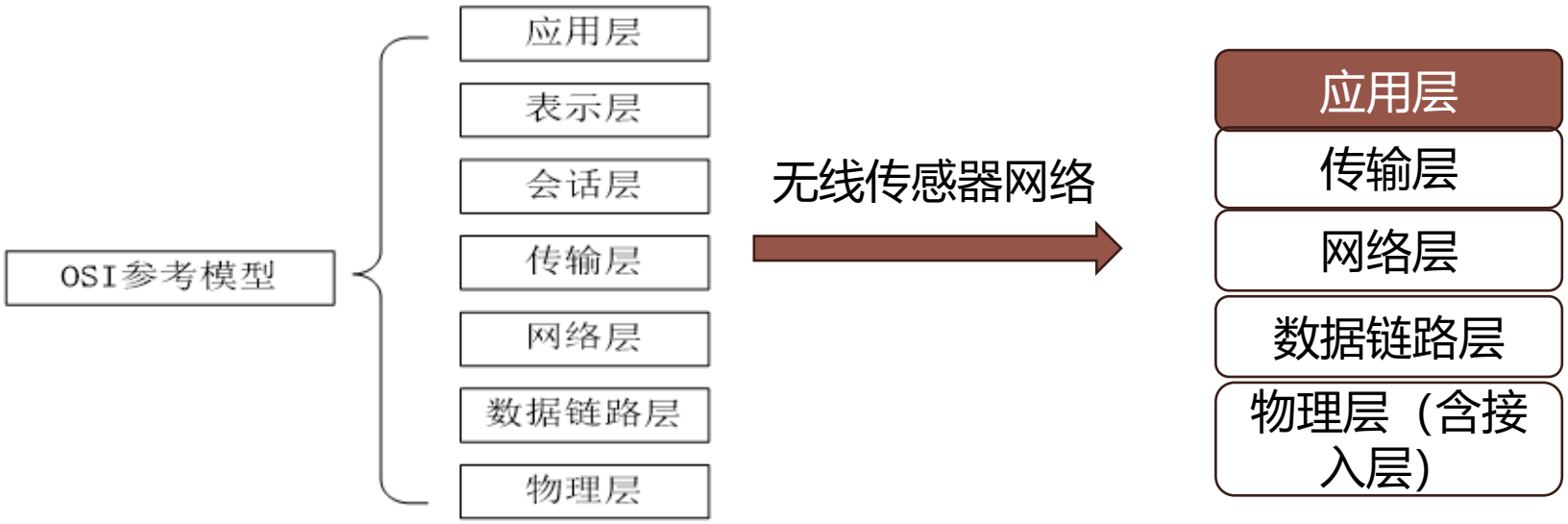
通信层：无线网络、RFID、卫星，etc…

设备和资源层：传感器、电源管理，etc…



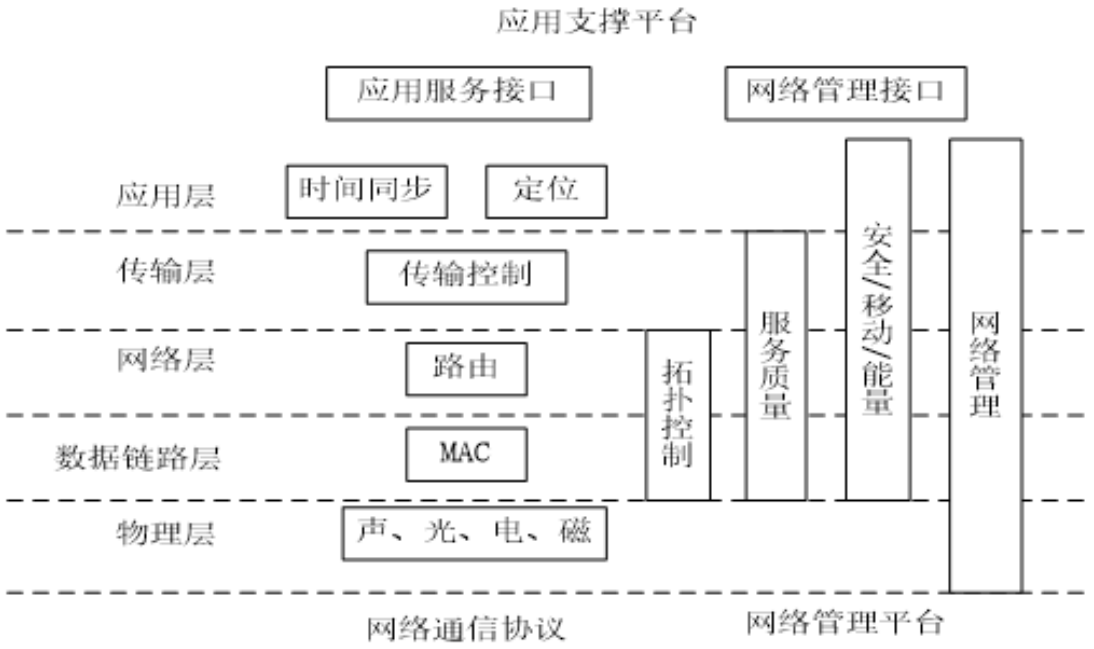
从传感器网络到IoT：无线网络架构基础

- 聚焦物联网功能划分中的设备层和通信层（网络层）
- 传统网络协议OSI参考模型（协议层模型）
  - 开放式系统互联网络参考模型（OSI）共有7个层次，从底向上依次是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。设计中，除物理层和应用层外，其余每层都和相邻上下两层进行通信。



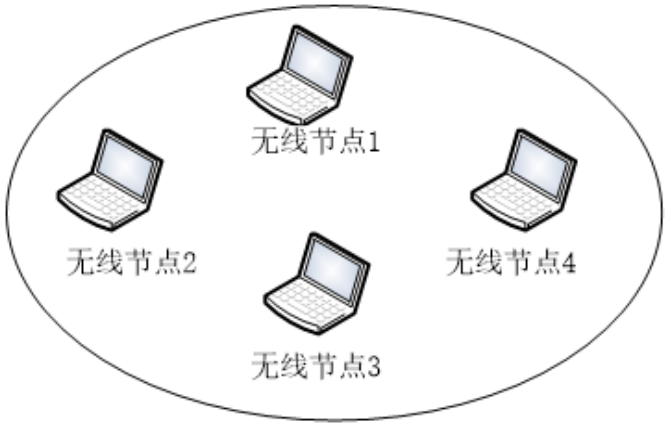
（续）传感器网络协议栈

- 物理层：为终端设备提供物理传输通路，建立传输规则。主要任务包括建立传输规范、选择给定传输介质下的信道、确定数据收发率等。典型协议有802.11（WiFi）及802.15.4（Zigbee）协议等。
- 数据链路层：建立可靠的点对点（点到多点）链路，并控制信道分配、负责数据流的的多路复用、数据成帧与帧监测、差错控制和功率控制等。
- 网络层：将数据从传感器节点可靠地传输到汇聚节点（Sink）。主要任务有路由发现、分组路由、网络互联、拥塞（Congestion）控制。
- 传输层：进行数据流的传输控制、完成数据格式转换。典型的传输层协议有TCP和UDP协议等。
- 应用层：面向用户提供的网络服务，如时间同步服务、节点定位服务等。

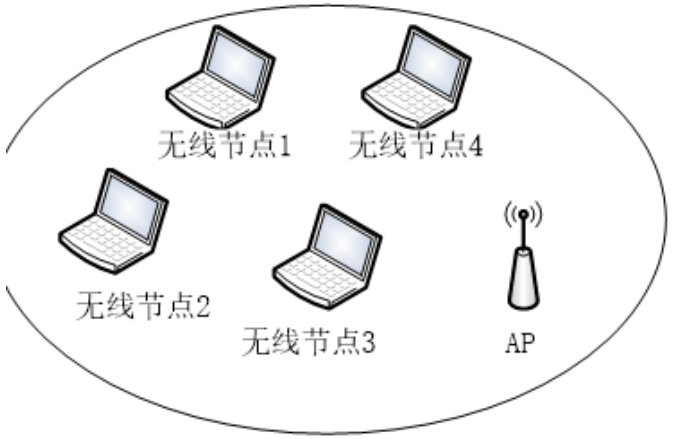


典型传感器网络拓扑结构（以802.11下网络为例）

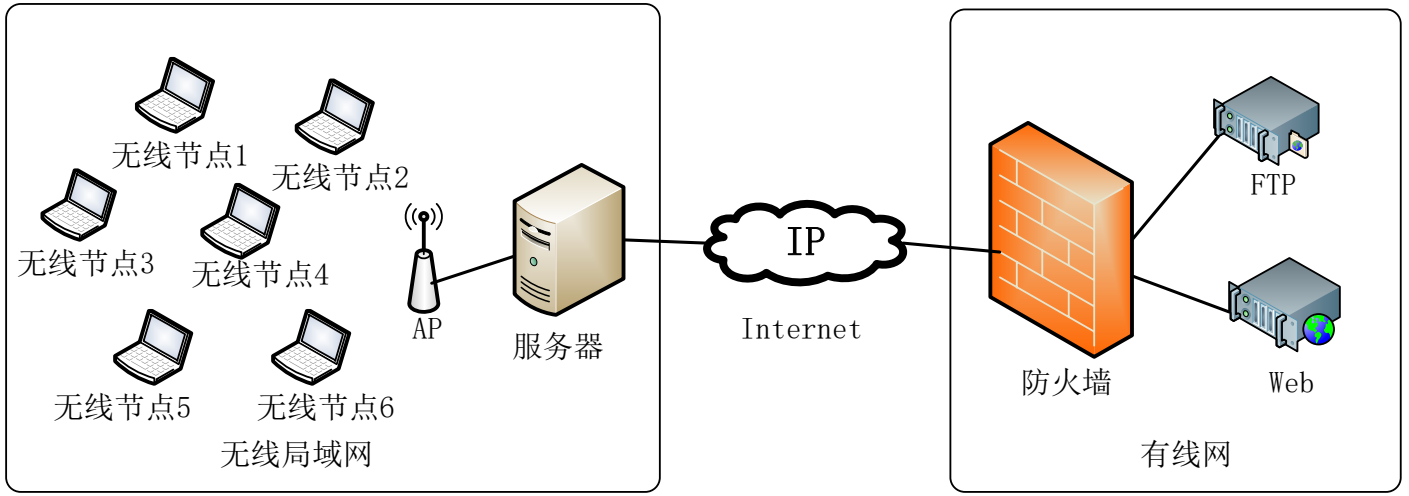
1. Ad-hoc结构



2. 基础结构（Infrastructure-based）网络

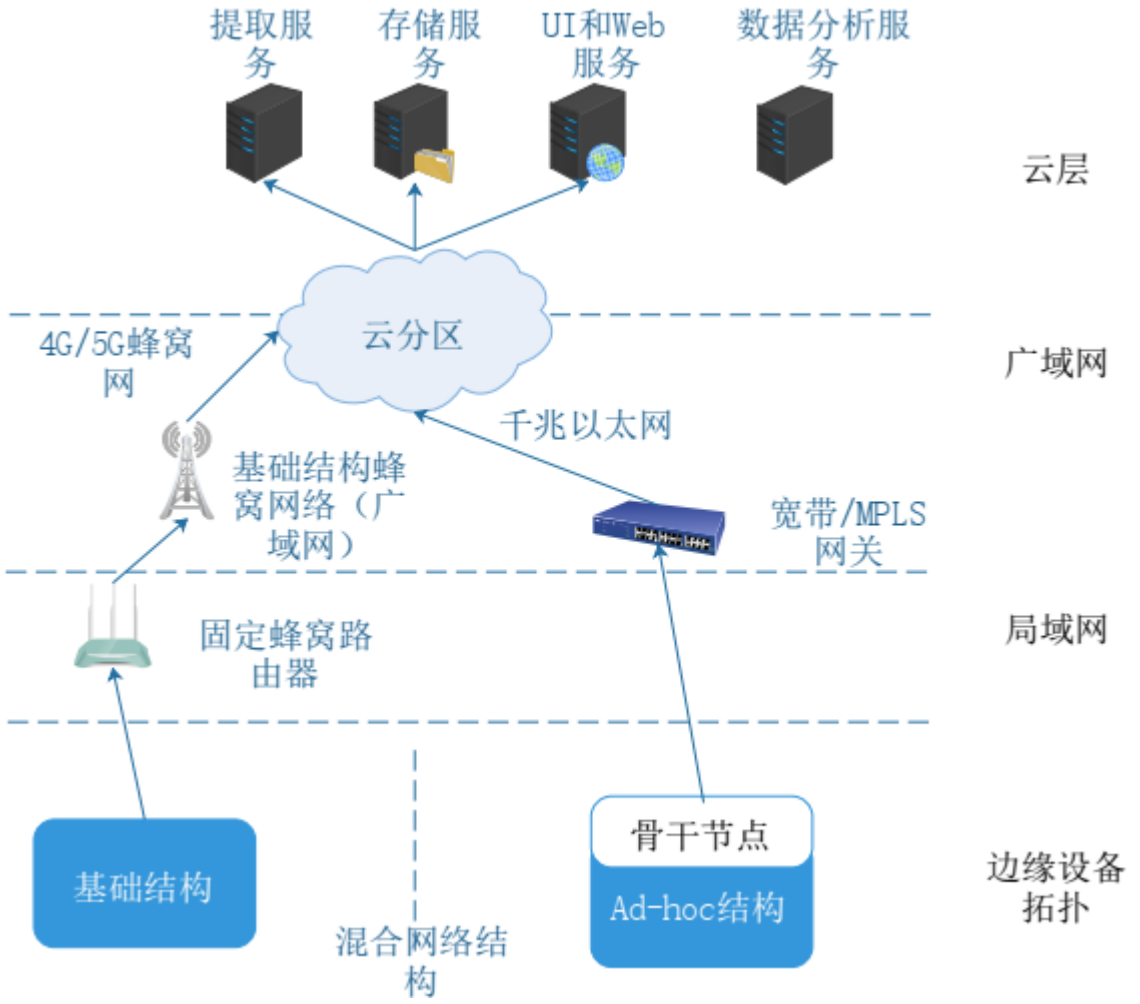


3. 扩展服务集网络



IoT网络对传感器网络的扩展

- 基于扩展服务式网络结构的扩展
  - 传感器、边缘设备可能基于多种结构（Ad-hoc、Mesh、分层/主-从）组成非IP的边缘网络。
  - 路由器、网关为边缘网络数据的汇聚节点。
  - 广域网络由无线蜂窝、有线网或卫星网络提供商提供，并可能基于TCP/IP协议为云端服务提供面向消息的中间件（Message-Oriented Middleware，MOM）通信协议，从而提供应用或服务间的解耦。
  - 相比传感器网络的单一数据采集功能，IoT网络面向有监督控制和数据采集（**S**upervisory **C**ontrol **a**nd **D**ata **A**cquisition，SCADA），通过通信层连接SCADA的数据采集层和监控与工厂管理服务，从而将数据存储-分析-决策服务有条件地部署在云端。

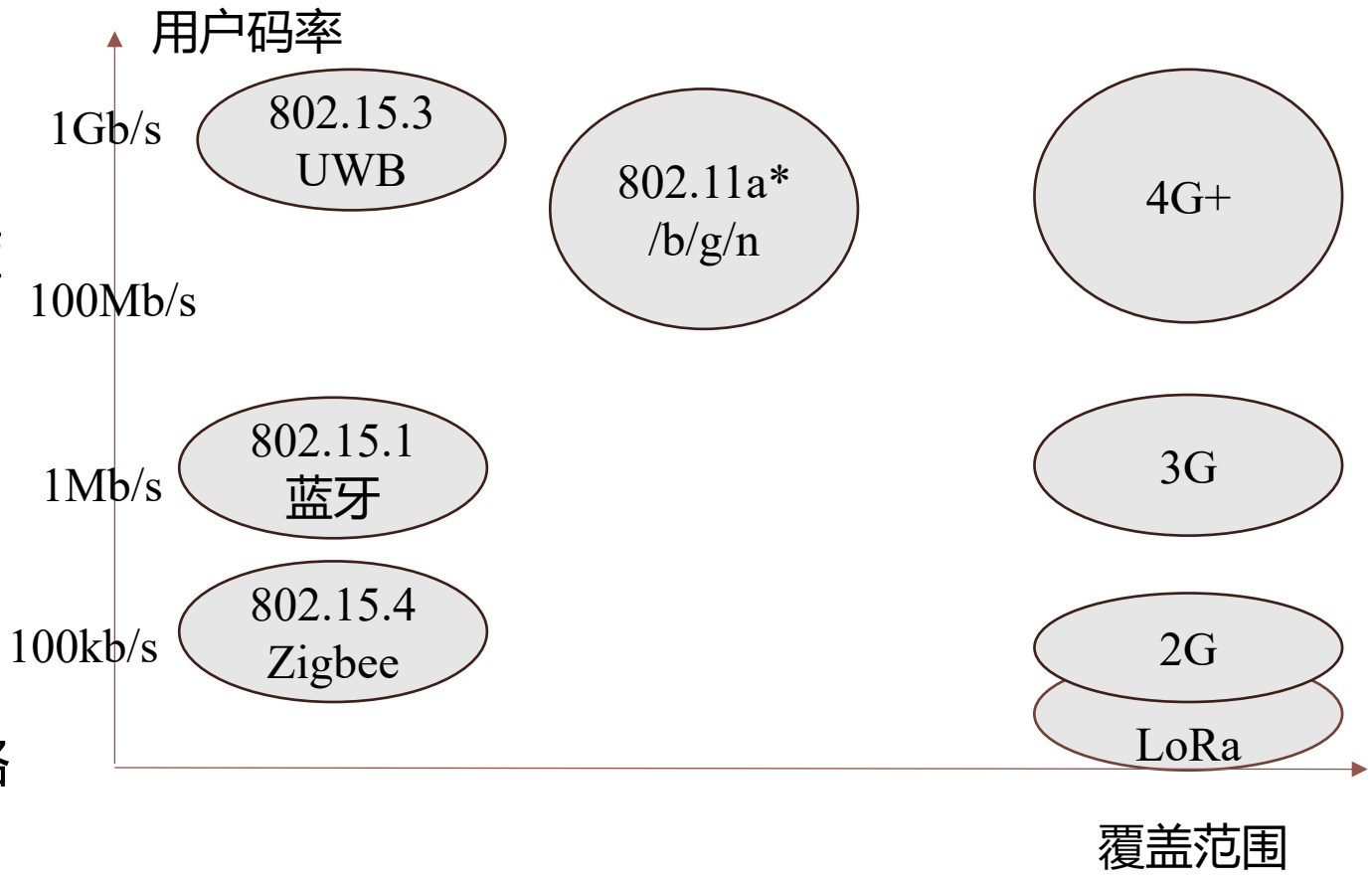


补充：（工业）IoT与现场总线的区别

- 现场总线是指以工厂内的测量和控制机器间的数字通讯为主的网络，也称现场网络
  - 是将传感器、各种操作终端和控制器间的通讯及控制器之间的通讯进行特化的网络。
  - 现场总线指安装在制造或过程区域的现场装置与控制室内的自动装置之间的数字式、串行、多点通信的数据总线。
- 现场总线的特点
  - 现场总线作为一种网络形式，专门为实现在严格的实时约束条件下工作而特别设计的。
  - 由于严格的实时性要求，这些现场总线的网络构成通常是有线的。
  - 由于现场总线通过报告传感数据控制物理环境，所以从某种应用目的上说它与工业IoT网络一致，所以可以将工业IoT网络看作是无线现场总线的实例。

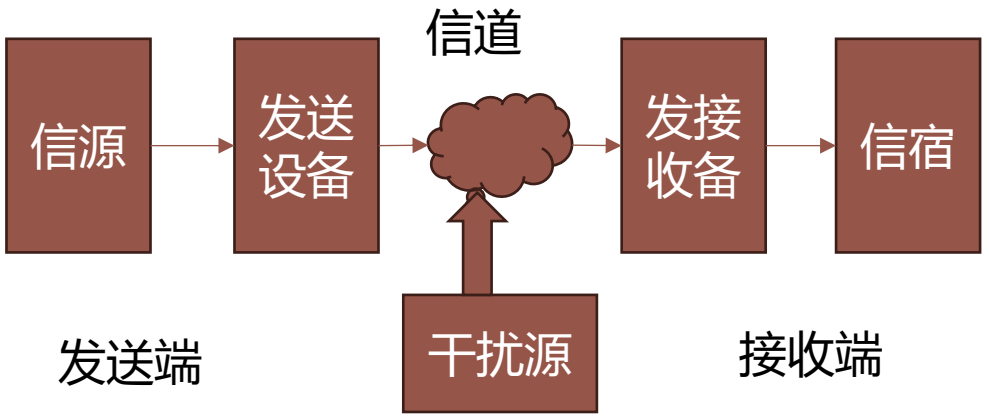
IoT相关无线通信协议概览

- IoT相关无线通信技术（中远距离）划分
  - 无线个人区域网（WPAN）：包括蓝牙、Zigbee、超宽带机器间通信（UWB M2M）等。
  - 无线局域网（WLAN)：802.11 WiFi。
  - 宽带无线接入（WMAN）：WiMax。
  - 蜂窝无线系统（WAN）：4G、5G。
  - LoRaWAN：Long Range，LoRa，由众多设备和少量网关组成的星型网络通过线性扩频技术和基于ALOHA的多址接入协议完成组网。
  - 卫星通信：卫星电话、定位。



物理层和无线通信理论概览

- 物理层的基本概念
  - 现有无线网络中的物理设备和传输介质的种类非常多（如无线频谱、可见光、声音等），而通信手段也有许多不同的方式。引入物理层的作用正是要尽可能地屏蔽掉这些差异。
- 物理层的主要功能
  - (1) 为数据终端设备(Data Terminal Equipment，DTE)提供传送数据的通路；
  - (2) 传输数据；
  - (3) 其他管理工作。物理层还负责其他一些管理工作，如信道状态评估、能量检测等。



物理层的主要技术

无线通信物理层的主要技术包括介质的选择、频段的选择、调制技术和扩频技术（本节略）。

(1) 介质和频段选择

- 无线通信的介质包括电磁波和声波。电磁波（含各种频段的光波）是最主要的无线通信介质，而声波一般仅用于水下的无线通信；

(2) 调制技术

- 调制和解调技术是无线通信系统的关键技术之一。通常信号源的编码信息(即信源)含有直流分量和频率较低的频率分量，称为基带信号。
- 调制技术通过改变高频载波的幅度、相位或频率，使其随着基带信号幅度的变化而变化。解调是将基带信号从载波中提取出来以便预定的接收者(信宿)处理和理解的过程。

物理层原理：信号在介质中的传播和衰减

- 理想情况下信号从发射器到接收器的“视线”直线（Line of Sight，LoS）中传播
  - 目前无线传感器网络采用的主要传输介质包括无线电、红外线和光波等。
- 网络设计者需根据发射功率、收发机距离和介质频率选择物理层协议，保障信号能够被正确接收
  - 为此引入LoS下接收功率分析方程（各向同性天线情况下）

$$P_{rx} = P_{tx} G_{tx} G_{rx} \frac{\lambda^2}{(4\pi R)^2}$$

其中，下标tx/rx代表发射机transmitter和接收机receiver，λ 代表载波波长 λ =c/f，G_{tx} 和G_{rx}分别代表传输和接收天线增益，R 代表发送端和接收端的距离。

物理层原理：信号在介质中的传播和衰减

- 理想情况下LoS接收功率分析方程

$$P_{rx} = P_{tx} G_{tx} G_{rx} \frac{\lambda^2}{(4\pi R)^2}$$

- 公式解释：
  - 接收天线的有效接收面积（A_e）与其增益（G_{rx})通过以下公式关联：

$$A_e = \frac{G_{rx}\lambda^2}{4\pi}$$

  - 电磁波在自由空间中传播时，能量均匀扩散到以发射点为中心的球体表面$4\pi R^2$，所以接收天线获得的能量为：

$$P_{rx} = P_{tx} G_{tx} \frac{A_e}{4\pi R^2} = P_{tx} G_{tx} \frac{G_{rx}\lambda^2}{(4\pi R)^2}$$

信号在介质中的传播和衰减（续）

- 在没有任何信号增益情况下，增强接收信号只能通过:
  - 增加发射功率;
  - 降低损耗。
- 自由空间路径损耗（FSPL）的经验公式给出了无线信号直线传播的损失情况:

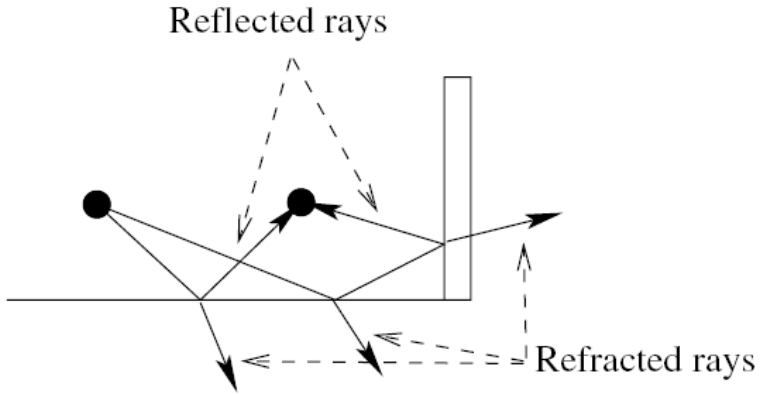
$$\mathrm{FSPL(dB)} = 10\log_{10}\left(\left(\frac{4\pi Rf}{c}\right)^2\right)$$

$$= 20\log_{10}\left(\frac{4\pi Rf}{c}\right) = 20\log_{10}R + 20\log_{10}f - 147.55$$

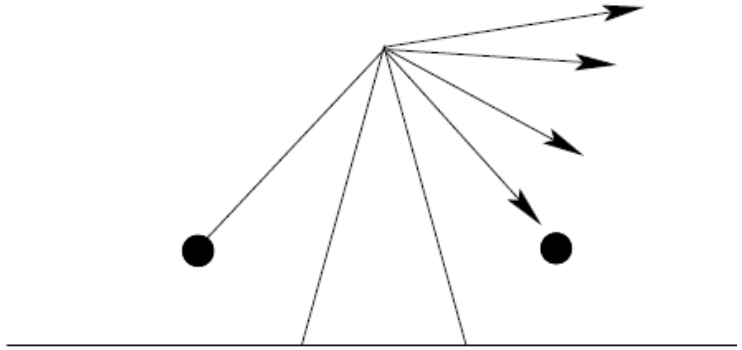
其中，影响因素包括信号频率f，发送端和接收端的距离R 和光速c。

非理想情况下无线信号传播受到的干扰

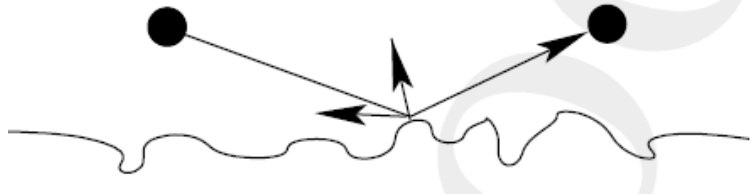
电磁波的多径传播：信号传播时遇到高大障碍物时会发生反射，遇到大障碍物（大于波长）时会发生衍射（绕射），遇到小于波长的障碍物时会发生散射。



反射和折射



衍射（绕射）



散射

无线信号在非视线（NLoS）传输中的衰减

(1) 慢衰落 （Slow Fading）

- 绕射能力弱的信号会在大障碍物后形成信号强度小的阴影区（Shadowing）。
- 同时，由信号源和接收器之间的相对运动，接收器端信号的频率会发生变化，称为多普勒扩展。

由阴影和多普勒扩展导致的长于相干时间的衰落称为慢衰落。

(2) 快衰落（Fast Fading）

- 由于衍射和反射造成来自不同路径信号“自干扰”的信号衰落效应，其可有如下冲击响应描述：

$$h(t) = \sum_{k=1}^{n} a_k \delta(t - \tau_k) \exp(j\theta_k)$$

其中， a_k是路径k上的信号幅度， τ_k是其上的信号到达时延， θ_k是其上的信号相位。快衰落多发于人口密集的都市区。

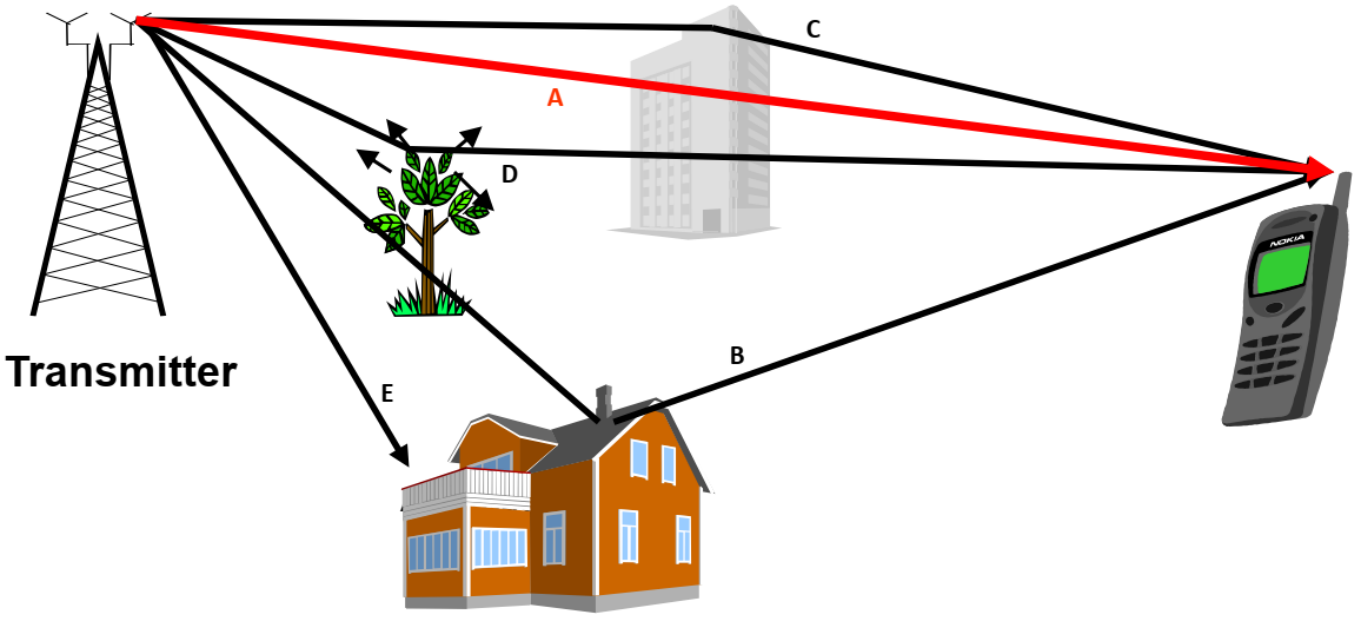
由快衰落（多径干扰）导致的符号间干扰

（续）快衰落（Fast Fading）

- 由于衍射和反射造成来自不同路径信号“自干扰”的信号衰落效应，其可有如下冲击响应描述：

$$h(t) = \sum_{k=1}^{n} a_k \delta(t - \tau_k) \exp(j\theta_k)$$

其中，a_k是路径k上的信号幅度，τ_k是其上的信号到达时延，θ_k是其上的信号相位。快衰落多发于人口密集的都市区。



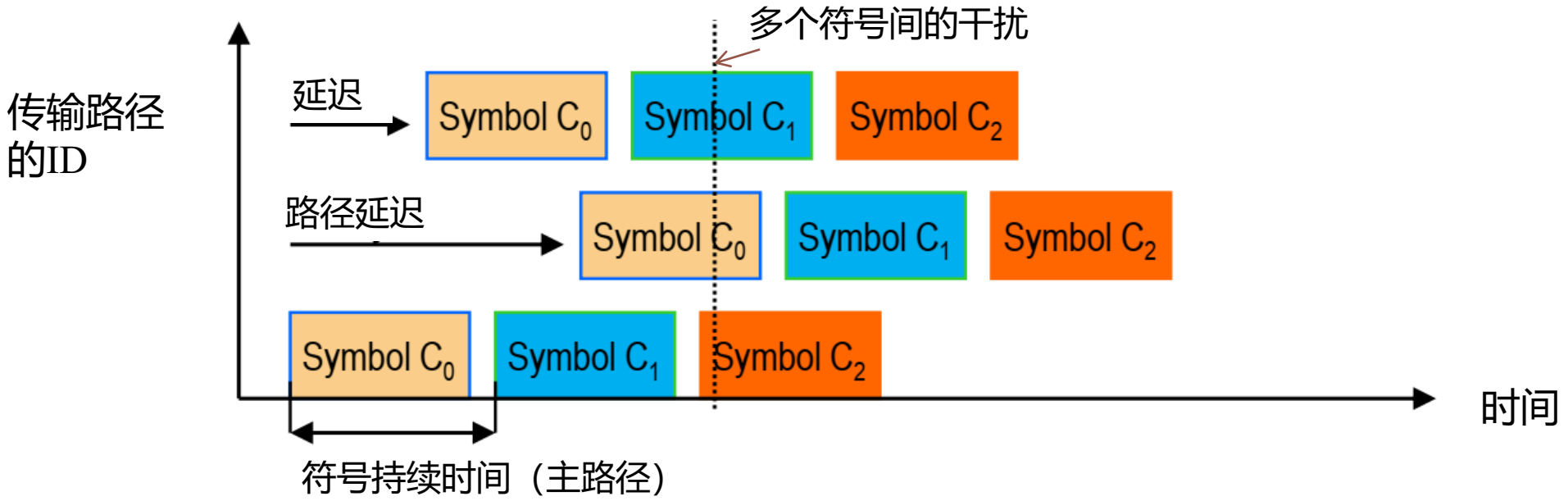
路径A：自由空间视距内传播；
路径B：反射（信号遇到比波长大得多的物体）；
路径C：衍射（障碍物尺寸与信号波长相当）；
路径D：散射（障碍物尺寸小于信号波长或表面不规则）；

路径E：大尺度阴影效应（Shadowing）中，由于动态障碍物遮挡造成的阴影区域动态变化（生灭过程，birth-death process）；
注意：此处多普勒扩展被忽略了。

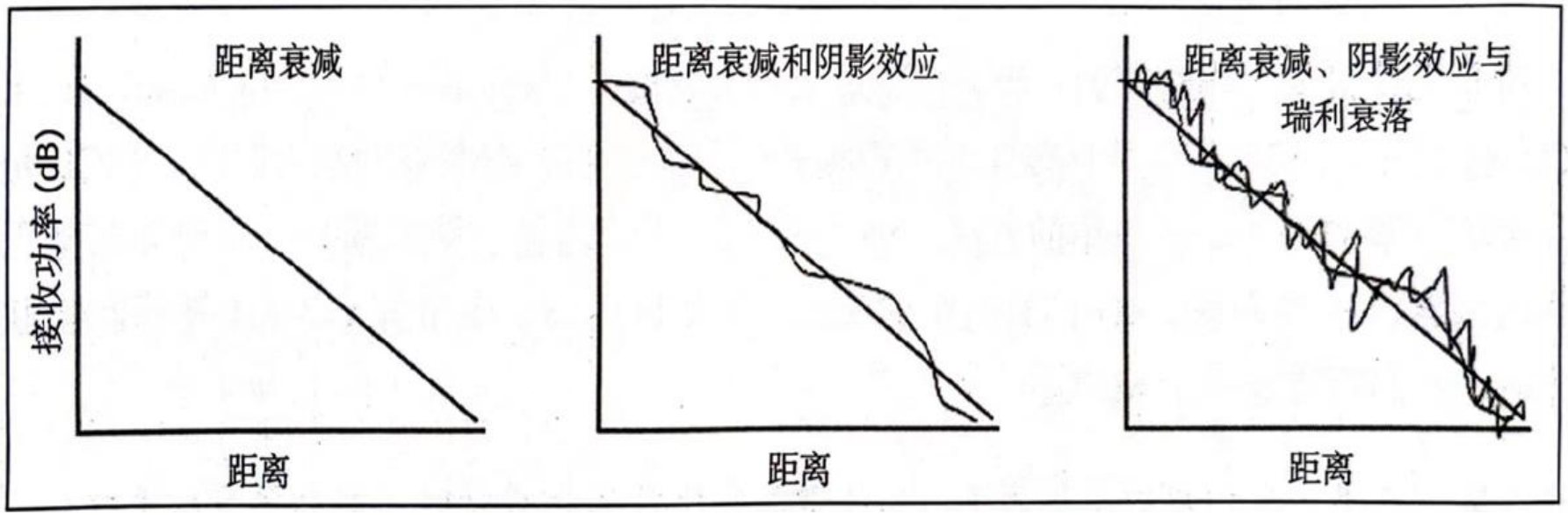
由快衰落（多径干扰）导致的符号间干扰（续）

符号间干扰（Inter-Symbol Interference, ISI）

- 符号：数字信号被**调制**后，对应一组数字信息（bits）的模拟信号特定波形。单个Symbol可能携带多个bit的信息。
- 在高速率单载波传输中，符号持续时间（周期，Duration）很短。
- 当多径干扰导致不同路径对LoS路径的时延扩展超过符号持续时间时，接收端的符号会发生重叠，这种现象称为符号间干扰（ISI）。



快衰落和慢衰落的区别



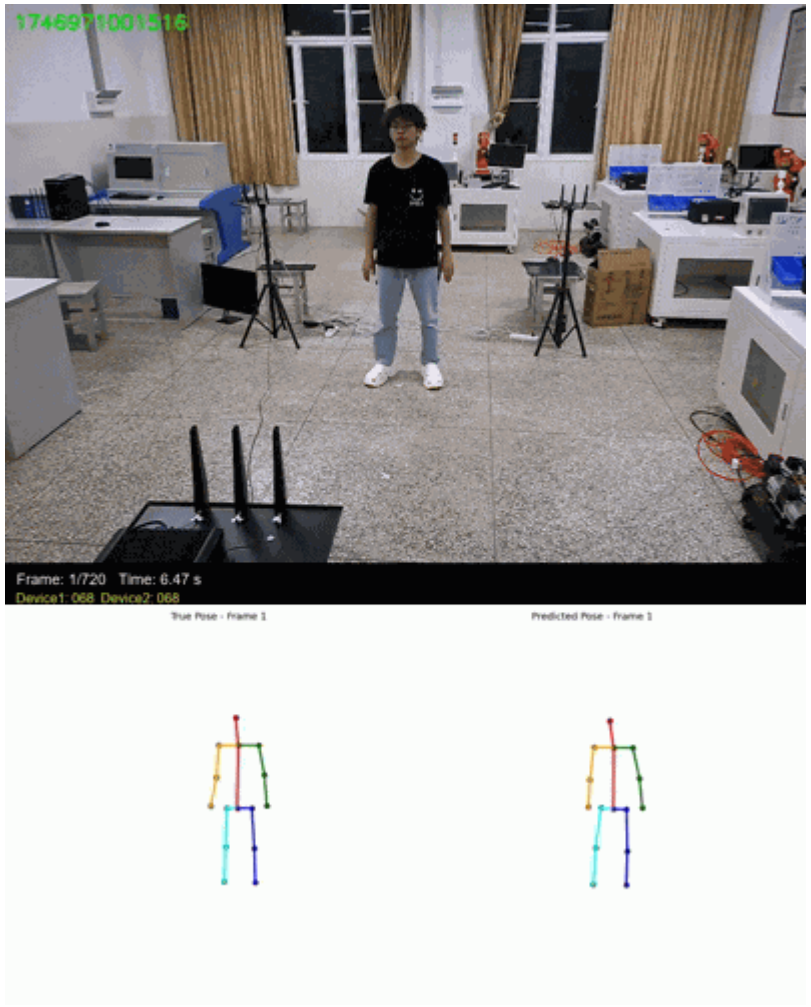
上图：射频信号的三种衰落效应。

- **左：**视线范围内的一般路径衰减，即接收功率随着距离的增加而线性减少。
- **中：**由大型结构、地形或运动引起的慢衰落（阴影效应，Shadowing），曲线波动表示大型结构、运动或地形对信号的阻挡导致信号强度变化。
- **右：**在距离衰减、慢衰落基础上叠加了快衰落（如瑞利衰落），曲线不仅有波动，还显得更加随机和剧烈，这反映了多路径传播引起的快速信号变化。

来源：[美]佩里 利著，中国移动研究院北京分院译，《物联网系统架构设计与边缘计算（原书第二版）》

多径效应原理在物联网传感上的应用

无线信号（如WiFi、毫米波）收发装置作为传感器检测人体：



https://github.com/ruvnet/RuView

2026年一季度热门开源项目： WiFi DensePose 是一个把普通 WiFi 信号变成人体姿态 + 生命体征 + 存在检测的开源项目。

无线信道比特率（信道容量）估计

香农（Shannon）定理：单天线信道支持的最大比特率（信道容量，即无差错数据传输速率，Error-Free Data Rate）由如下公式决定：

$$C = B\log_2\left(1 + \frac{P_{rx}}{N}\right)$$

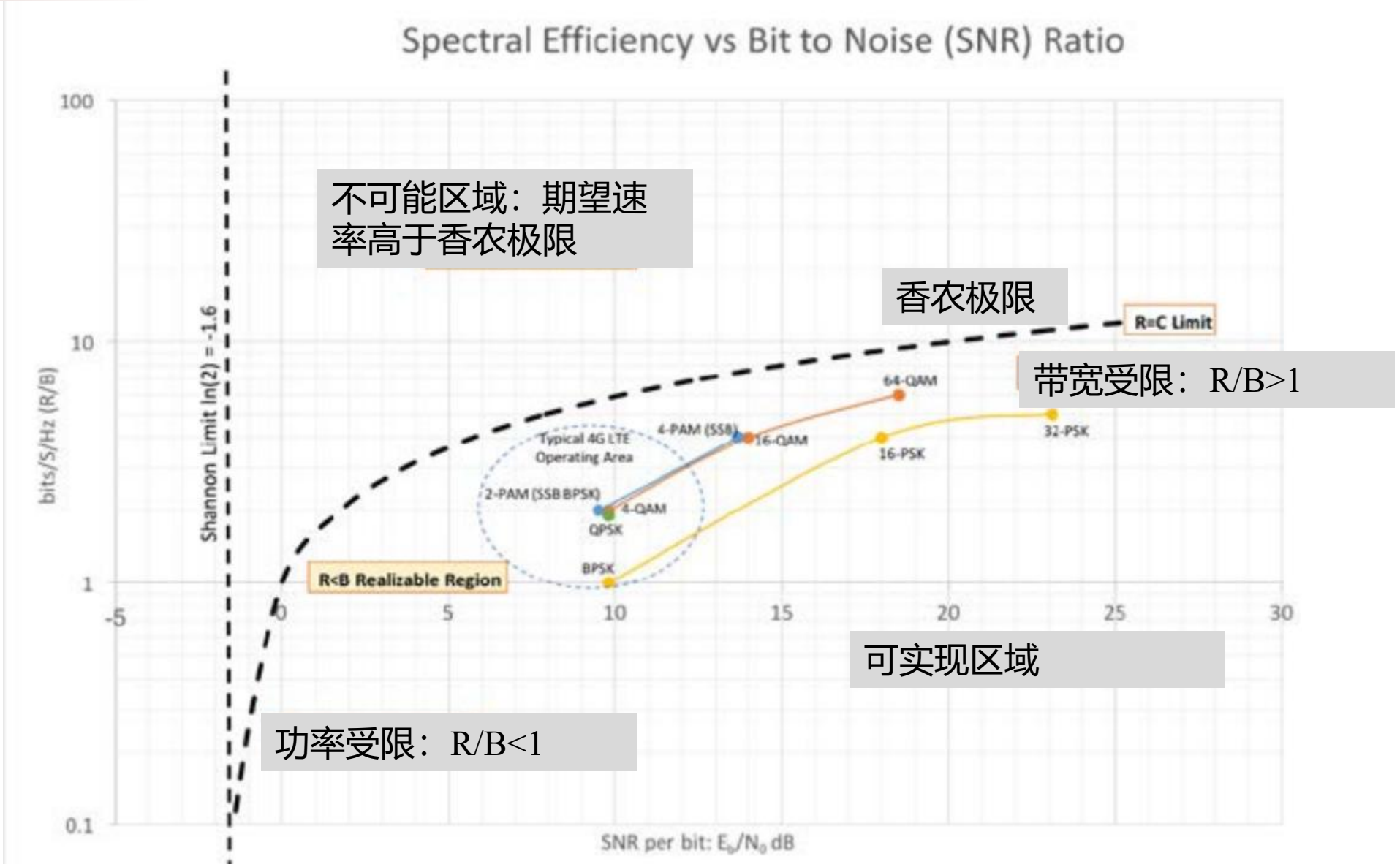
其中， B 表示信道的带宽 (Hz)， P_{rx}**是接收端的平均信号功率**， N是接收端平均噪声功率，加干扰时变为干扰加噪声项，Interference-plus-Noise，即$(I + N)$项。

(1) 信噪比（Signal-to-Noise Ratio，SNR）和信号-干扰加噪声比（Signal-to-Interference-plus-Noise Ratio，SINR）

- $\frac{P_{rx}}{N}$ 称为接收端信噪比，提高信噪比是提高信道容量的通用有效手段。
- 如有其他干扰源信号在接收端的功率为*I*，香农公式中的“信噪比”拓展为信号-干扰加噪声比：$\frac{P_{rx}}{I+N}$
- 例：期望在5000Hz带宽上达到200b/s的传输速率，那么所需的最小SNR为：

$$\text{SNR} = 0.028 = -15.28\text{dB}$$

频谱效率（bit/s/Hz）与信噪比



调制解调与信道编码

调制技术：将低频信源产生的基带信号转换/嵌入到高频载波信号的技术。基带信号称为调制信号，转换后的高频信号称为已调信号。

(1) 模拟调制技术

- 利用输入的模拟信号直接调制载波的振幅（振幅调制，AM），频率（频率调制，FM）和相位（相位调制，PM）。

(2) 数字调制技术

- 常见数字调制技术有振幅键控（ASK，Amplitude Shift Keying），移频键控（Frequency SK）、移相键控（Phase SK），和正交幅度调制（QAM，Quadrature Amplitude Modulation）等。

(3) 通过调制，发送一个信号波形（一个Symbol）可对应传输多个bit，如QAM4可表示2个比特的信息（00,01,10或11）。但随着比特数增加，对应的误码率也将增加。

调制符号（Symbol）与信息比特的关系

1个符号（Symbol）携带的比特（Bit）数由调制阶数M决定

$$\text{每符号比特数} = \log_2 M$$

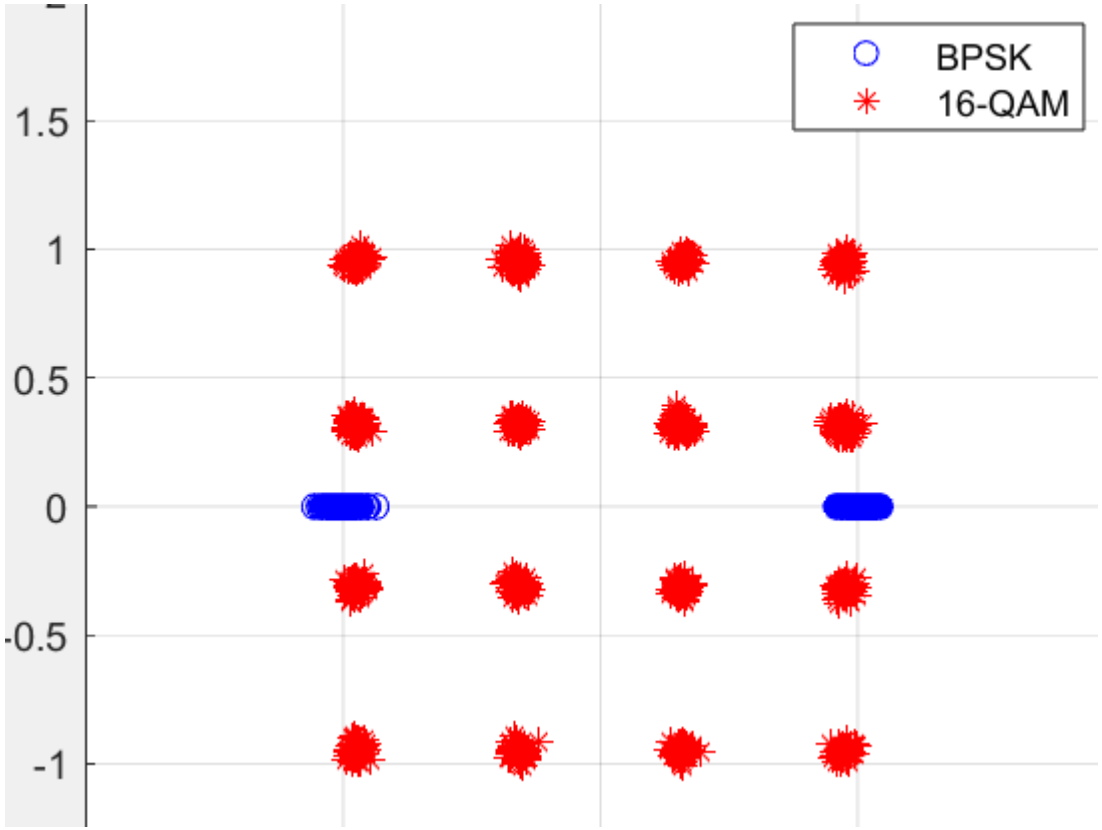
- 符号速率（Symbol Rate）称为波特率（Baud Rate），表示每秒传输的符号数。
- 符号周期（Symbol Duration）：符号速率的倒数：$T_s = 1/\text{波特率}$。
  - 例：1 MBaud符号速率对应符号周期为1微秒。

调制方式	调制阶数M	每符号比特数	典型应用场景
BPSK	2	1	高可靠性低速率（如卫星通信）
QPSK	4	2	4G/5G控制信道
16-QAM	16	4	Wi-Fi、5G数据信道
64-QAM	64	6	高速率场景（如光纤通信）

调制符号（Symbol）与信息比特的关系

不同调制方法星座图（Constellation）示意及图例说明

- BPSK vs 16-QAM 星座图；
- 信道噪声为高斯白噪声，信噪比为30dB；
- x轴：In-phase分量（I分量），信号在实数轴上的投影，与参考载波相位对齐；
- y轴：Quadrature（Q分量），信号在正交轴（虚轴）上的投影分量，与参考载波相位相差90°。
- 由I分量和Q分量共同决定$\log_2 M$个数字信号组的位置（自左上至右下）
  - BPSK为{0,1},
  - 16QAM为{0010,0110,1110,1010；0011；0111；1111；1011；0001,0101,1101，1001,0000,0100,1100,1000}),



调制解调与信道编码（续）

信道编码：通过在待传信息输序列里对原始信息编码，生成冗余（监督）码字，然后通过信道传输到接收端。在接收端，利用信息码元和监督码元的相关（监督）规律进行解码，通过比较冗余信息发现或纠正差错，用以提高信息传输的可靠性。

(1) 常用信道编码技术

- 分组码（如CRC校验码），卷积码，网格编码，Turbo码等。

(2) 误码率（Bit-Error Rate，BER）

- 误码率指接收端解码后收到的错误信息码字占总接传输息码元的比例。例如，原始序列为1010110100，接收序列为0010101010，则BER=50%。
- 给定前向编码方法的前提下，BER（在统计意义上）可用SNR的函数描述。

数据链路层协议

数据链路层功能

- 数据链路层主要负责多路数据流、数据结构探测、媒体访问和误差控制，从而确保通信网络中可靠的Point-to-Point与Point-to-Multipoint连接。
  - 注：数据链路层的误差控制是基于编码的FEC（Forward-Error-Correction，前向误差控制/前向纠错编码）技术的拓展，典型协议有自动重传请求（Automatic Repeat-reQuest，ARQ）和混合ARQ（Hybrid Automatic Repeat-reQuest，HARQ）技术等。
- 无线传感节点和边缘节点的物理约束（例如，机上能量和数据处理能力的约束）决定了完成这些功能的实现方式。

误差控制

- 一般地，基于ARQ的误差控制主要采用重新传送恢复丢失的数据包/帧。
- 其他传统无线网络的数据链路层广泛利用了基于ARQ的误差控制方案。但由于IoT网络中的传感节点往往面临能量与计算资源不足的问题，IoT网络应用中ARQ协议的部署会面临资源限制。
- 另外，FEC方案具有固有的解码复杂性，需要无线传感节点消耗大量处理资源。因此，基于低复杂度编码与解码方式的简单误差控制，是IoT网络中误差控制的期望解决方案。

数据链路层的媒体访问控制（Medium Access Control）

- 多跳自组织无线传感网络MAC层协议需要实现如下两个目标：
  - 针对于IoT网络服务区域内密集布置的节点，以及它们之间可能产生的多跳无线通信，需要建立数据通信链接以获得基本的网络基础设施。
  - 为使无线传感节点公平有效地共享有限的通信资源（如信道），需要对共享媒体的访问进行管理。
- IoT网络的MAC协议必须具有能量保护、移动性管理和失效恢复策略。
- 考虑现有的MAC解决方案，主要包含以下几种访问方式：
  - （1）基于TDMA或FDMA等的静态媒体访问（静态Chanalization）。
  - （3）基于CSMA的随机媒体访问 （Random Access）。
  - （2）基于混合TDMA/FDMA-CSMA的媒体访问。

媒体访问控制

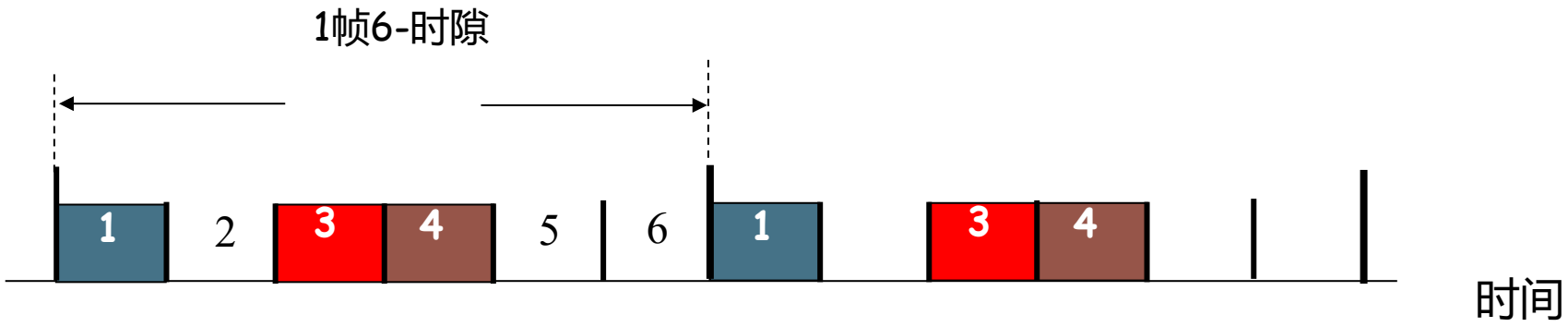
名词解释：

- **信道分割**（Channel Partition/Division）
  - 对信道使用权利的分割可以是物理意义上的，如基于频谱分割的信道复用（FDMA），也可以是逻辑意义上的，例如基于时间片的信道分割（TDMA）和基于正交编码的信道分割（CDMA）。
  - OFDMA（正交频分多址）在将整个频段划分为更小的子载波的基础上，进一步将子载波的利用时间（即一帧，Frame）划分为多个子帧（Sub-frame），每个子帧划分为多个时隙（Slot），在频率-时间两个维度上完成了对信道的划分。
- **随机访问**（Random Access）：
  - 对不可分割的信道，允许多个节点/连接尝试竞争性同时访问，同时访问的结果导致信道冲突（Collision）和传输失败。
  - 随机访问机制要求传输能够从冲突中复原，并在多节点冲突的场景下随机选择优胜者。

媒体访问控制方法简析：TDMA

TDMA：时分多址

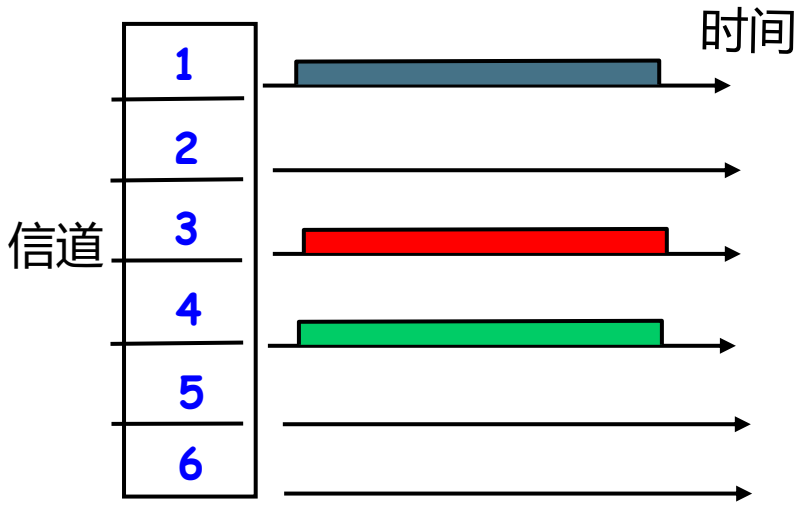
- 逻辑资源片静态分配（每个节点分配固定时间片）。
- 优点：保证分配公平，不存在时间片冲突。
- 缺点：频谱效率低，节点数量无法扩展。
- 示例：6节点，节点1,3,4当前有数据传输，节点2,5,6为空（idle）



媒体访问控制方法简析：FDMA

FDMA：频分多址

- 物理资源片静态分配（每个节点分配固定信道）。
- 优点：保证分配公平，不存在信道冲突。
- 缺点：频谱效率低，节点数量无法扩展。
- 示例：6节点，节点1,3,4当前有数据传输，节点2,5,6为空闲（idle）



媒体访问控制方法：CDMA

CDMA：码分多址

- 逻辑资源片静态分配（每个节点占用完全信道，但分配固定正交编码）。
- Bit时隙：一个数据比特占用一个时隙（slot）。
- M个微时隙（micro-slot）：一个数据bit分割为M个微时隙，对应长度M的编码序列$(c_1,\ldots,c_M)$。
- 编码：在发送端，对一个bit数据序列$\{d_1,\ldots,d_i,\ldots\}$中的第i个数据比特$d_i$,编码后，输出固定长度为M的序列:

$$Z_{i,m} = d_i c_m$$

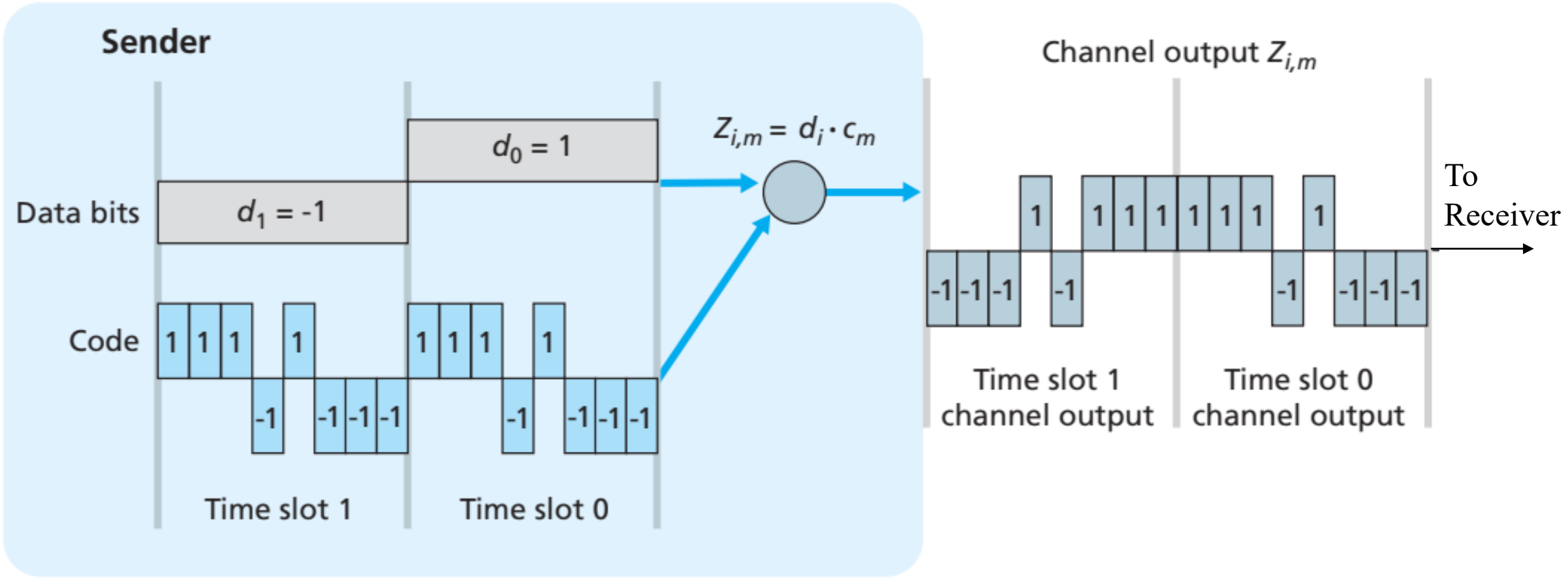
其中，$Z_{i,m}$是编码后对第i个比特输出的第m个编码。

- 解码：在接收端，将M个序列经下式子复原为数据比特:

$$\tilde{d}_i = \frac{1}{M}\sum_{m=1}^{M} Z_{i,m} c_m$$

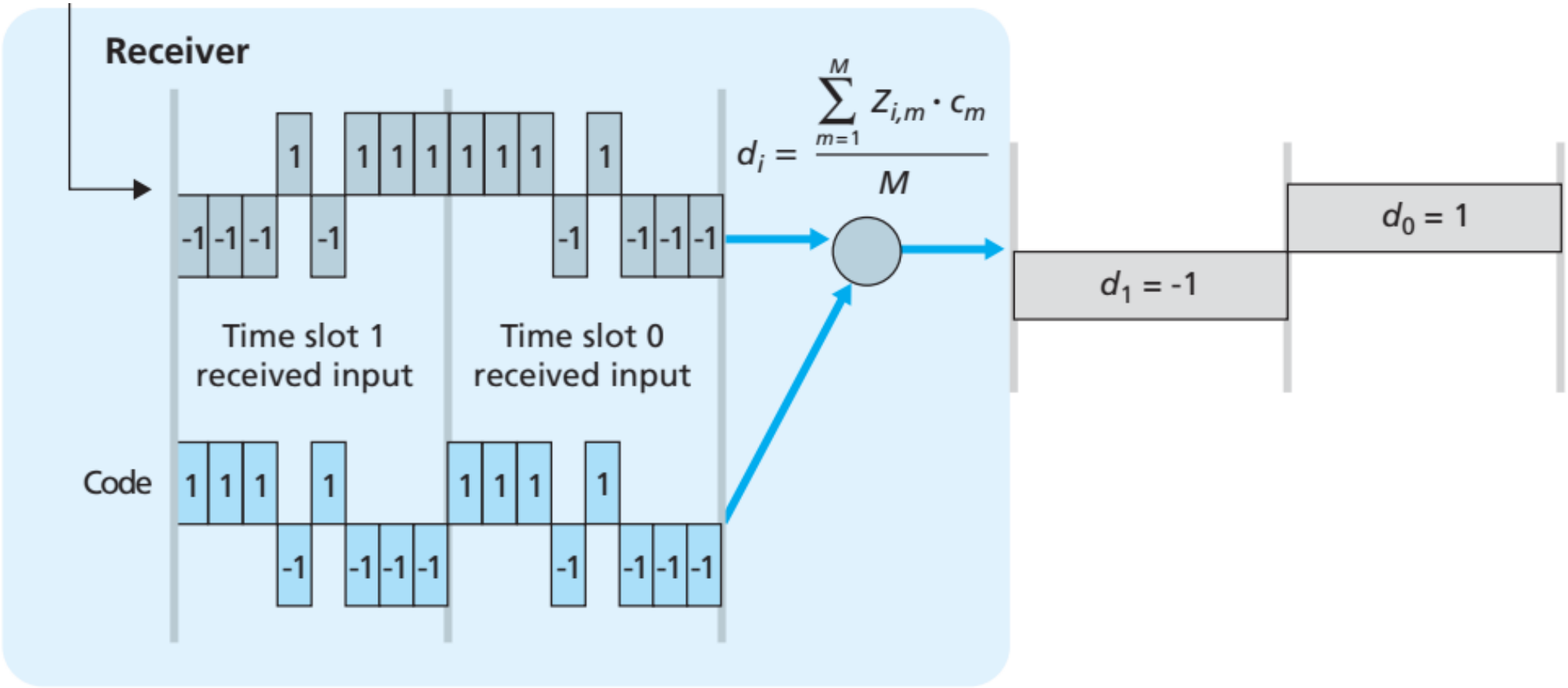
媒体访问控制方法：CDMA（续）

- 示例：CDMA码为（1,1,1，-1,1，-1，-1，-1）
- 发送端编码（用-1表示数据bit值0）：



媒体访问控制方法：CDMA（续）

- 示例：CDMA码为（1,1,1，-1,1，-1，-1，-1）
- 接收端解码（同样，用-1表示数据bit值0）：



媒体访问控制方法：CDMA（续）

当出现多个同时发送的编码（互相干扰时）：

- 每个信源使用正交编码（内积为0的编码，示例：双信源 ）
  - 信源1的CDMA码为（1, 1, 1，-1, 1，-1，-1，-1），
  - 信源2的CDMA码为（1, -1, 1，1, 1，-1，1，1）。
- 同单一信源的变化发生在每个信源对应的接收端
  - 第s个信源对bit-i编码后：

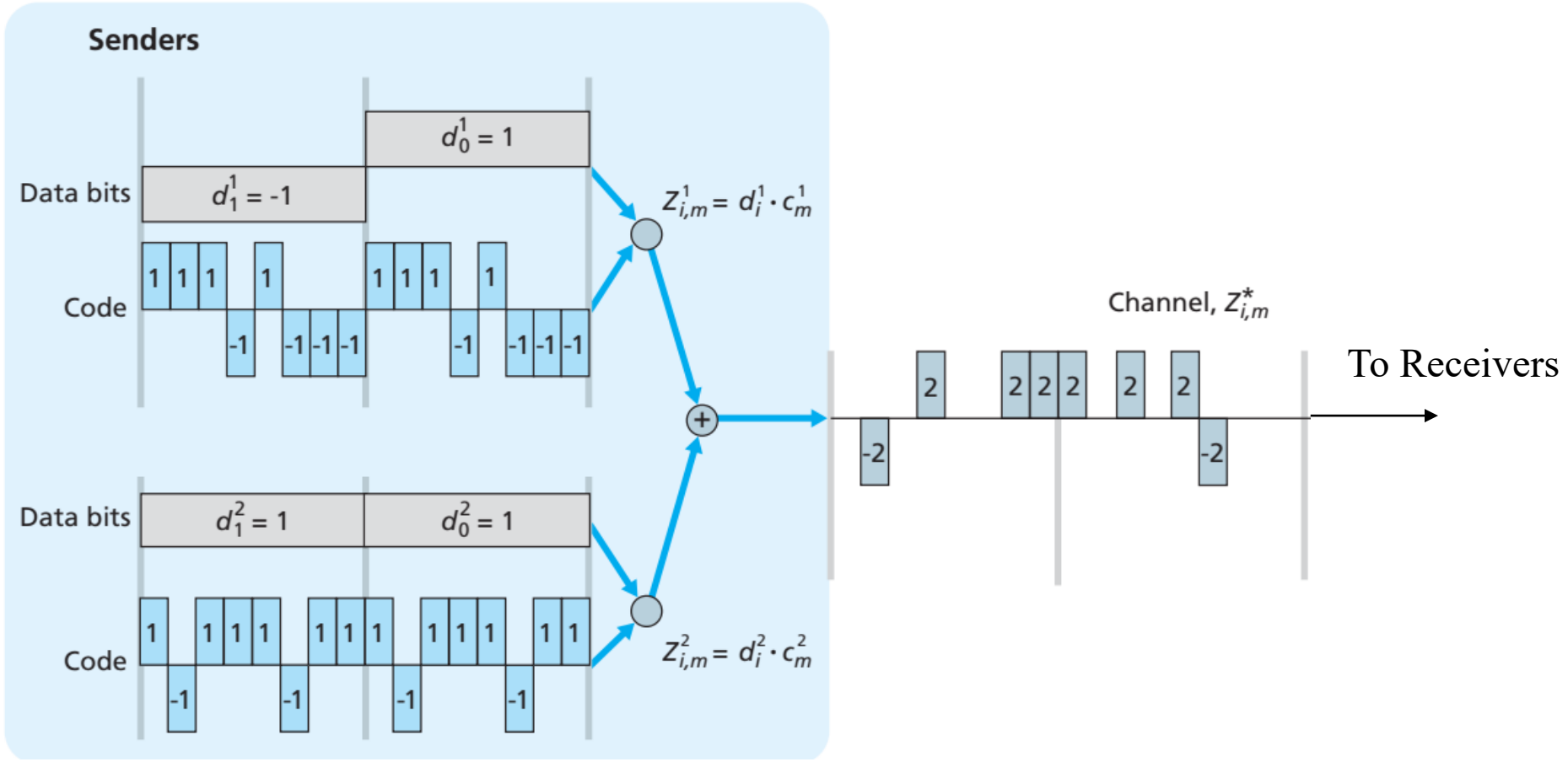
$$Z_{i,m}^{s} = d_i^s c_m^s$$

  - 在它的接收端：收到所有发送方传输bit 的总和

$$Z_{i,m}^{*} = \sum_{s=1}^{S} Z_{i,m}^{s}$$

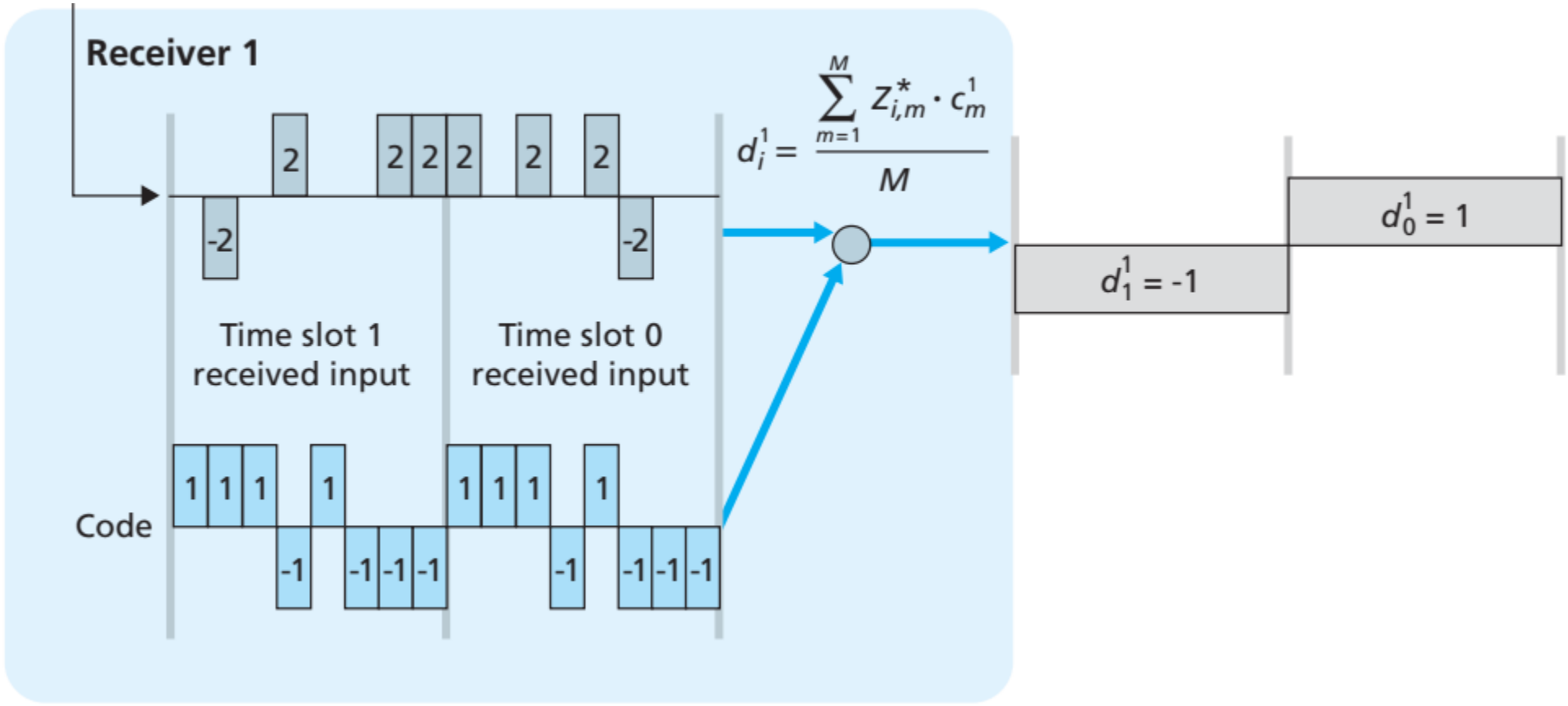
媒体访问控制方法：CDMA（续）

- （示例）接收端：双信源的CDMA码分别为（1,1,1，-1,1，-1，-1，-1），（1, -1, 1，1, 1，-1，1，1）



媒体访问控制方法：CDMA（续）

- （示例）接收方1的解码结果：双信源的CDMA码分别为（1,1,1，-1,1，-1，-1，-1），（1, -1, 1，1, 1，-1，1，1）



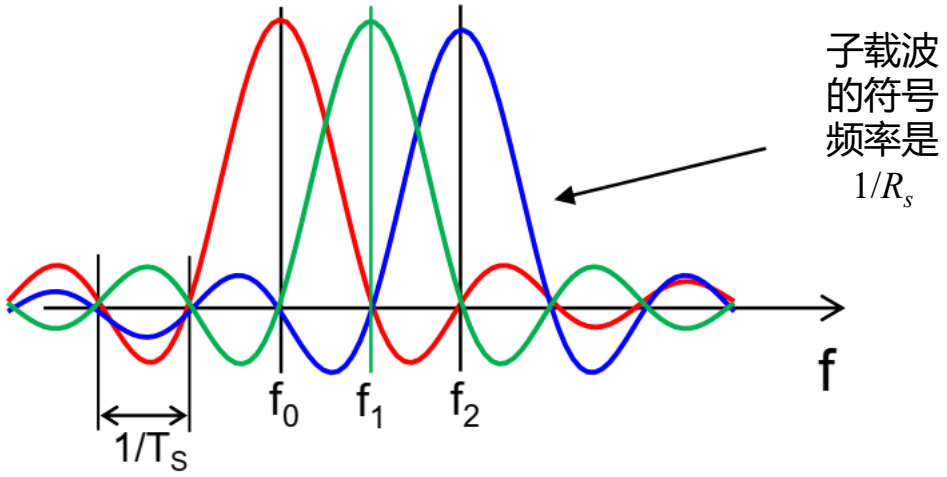
媒体访问控制方法：同时分割时间片和频率

OFDMA：Orthogonal Frequency Division Multiple Access，正交频分复用

- 将高速Symbol流分割为多个低速并行子载波，各子载波在频域上正交排列，避免干扰
  - 由于增加了每个子载波上Symbol的时长，有效抑制了**符号间干扰（ISI）**。
  - OFDMA的“正交”不同于FDAM的“正交”，**OFDMA的子载波频段互有重叠**，通过子载波在时域信号上一个Symbol周期内的能量积分检测实现正交（下式为一般正交检测公式）：

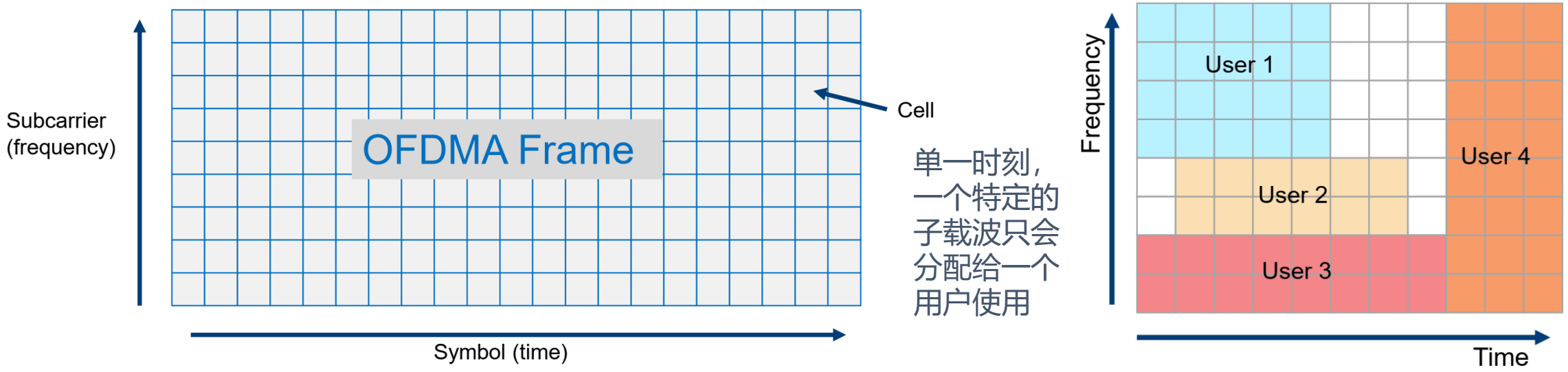
$$\int_0^T A(t)B(t)dt = 0$$

  - 特征：每一个子载波的峰值出现在其他子载波的能量零点上。



正交频分复用（OFDMA）

- OFDMA在逻辑意义上的“时频网格”资源块（Resource Block）划分



- 类比于CDMA，OFDMA也是通过构建（子载波）正交基实现多用户信号的正交（分离）。
- OFDMA的正交基不再是简单的编码向量，而是子载波信号的傅里叶反变换基：

$$\int_0^T e^{j2\pi f_k t} \cdot e^{-j2\pi f_m t} dt = 0 \quad (k \neq m)$$

- 正交性由IFFT（快速傅里叶反变换）的基函数提供（请参见本学期机器人专业课程：信号与系统。课后调研：子载波和正交基的关系是什么？）。

总结：静态媒体访问控制

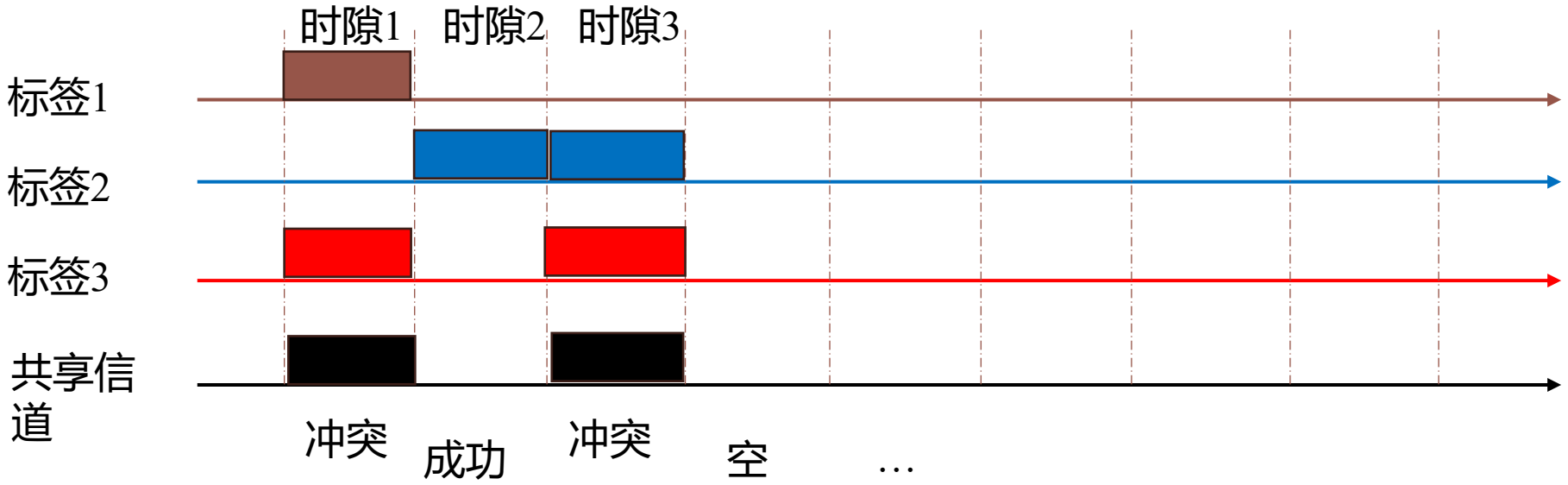
- TDMA和FDMA仍然适用于小用户规模、轻量级IoT组网和部署中的媒体访问控制（MAC）层协议。
- CDMA是2G和3G移动蜂窝网络时期的主流MAC层协议，例如：
  - 2G和3G通信：CDMA是3G移动通信系统（如IS-95、CDMA2000、WCDMA、我国的TD-SCDMA，时分同步码分多址协议）的核心技术，提供了较好的频谱效率和多用户接入能力。
  - 卫星通信：在卫星通信系统中，CDMA被广泛应用于多用户接入和抗干扰场景。
- OFDMA是当前4G、5G网络，以及中远距离IoT网络组网的基础MAC层协议之一
  - 4G和5G通信：OFDMA是4G LTE和5G NR等现代移动通信系统中的关键技术，支持高速数据传输和大规模用户接入。
  - 无线局域网：在无线局域网标准中，如Wi-Fi 6（IEEE 802.11ax），OFDMA被采用以提高频谱效率和系统性能（如抗用户间干扰）。

随机访问控制（Random Access Protocol）

- 节点有数据传输需求时：
  - 尝试完全占用待访问信道。
- 同一信道有多于一个节点访问时：
  - 信道冲突（Collision）。
- 随机访问控制协议：
  - 指定如何探测冲突：Collision Detection（CD）机制。
  - 指定如何避免冲突：Collision Avoidance（CA）机制。
  - 指定节点如何从冲突中恢复传输：随机延迟重传机制（Delayed Retransmission）。
- 相关协议：
  - RFID：ALOHA、Slotted ALOHA。
  - 802.11、802.15：CSMA、CSMA/CD、CSMA/CA。

随机访问控制：ALOHA协议

- ALOHA协议特点:
  - 尝试完全占用待访问信道。
  - 随机访问：终端本身无法预计其他终端的数据发送时刻。
  - 信道竞争：所有站点自由竞争信道使用权，只有一个站点能够胜出。
- 电子标签（RFID）常用协议：Frame Slotted ALOHA（帧时隙协议，FSA）
  - 大帧（Frame）：RFID读写器定义的一段时长，其中包含多个时隙（Slot）。
  - 时隙（Slot）：节点只在时隙开始时传输，节点是同步的。



随机访问控制：时隙ALOHA协议（续）

- 时隙ALOHA冲突检测和重传:
  - 冲突时，接收方（如RFID读写器）不向发送方（标签）确认控制帧（ACK），发送方在一定时间内收不到ACK，则确认冲突。
  - 发送方（标签）根据随机数发生器随机选择下一大帧中的某个时隙重传。
  - 信道竞争：所有站点自由竞争信道使用权，只有一个站点能够胜出。
- 时隙ALOHA的在N个节点下的传输效率：
  - 每个节点在一个时隙内以概率p传输一数据帧。
  - 一个给定节点在某时隙无冲突（传输成功）的概率是$p(1-p)^{N-1}$。
  - N个节点中有一节点传输成功的概率是$\binom{N}{1}p(1-p)^{N-1}$。
  - 最大化上述概率（最优传输概率）的p为：$p=1/N$。
  - 将$p=1/N$代入，当$N\to\infty$时，有

$$\lim_{N\to\infty} Np(1-p)^{N-1}=1/e=0.37。$$

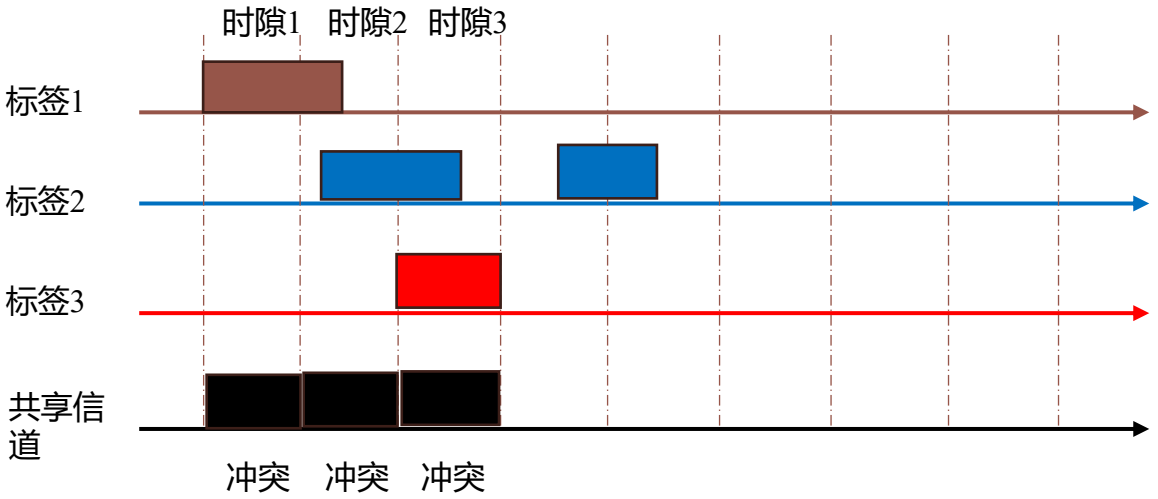
根据一阶最优条件：

$$\frac{df}{dp}=N(1-p)^{N-1}-N(N-1)p(1-p)^{N-2}$$
$$=N(1-p)^{N-2}\left(1-p-(N-1)p\right)=0$$
$$\Rightarrow p=1/N$$

$$\lim_{N\to\infty}\left(1-\frac{1}{N}\right)^{N-1}=\lim_{N\to\infty}\frac{\left(1-\frac{1}{N}\right)^{N}}{1-\frac{1}{N}}=\frac{e^{-1}}{1}=\frac{1}{e}$$

随机访问控制：纯ALOHA协议

- 纯ALOHA协议的特点:
  - 节点之间没有同步。
  - 节点在一个时隙中收到数据报后立刻发送。
  - 节点发现冲突后以概率p重传。



- 纯ALOHA的在N个节点下的传输效率：
  - N个节点必须保证在相邻两时隙中只有一节点传输成功（即每个其他节点在相邻两时隙中都不发送），其概率是$Np(1-p)^{2(N-1)}$。
  - 类似时隙ALOHA，对p求导有最优重传概率：

$$\frac{\mathrm{d}f}{\mathrm{d}p} = N(1-p)^{2(N-1)} - N2(N-1)p(1-p)^{2(N-1)-1} = N(1-p)^{2(N-1)-1}\left((1-p)-2(N-1)p\right) \Rightarrow p = \frac{1}{2N-1}$$

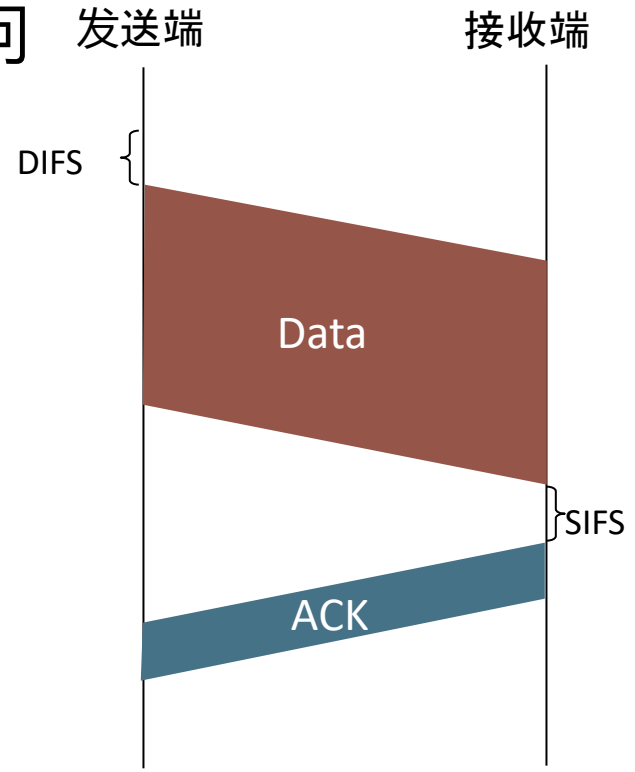
  - 最大传输效率：

$$P_{\max} = N \cdot \frac{1}{2N-1} \cdot \left(1-\frac{1}{2N-1}\right)^{2(N-1)}$$

思考：当N→∞时，其极限为多少？

随机访问控制：CSMA协议

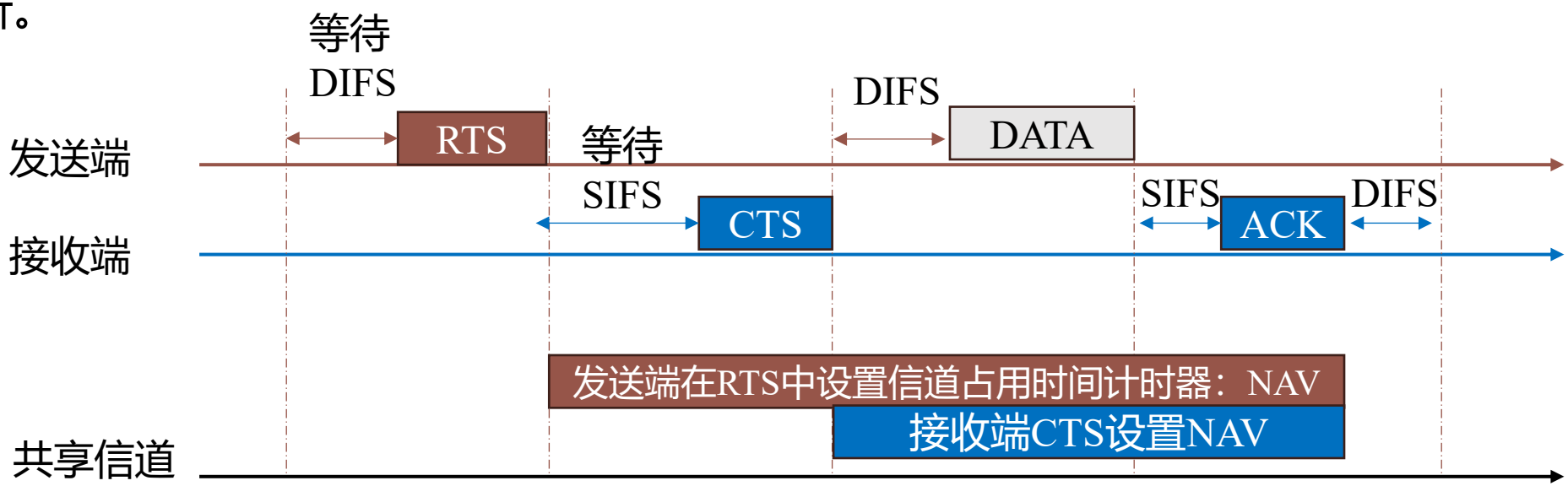
- CSMA: Carrier Sense Multiple Access，载波侦听多路访问
  - （1）Listen-before-transmit（传输前侦听）。
  - （2）若信道为空（idle），则传输数据。
  - （3）若信道忙（busy），延迟传输。
- CSMA/CD：带冲突检测的CSMA（用于以太网）
  - 用发送帧监听信道确认冲突。
  - 如有信道冲突，随机延迟后重发。



- CSMA/CA：带冲突避免的CSMA（用于802.11）
  - 发送端用特定小帧（Distributed Inter-Frame Space: DIFS）监测信道以确认冲突。
  - 接收端用特定小帧（Short Inter-Frame Space：SIFS）确认发送是否成功。
  - 若确认冲突或在传输帧未获得CTS，发送端经过一个随机backoff时间（回退时间）后重新监听。

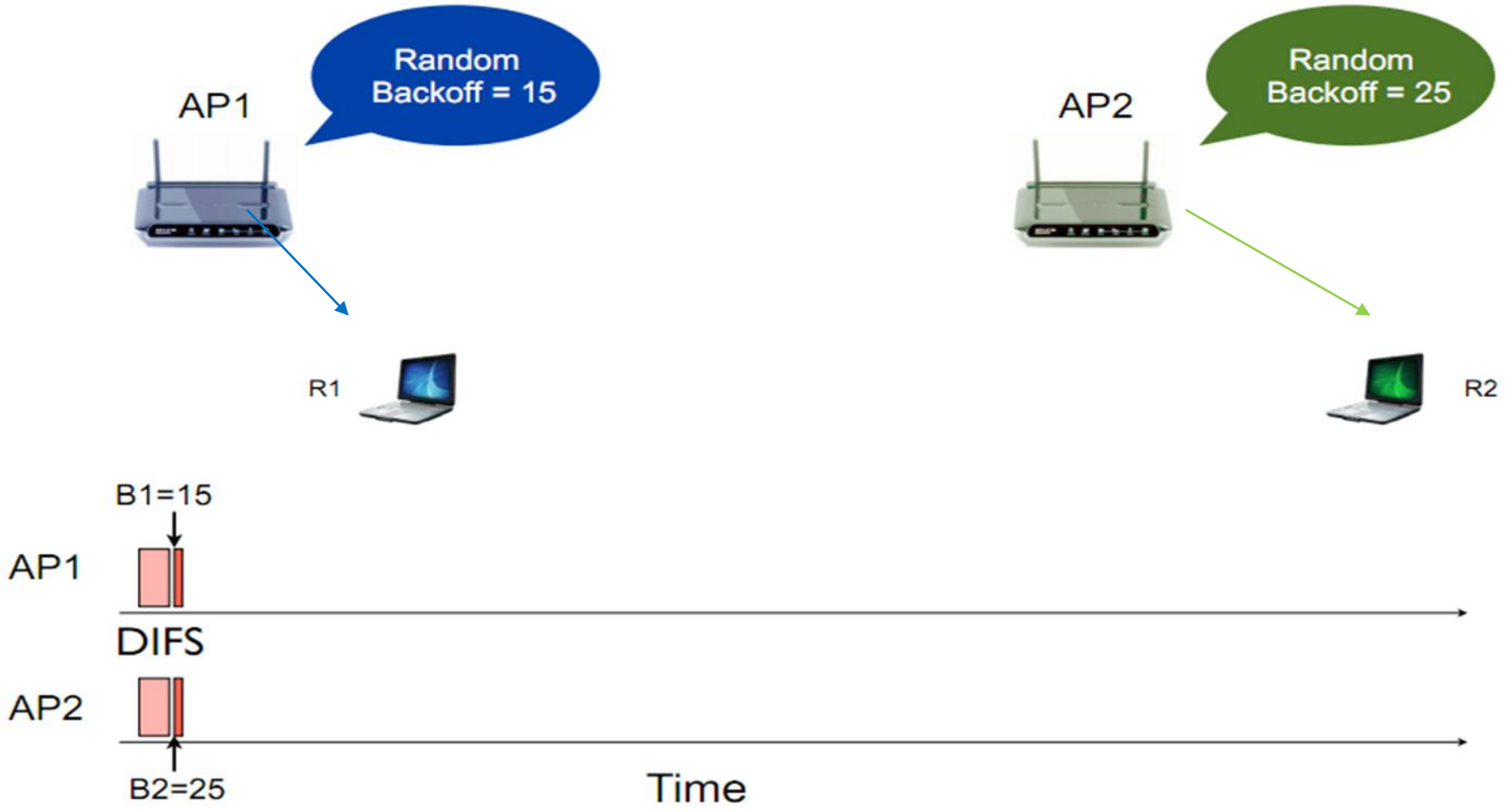
CSMA协议中的Ad-hoc（分布协调）模式

- Ad-hoc模式下的RTS-CTS握手
  - 发送端用特定小帧（Distributed Inter-Frame Space: DIFS）监测信道以确认冲突：向接收端发送RTS（Request-To-Send）控制帧。
  - 接收端用特定小帧（Short Inter-Frame Space：SIFS）确认发送是否成功：发送CTS（Clear-To-Send）响应帧。
  - 若确认冲突或在传输帧未获得CTS，发送端经过一个随机backoff时间（回退时间）后重新监听。



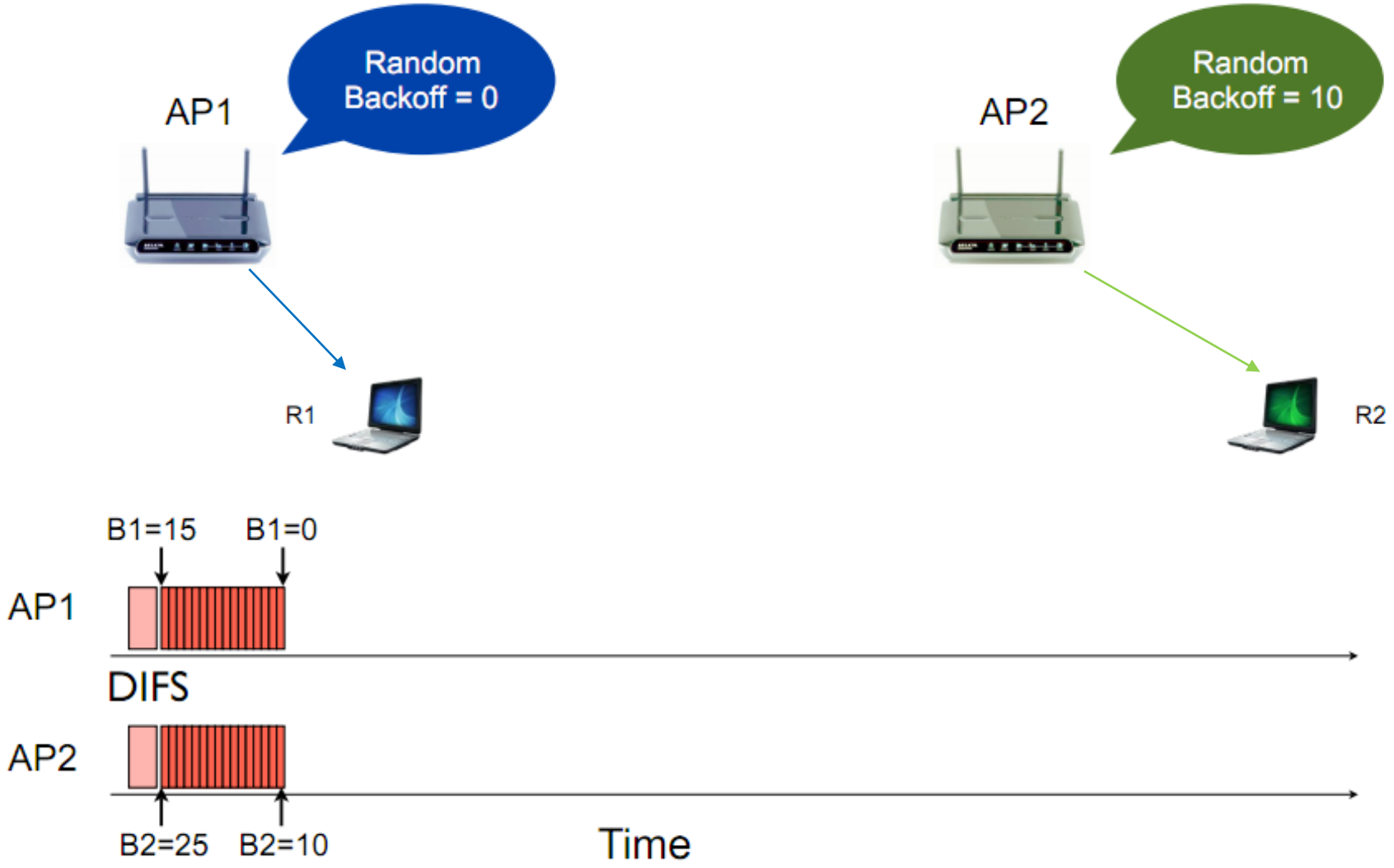
CSMA/CA 冲突后的Backoff举例

t=0时：两发送端分别确认backoff时间（时隙个数：AP1=15 vs. AP2=25）



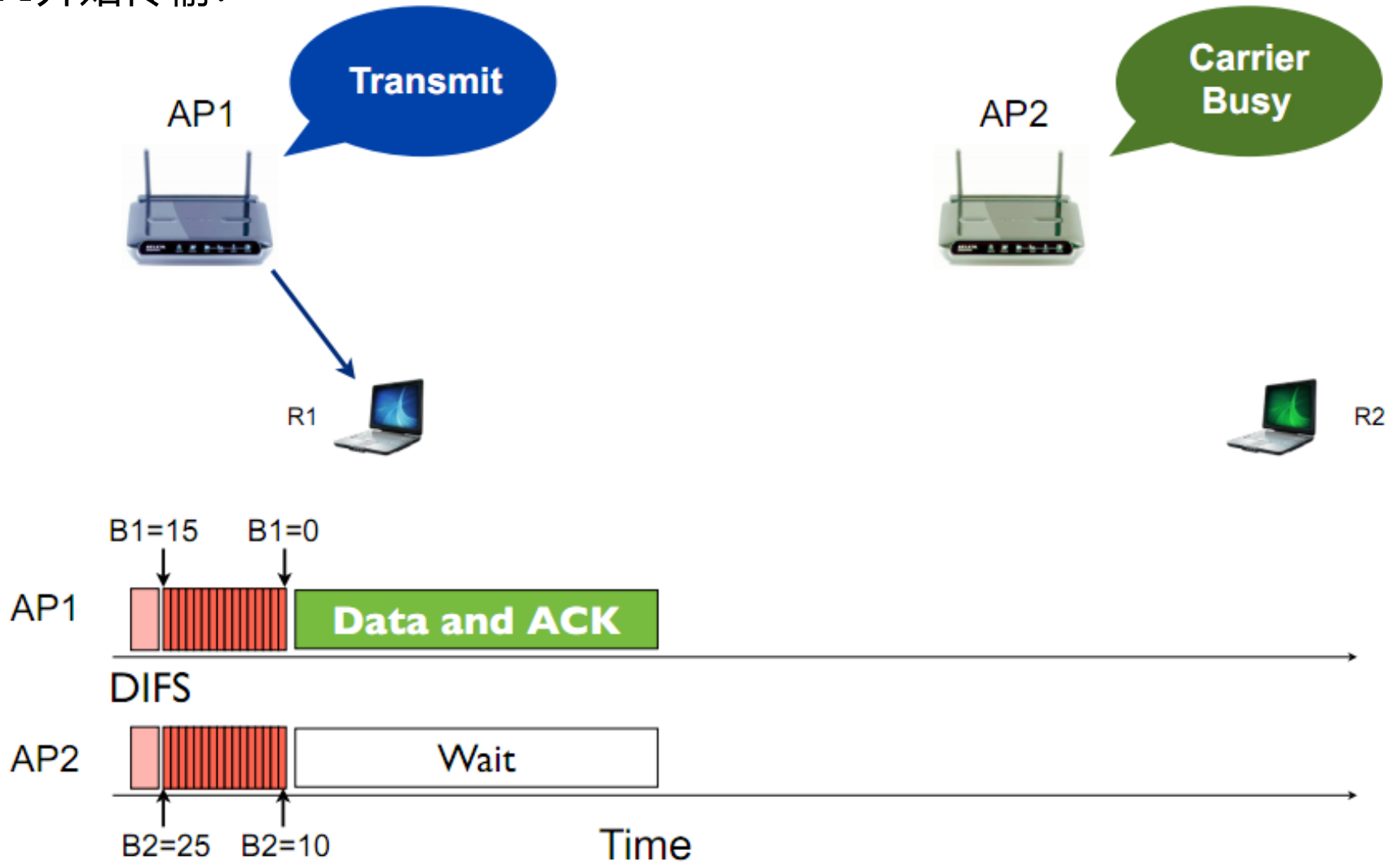
CSMA/CA Backoff举例：计时器状态（续）

t=15 time slots时：



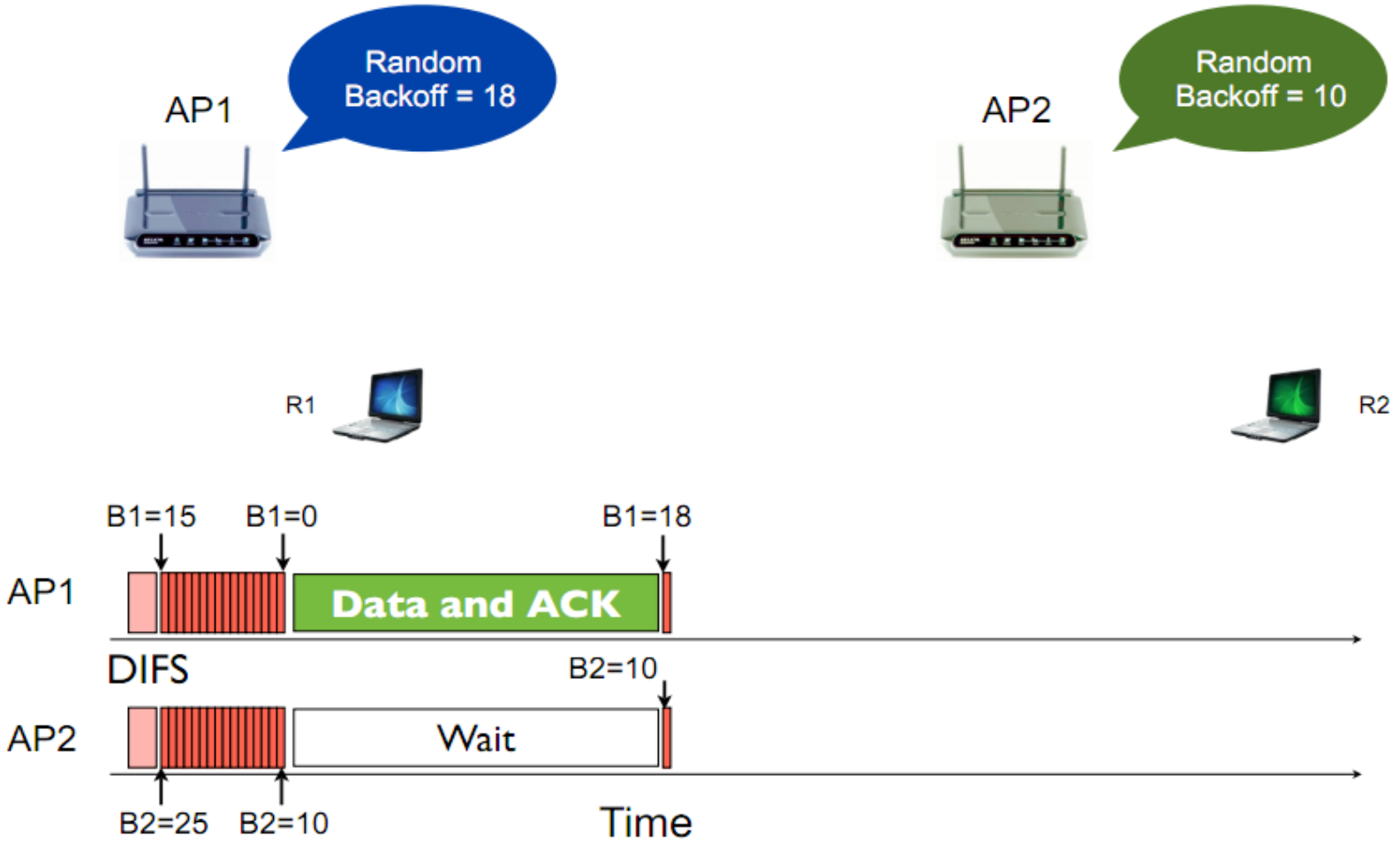
CSMA/CA Backoff举例：计时器状态（续）

发送端AP1开始传输：



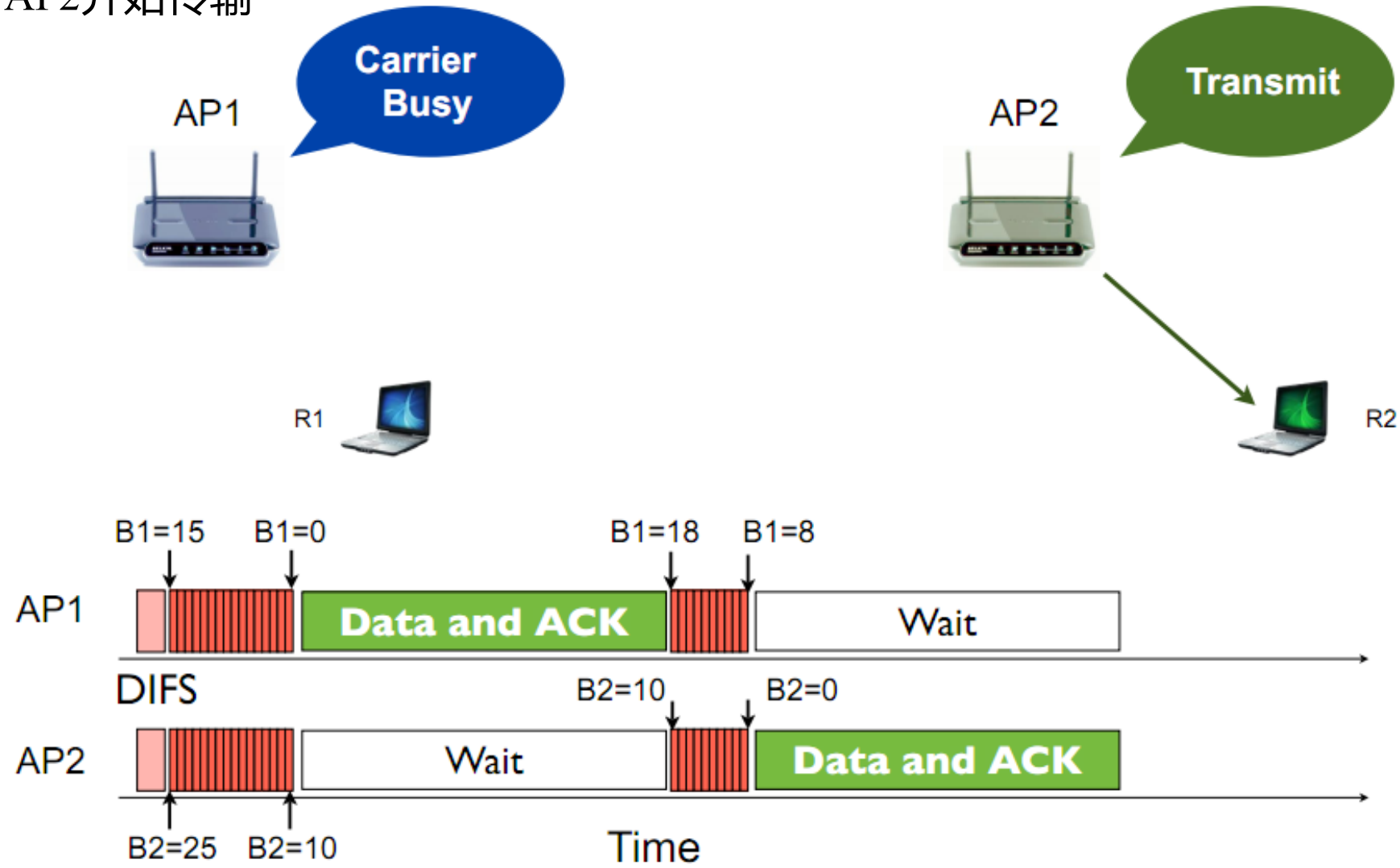
CSMA/CA Backoff举例：计时器状态（续）

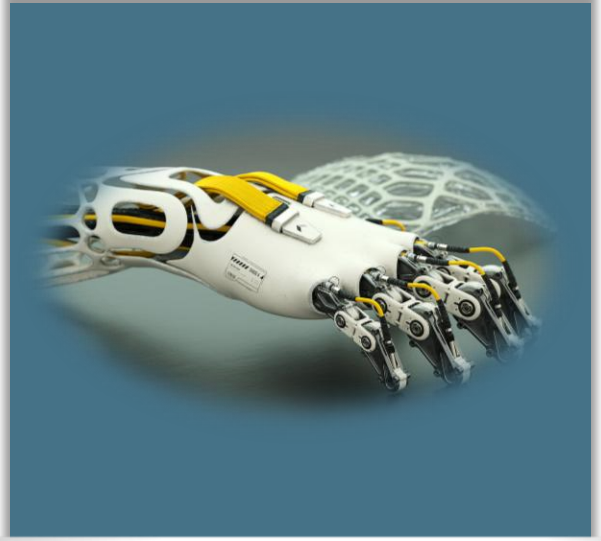
t=25 slots时：两发送端重新确认random backoff时间（时隙个数：AP1=18 vs. AP2=10）



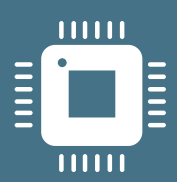
CSMA/CA Backoff举例（续）

t=30 slots： AP2开始传输





第二章
物联网接入与组网

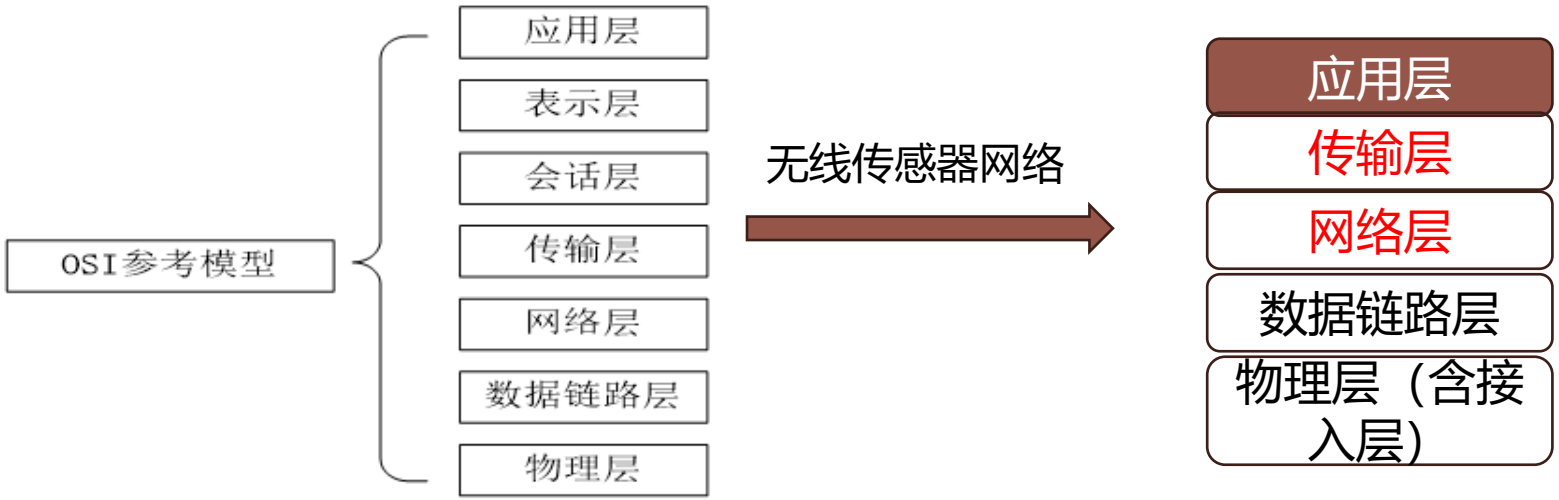



2.1 无线网络基础

2.2 TCP/IP协议简介

2.3-2.4 近距离和中远距离无线通信

回顾：网络协议封装参考模型



- 传输层协议：面向应用层提供数据报文通信服务
  - TCP（Transmission Control Protocol）：面向TCP连接的协议。
  - UDP（User Diagram Protocol）：无连接报文运输服务。
- 网络层协议（IP协议）：面向数据转发（Forwarding）和路由选择（Routing）的协议
  - 网际协议：IPv4/IPv6协议，寻址协议（地址转换协议，Address Resolution Protocol），路由协议，控制报文协议（ICMP，Internet Control Message Protocol）等。
  - IP协议封装上层数据报后进行传输。

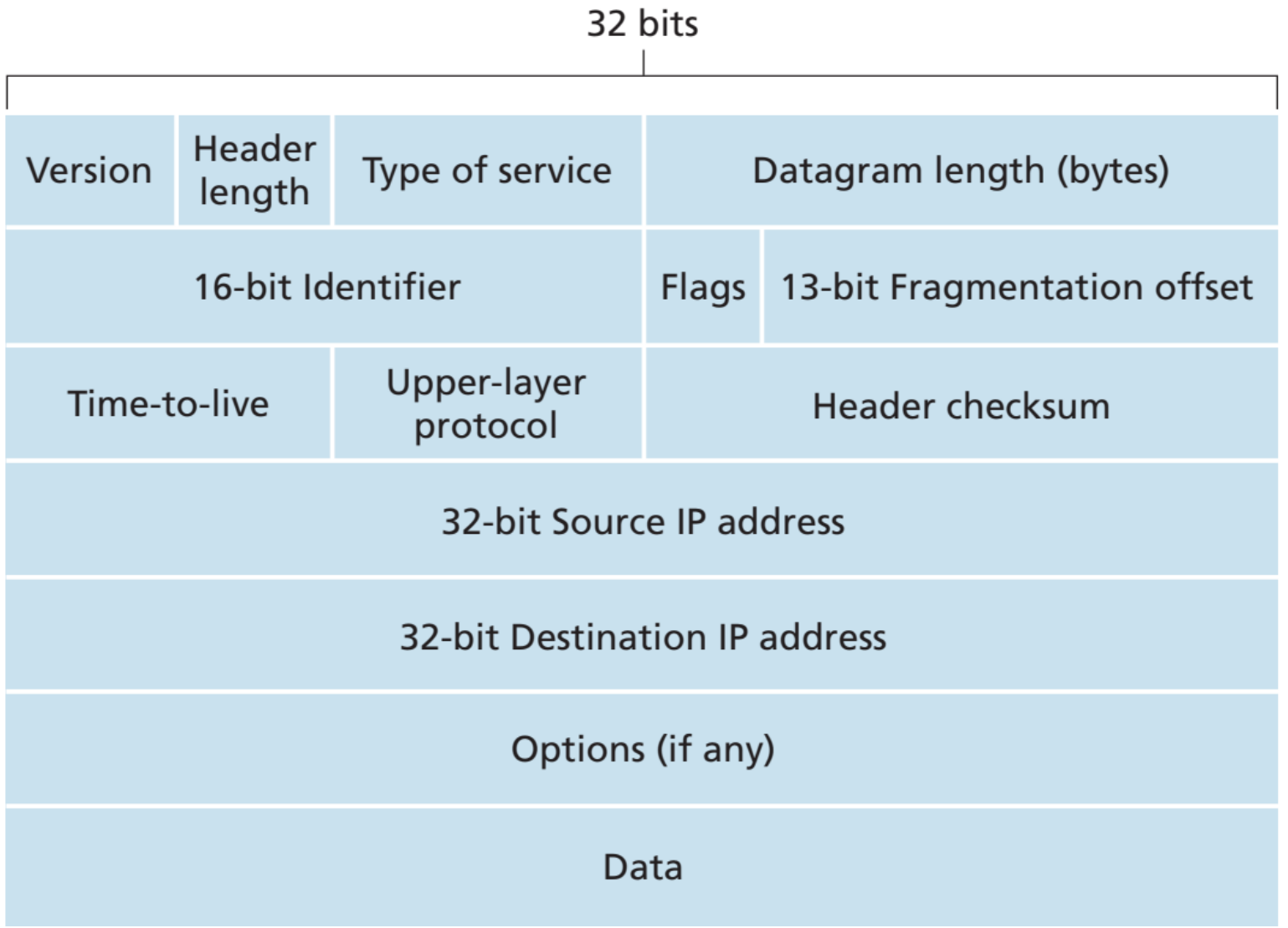
网络层协议概览

- IP协议
  - 用有限字长的地址（IPv4：32位，IPv6：128位）区分两台终端。
- ARP（地址转换）协议
  - 根据IP地址获取MAC地址（物理地址）。
  - MAC(物理)地址：一个48位的二进制数，由6个字节（每字节一个十六进制数）表示，形式为xx:xx:xx:yy:yy:yy。写在设备固件（firmware）中，其中前三个字节为IEEE分配给网络制造商的唯一标志符，后三个字节为制造商分配。
- 路由协议
  - 通过网关向不同子网中的两个地址（终端）以广播形式转发数据包。
- ICMP（控制报文）协议
  - 封装在IP数据包中，用于在IP节点、路由器中传递控制消息。
  - 控制消息：网络是否畅通，节点是否可达，路由是否可用（重定向），时间是否超时等消息。

IPv4报头格式（参见《计算机网络：自顶而下的方法》，机械工业出版社）

报头格式简介：

- Type of Service: 服务类型，定义数据报优先级。
- Flags/Offset：与大IP报文分片和组装相关，IPv6中已不支持。
- TTL：报文生存时间，防止循环转发。
- ULP：指定上层协议，指示报文应交给传输层的那个协议（TCP、UDP）。
- HC：首部校验和，用于报头校验，不包含数据部分。
- 源和目标地址：标识本报文的来-去地址。
- Options：可变长度字段，IPv6中已不支持。



参考：Linux中IP报头的定义（参见：include/uapi/linux/ip.h）

```
struct iphdr {
    #if defined(__LITTLE_ENDIAN_BITFIELD)
        __u8    ihl:4,
                version:4;
    #elif defined(__BIG_ENDIAN_BITFIELD)
        __u8    version:4,
                ihl:4;
    #else
    #error  "Please fix <asm/byteorder.h>"
    #endif
    __u8    tos;
    __be16  tot_len;
    __be16  id;
    __be16  frag_off;
    __u8    ttl;
    __u8    protocol;
    __sum16 check;
    __be32  saddr;
    __be32  daddr;
    /*The options start here. */
};
```

报头格式简介：

- ihl: IP头部的长度，以4字节为单位。
- version: IP协议版本，对于IPv4来说通常是4。
- tos: 类型服务（Type of Service）字段，用来表示服务质量。
- tot_len: IP数据报的总长度，包括头部和数据。
- id: 标识字段，用来唯一标识每一个IP数据包。
- frag_off: 分片和重组的偏移字段。
- ttl: 生存时间（Time-To-Live），每经过一个路由器就减1，直到为0时数据包被丢弃。
- protocol: 上层协议类型，例如TCP或UDP。
- check: IP头部的校验和。
- saddr 和 daddr: 源IP地址和目标IP地址。

IPv4中的两级分类编址（即子网划分）

- 早期的两级地址分割：一个32位IP地址分割为如下两项
  - 网络号：NetID，
  - 主机号：HostID。
- 早期的网络地址分类：根据NetID在32bit中的占用位数，分为
  - A类，由第一字节定义：0xxxxxxx.xxxxxxx. xxxxxxx. xxxxxxx。最多有2^7即128个A类网络地址组合。分配给大型网络。默认子网络掩码为255.0.0.0。
  - B类，由前第1、2字节定义： 10xxxxxx.xxxxxxx. xxxxxxx. xxxxxxx。分配给中型网络。默认子网络掩码为255.255.0.0。
  - C类，由前第1、2、3字节定义： 110xxxxx.xxxxxxx. xxxxxxx. xxxxxxx。分配给小型网络，每个网络只能容纳2^8-2台设备（主机）。默认子网络掩码为255.255.255.0。
  - D类，没有网络地址的概念： 1110xxxx.xxxxxxx. xxxxxxx. xxxxxxx。用于多目的地广播。
  - E类，没有网络地址的概念： 11110xxx.xxxxxxx. xxxxxxx. xxxxxxx。用于实验和开发保留用。

iPv4子网划分

- 现代化的三级地址分割方案：一个32位IP地址分割为如下三项

- 主要分割工具：子网掩码
  - 由32位1 和连续的 0 组成（一般写成四个用点分隔的十进制数）。
  - 1对应的部分：在IP地址中，这部分是网络位（包括网络号和子网号），标识了所属的网络区域。
  - 0对应的部分：在IP地址中，这部分是主机位，标识了该网络内的具体设备。
- 子网掩码示例
  - 11111111.11111111.11111111.00000000（二进制码，前24位是1）。
  - 转换为十进制码，可写成：255.255.255.0(/24)，其中，/24表示掩码中连续1的位数，即前24位为1。

iPv4子网划分：确定子网地址

- 子网掩码示例
  - 11111111.11111111.11111111.00000000（二进制码，前24位是1）。
  - 转换为十进制码，可写成：255.255.255.0(/24)，其中，/24表示掩码中连续1的位数，即前24位为1。
- 根据IP地址和子网掩码确定子网号：假设当前主机IP是 192.168.1.40。
  - 二进制化的IP地址是：11000000.10101000.00000001.00101000。
  - 将IP和掩码的进行“按位与”运算：

    11000000.10101000.00000001.00101000
    &&&&&&&&&&&&&&&&&&&&&&&&&&& = 11000000.10101000.00000001.00000000
    11111111. 11111111. 11111111. 00000000

  - 结果转换回十进制是 192.168.1.0，即当前主机所在的网络地址。
  - 判断两个IP是否属于同一个子网：用相同的掩码检验主机IP和目标机IP的网络地址是否一致。
  - 例如：192.168.1.10和192.168.1.150，根据子网掩码255.255.255.0(/24) 判断，同属192.168.1.0。

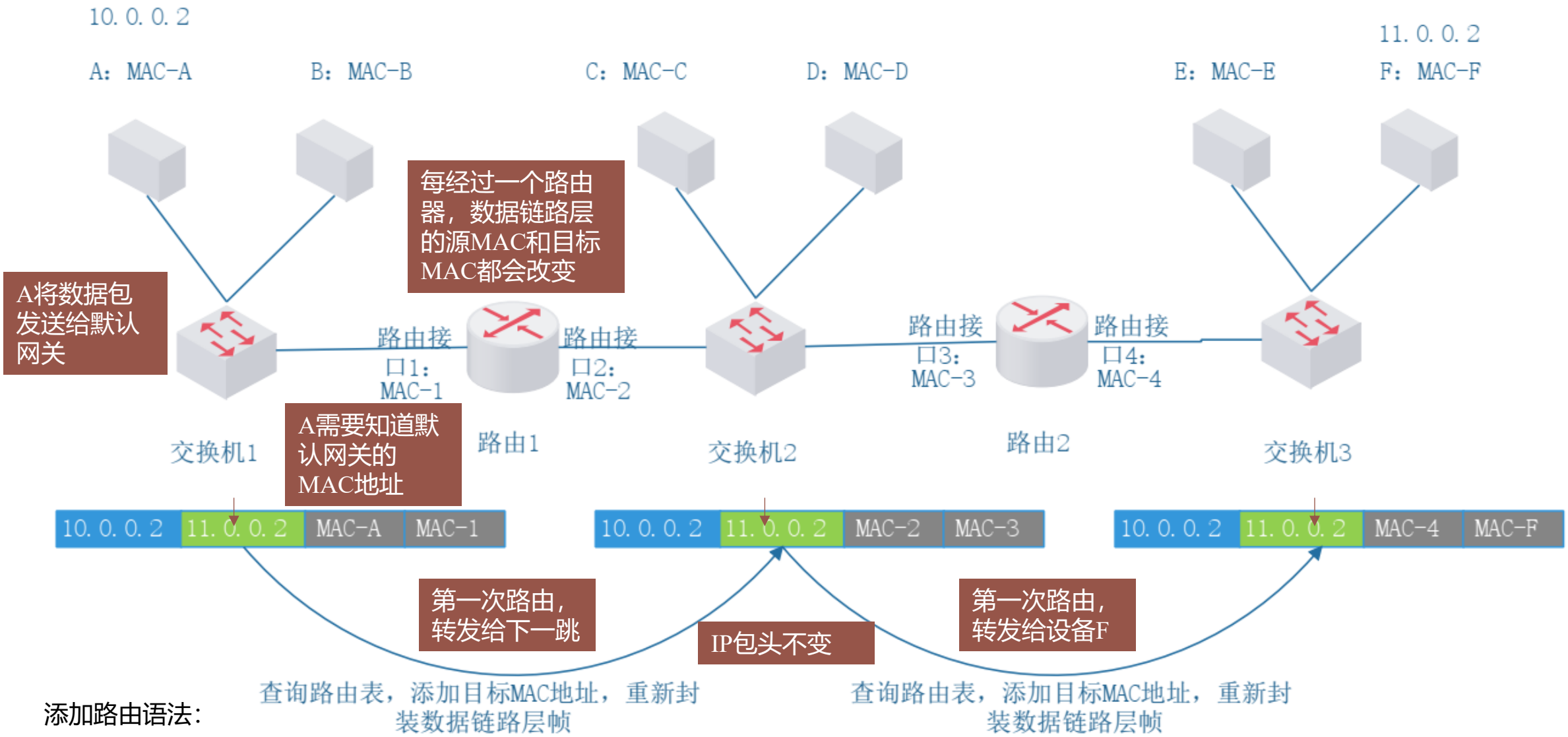
常见网络地址配置命令

- 配置“IP地址+子网掩码”：以windows下的命令行为例
  - netsh interface ipv4 show interfaces：查看当前网络接口列表。
  - netsh interface ipv4 set address name=“接口名称” static IP地址 子网掩码 默认网关：配置选定接口的静态IP地址和子网掩码。
  - 如：netsh interface ip set address "以太网" static 192.168.1.10 255.255.255.0 192.168.1.1
  - 上述命令行的解释：

参数	说明	设置内容
IP地址	子网内的唯一标识	192.168.1.10
子网掩码	确定网络边界	255.255.255.0
默认网关	出子网的下一跳地址	192.168.1.1

小练习：Linux环境下，节点的IP地址如何通过命令行配置？Windows环境下如何通过图形界面进行配置？

从一个子网IP到另一个子网IP的传输



添加路由语法：

传输层协议：TCP和UDP

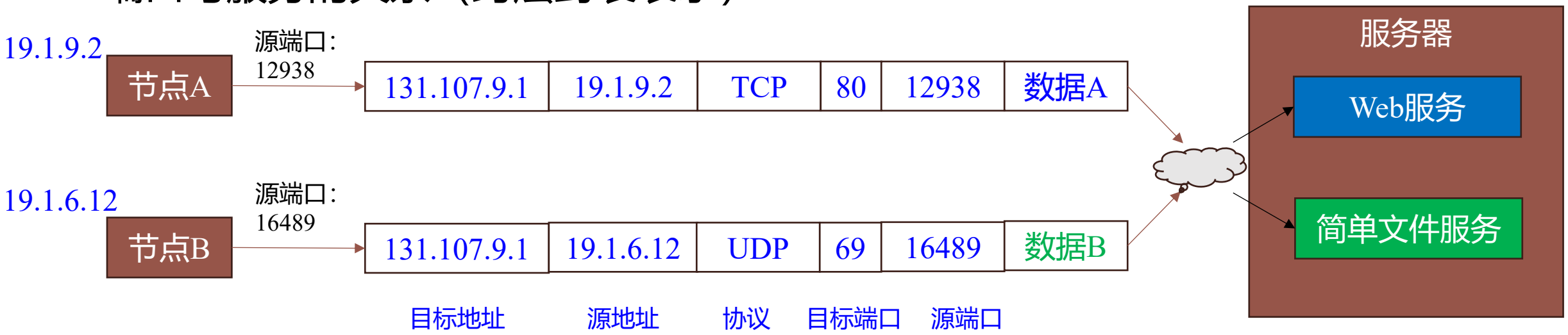
常见网络应用	应用层协议	下层（传输层）协议
Web	HTTP	TCP
安全终端访问	SSH	TCP
文件传输	FPT	TCP
流媒体服务	专用协议	UDP或TCP
VoIP	专用协议	UDP或TCP
远程文件服务	NFS	UDP

（续）传输层的端口号

- 接上页：应用层协议和传输层协议是多对一的关系。
- TCP/UDP协议附加一个端口号来标识应用层协议

应用协议：	HTTP	FTP	SMTP	POP3	TELNET	RDP	DNS	TFTP
端口号：	80	21	25	110	23	3389	53	69
	TCP						UDP	

- 端口与服务的关系（跨层封装表示）

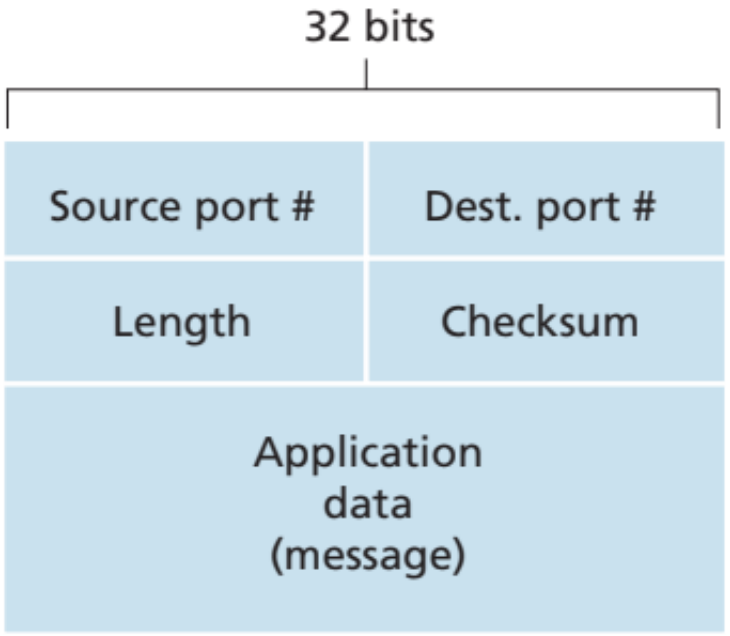


UDP（用户数据报协议，User Datagram Protocol）协议：直接基于IP层的无连接协议

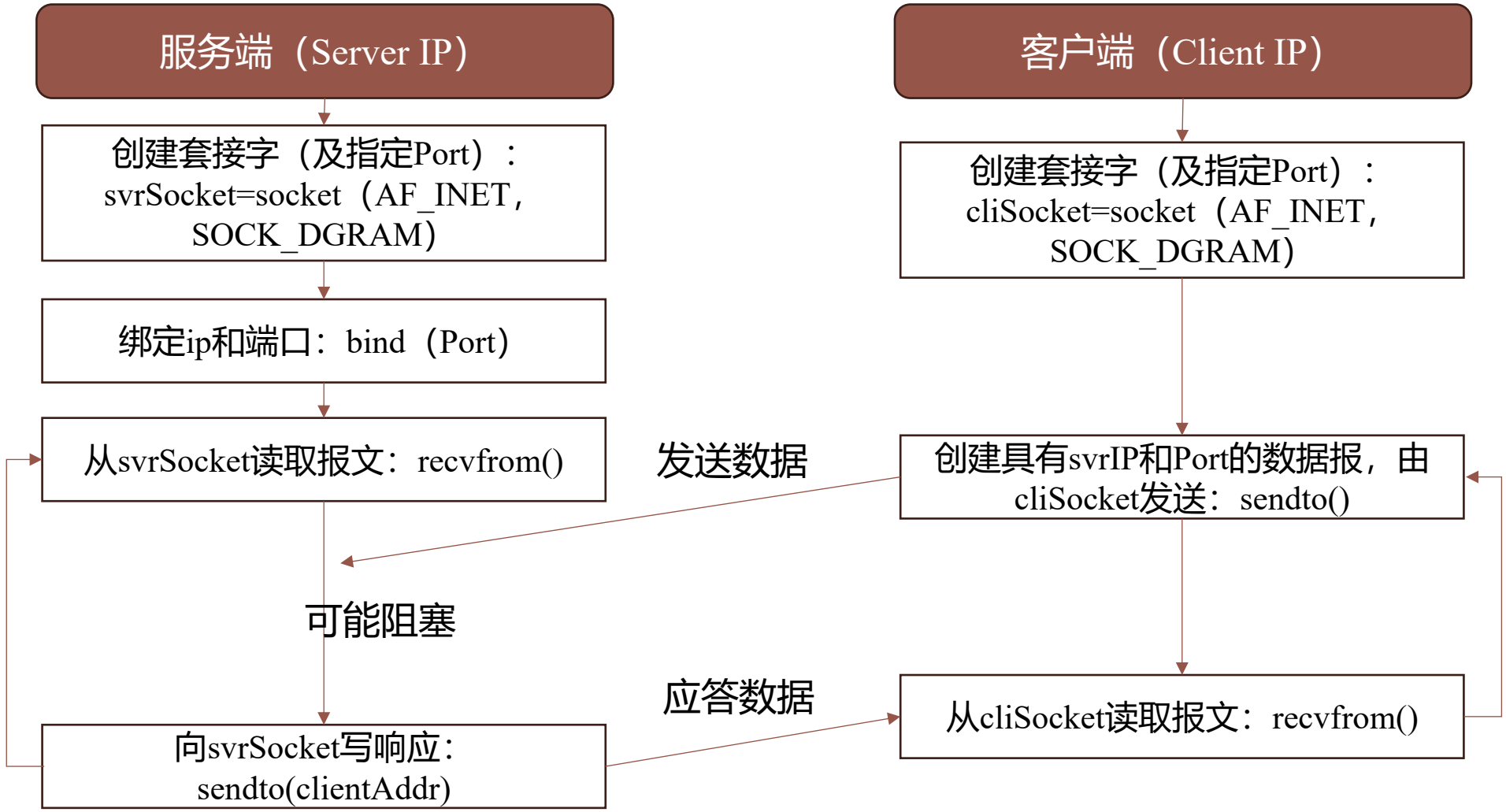
- UDP最小实现：
  - 1. 从应用进程得到数据；
  - 2. 对数据附加处理，确定多路复用的源和目的的端口号，形成报文；
  - 3. 报文交付网络层，用IP报文封装，根据目的IP交付接收端。
  - 特点：发送端和接收端在传输层没有握手，因此是无连接的。
- UDP报文结构

```c
#include <stdint.h>

typedef struct {
    uint16_t source_port;    // 源端口号
    uint16_t destination_port; // 目的端口号
    uint16_t length;         // UDP报文长度（包括头部和数据部分）
    uint16_t checksum;       // 校验和
    uint8_t data[];          // UDP数据部分，长度可变
} udp_header_t;
```



UDP套接字编程基础：C/S架构



注：AF_INET指示使用IPv4地址族（若需IPv6，则用AF_INET6）

UDP套接字编程基础

- 实现方式：C，C++，Python
  - 在Python中，使用socket库（“from socket import *”）；
  - 在C/C++中，使用sys/socket.h库头文件。

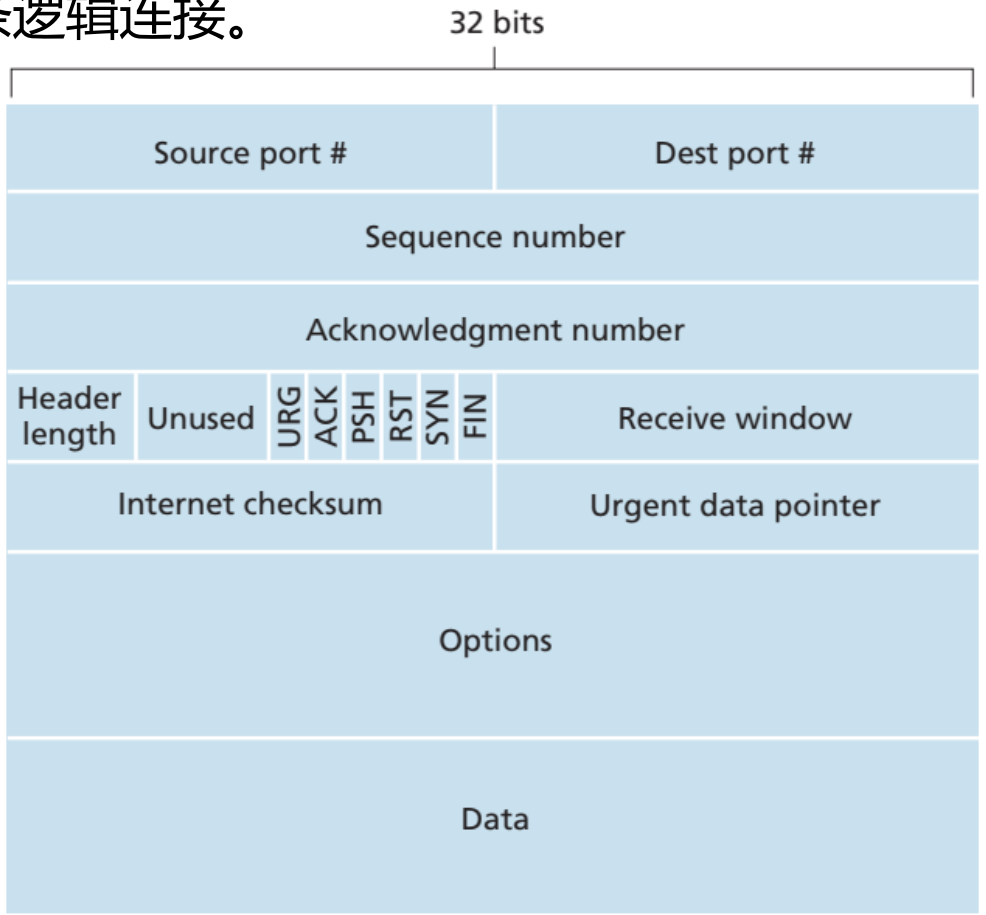
套接字类型：

- 数据报套接字（SOCK_DGRAM）：
  - 基于UDP，提供了一种无连接的、不可靠的数据传输服务。数据包以独立的形式被发送，并且保留了记录边界，不提供可靠性保证。
  - 在传输过程中，数据可能会丢失或重复，且无法保证在接收端按发送顺序接收数据。
- 流式套接字（SOCK_STREAM，参见后续讨论）：
  - 基于TCP实现，提供了一种面向连接的、可靠的数据传输服务。
  - 能够确保数据无差错、无重复地发送，并按顺序接收。
  - 流式套接字内部设置了流量控制，避免了数据流淹没接收方的情况。
- 原始套接字（SOCK_RAW）：
  - 与标准套接字（流式套接字和数据报套接字）不同，原始套接字可以读写内核没有处理的IP数据包。这意味着它可以对较低层次的协议如IP、ICMP等进行直接访问。
  - 当需要传送非传输层数据包（例如Ping命令时用的ICMP协议数据包）或者遇到操作系统无法处理的数据包时，就需要使用原始套接字来发送。

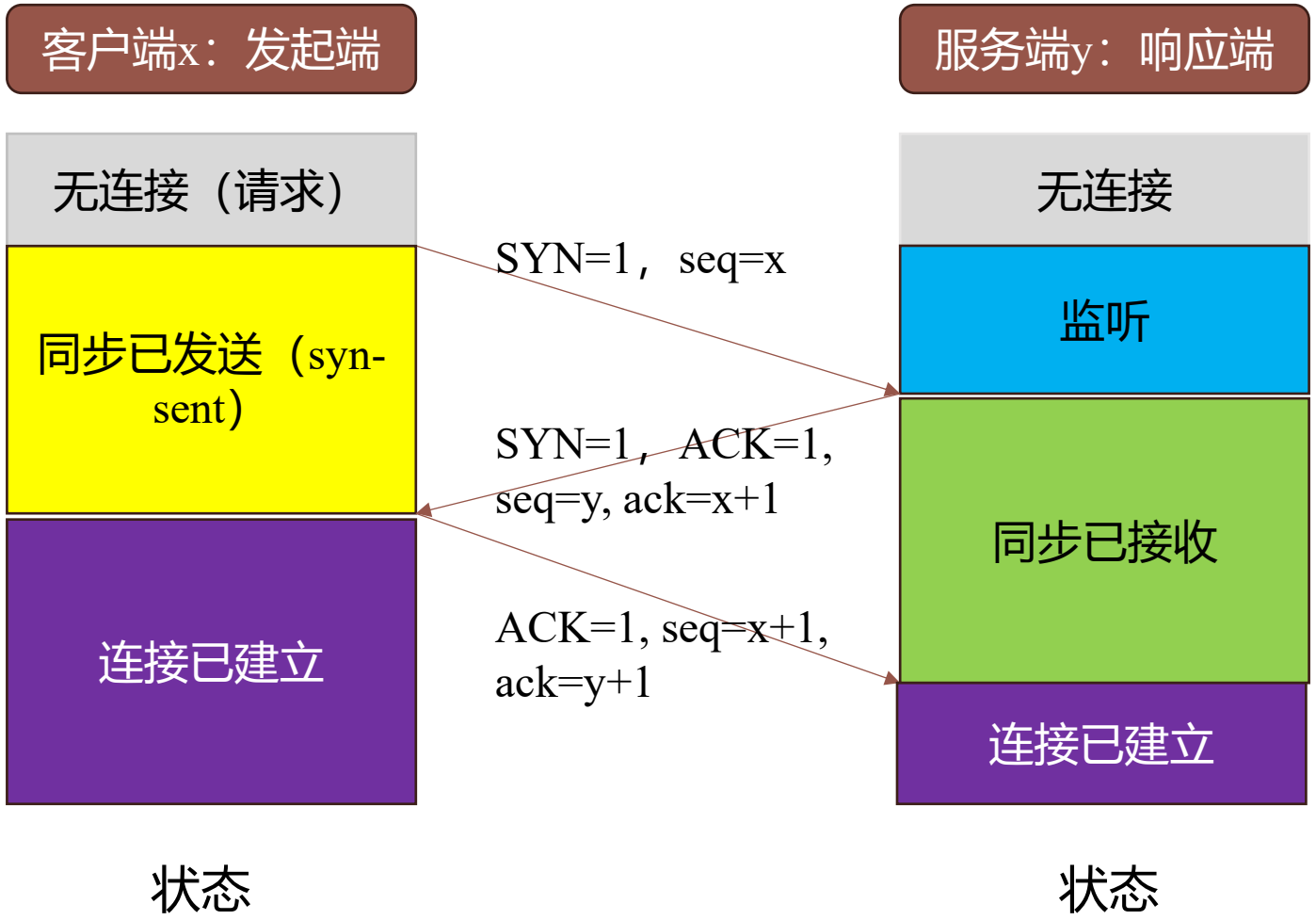
面向连接的TCP协议（传输控制协议，Transmission Control Protocol）

- “面向连接”
  - 在两个应用进程通信前，需要先相互“握手”：互相发送预备报文段。
  - 连接：与网络层或物理层连接不同，TCP连接是一条逻辑连接。
  - TCP连接对下层协议不可见。
- TCP报文结构：

```
typedef struct {
    uint16_t source_port;      // 源端口号
    uint16_t destination_port; // 目的端口号
    uint32_t sequence_number;  // 序列号
    uint32_t acknowledgment_number; // 确认号
    uint8_t data_offset_and_reserved:4; // 数据偏移量（首部长度）和保留位
    uint8_t tcp_flags:8;       // TCP标志位
    uint16_t window_size;      // 窗口大小
    uint16_t checksum;         // 校验和
    uint16_t urgent_pointer;   // 紧急指针
    // 后续是TCP选项字段和数据部分，但在这个结构中没有定义
} tcp_header_t;
```



TCP连接的建立：三次握手



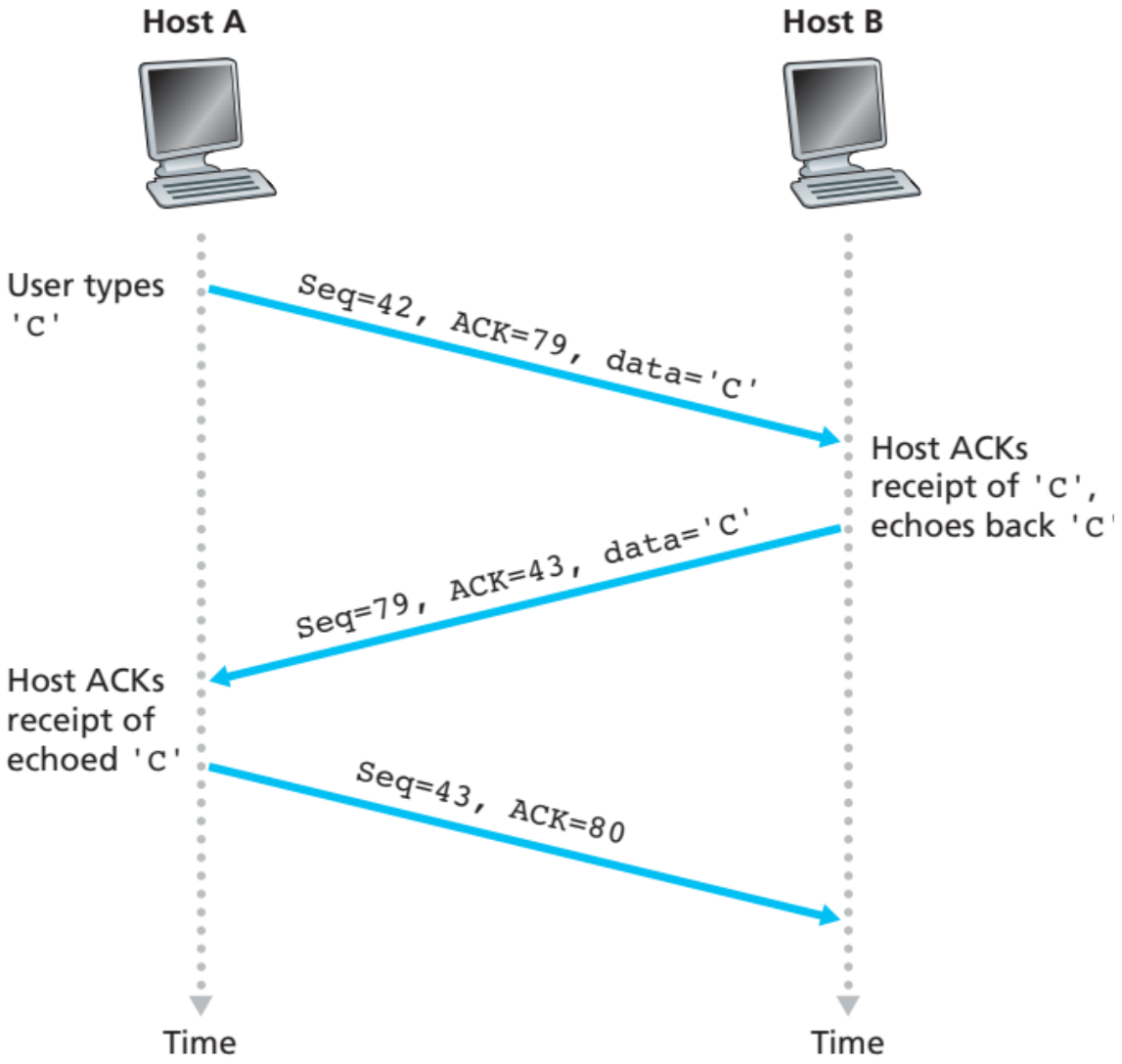
- SYN和ACK: 标志位，报文类型flag
  - SYN：连接请求报文；
  - ACK：请求确认报文；
- seq: Sequence 序号；
- ack：Acknowledgment确认号。

确认号ack = x + 1，表示服务端期望收到客户端下一个字节的序列号为x + 1

TCP连接（续）

注意：序号和确认号字段

- TCP报文
  - seq：表明报文是一个有序字节流。
  - 主机A/B填充的ack是它希望从对方主机B/A收到的下一段字节的序号（seq+1）。
  - 数据分组和确认分组都必须分别编号。
  - TCP只确认到第一个丢失字节段为止的字节段。
- 纠错机制
  - 发送端（主机A）设置超时计时器。
  - **自动重传请求**(ARQ)：只要没收到确认，发送方就重传。接收方只确认不请求。
- 举例（右图）：无差错格式
  - 注：主机A和B的报文起始序号分别为42和79。



TCP的ARQ（自动重传请求，Automatic Repeat reQuest）机制

- **累积确认：接收方返回的ACK号为已连续接收的最大字节序号+1：**
  - 发送-接收序列号配对：发送方将数据分段并附加序列号（Sequence Number）发送；接收方收到数据后，返回确认号（ACK）指明期望接收的来自发送方下一个序列号。
  - 示例：若接收方收到Seq=1-1000和Seq=2001-3000（中间1001-2000丢失），ACK仍为1001，提示需重传1001-2000。
- **超时重传（Timeout Retransmission）：**
  - 动态超时时间（RTO）：基于RTT（往返时间）计算，公式为：

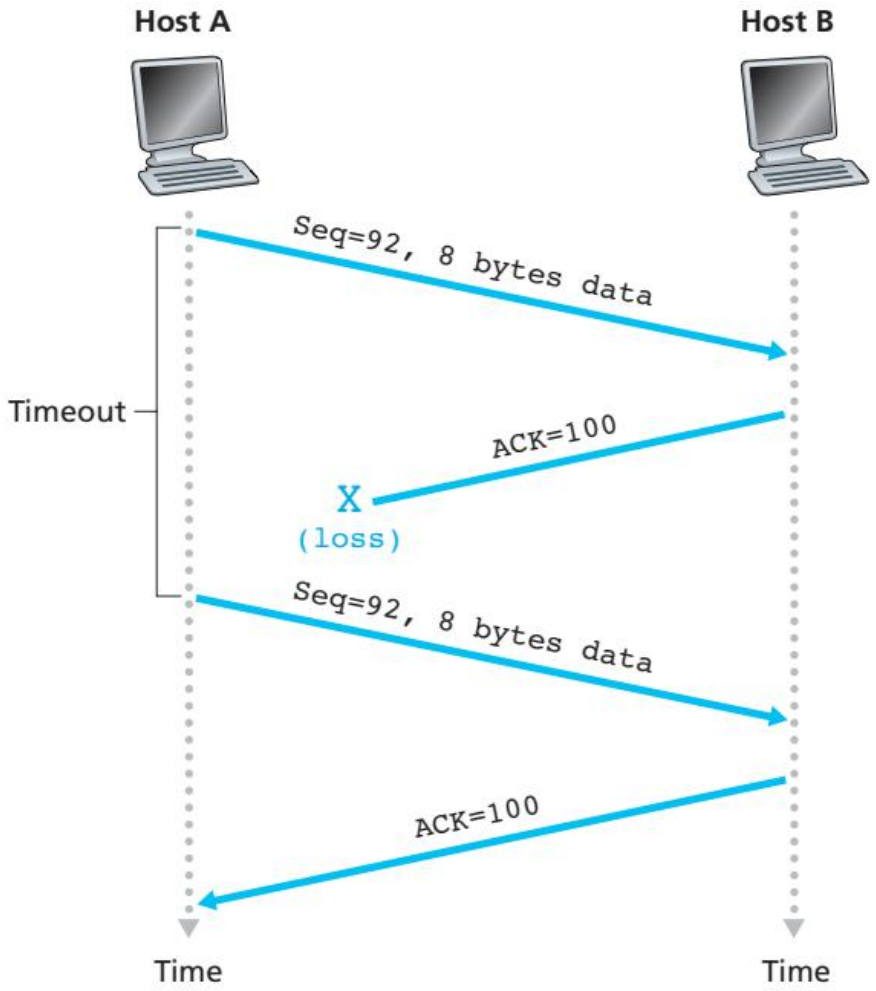
$$RTO=SRTT+4\times RTTVAR$$

其中：SRTT（平滑RTT）和RTTVAR（RTT变化量）通过指数加权移动平均更新。

  - 快速重传（Fast Retransmit）：
    - 触发条件：发送方收到3个重复ACK（如连续收到ACK=1001）。
    - 立即重传：无需等待超时，直接重传未被确认的数据段（如Seq=1001-2000）。
- **选择性确认（SACK，一般了解）：通过TCP中的SACK选项字段报告非连续接收的数据块，减少不必要的重传。**

连续ARQ：TCP报文的丢失-超时和重传

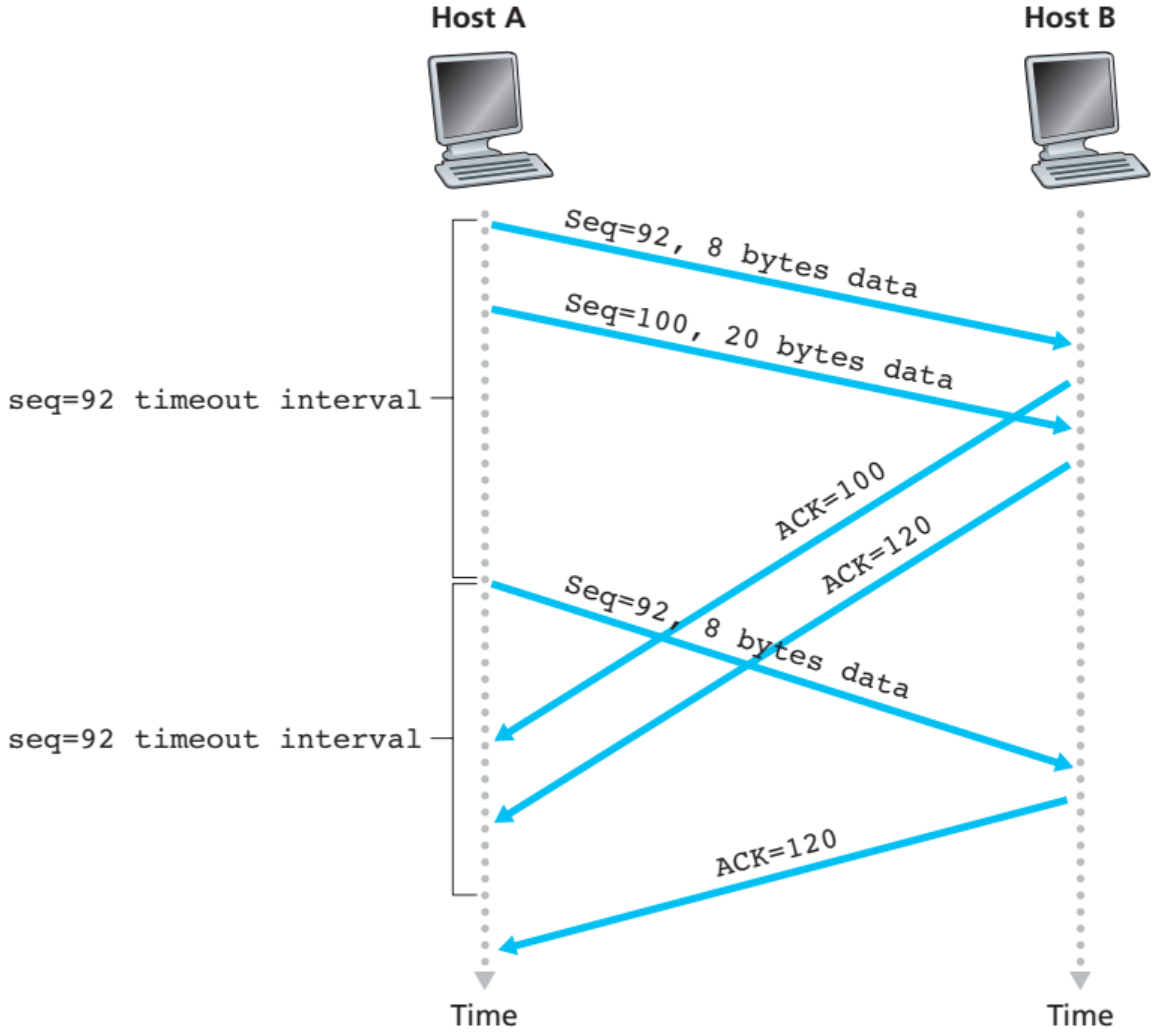
1. 由于ACK丢失导致的重传



- 接收端ACK报文丢失情况
  - （1）发送端序号为92的报文被接收端收到。
  - （2）接收端发往发送端序号为100的确认报文丢失。
  - （3）发送端出发等待超时事件。
  - （4）发送端自动重传92号报文。
  - （5）接收端重复收到92号报文（已接收），因此丢弃此报文。

连续ARQ：TCP报文的丢失-超时和重传（续）

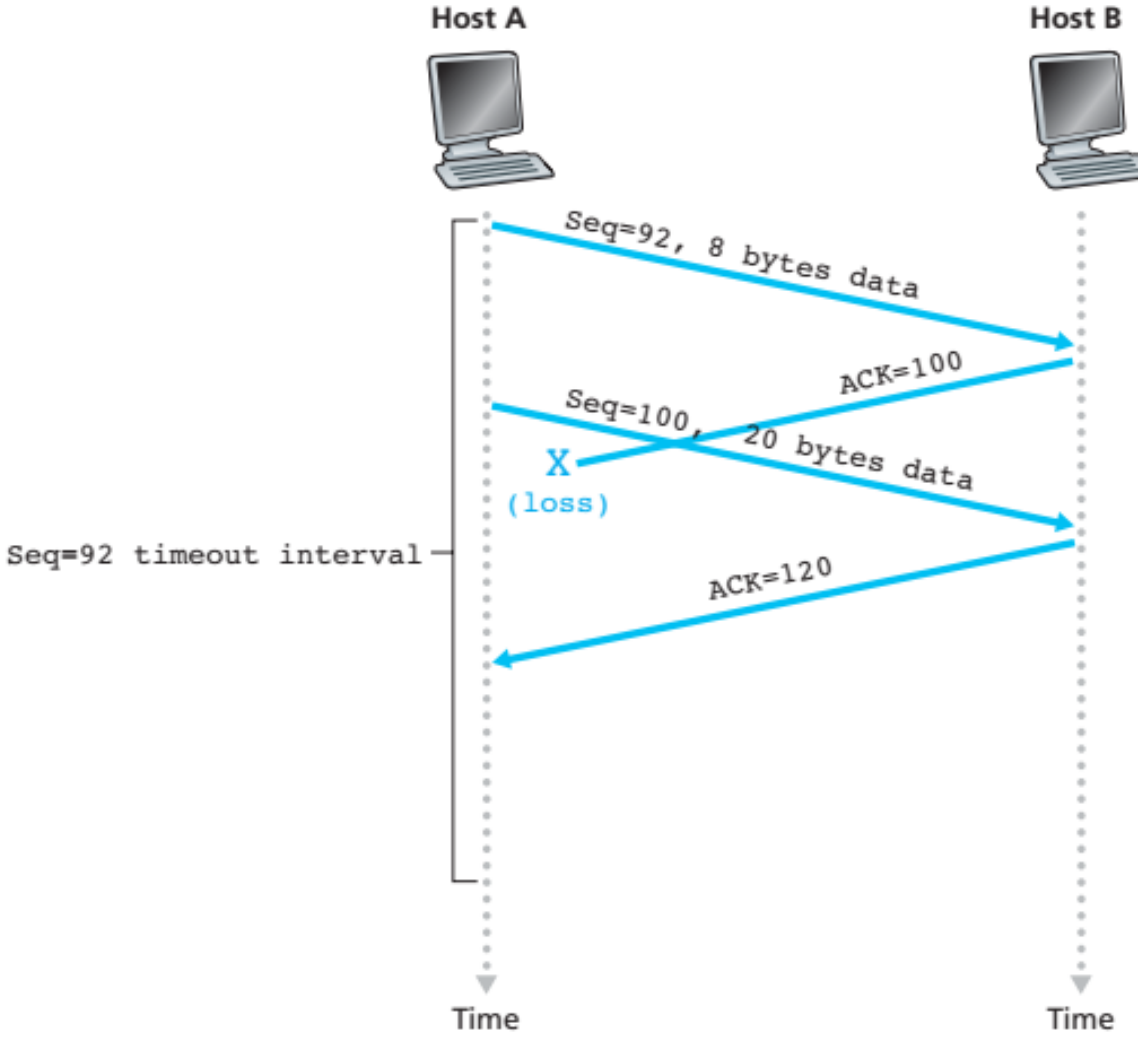
2. 由于超时导致的部分重传（超时重传+累积确认）



- 超时重传
  - （1）发送端A连续发送92号字节（长8字节）和100号字节（长20字节）两个报文。
  - （2）两个报文都被接收端B正确接收。
  - （3）92号和100号报文对应的下一字节序号，ACK=100号报文和ACK=120号报文都被A正确发送。
  - （4）发送端A在等待ACK=100号报文时触发超时事件。
  - （5）发送端A自动重传92号报文。
  - （6）100号报文对应的ACK=120号报文在新92号报文超时前收到，因此92号报文不会再次被重传。

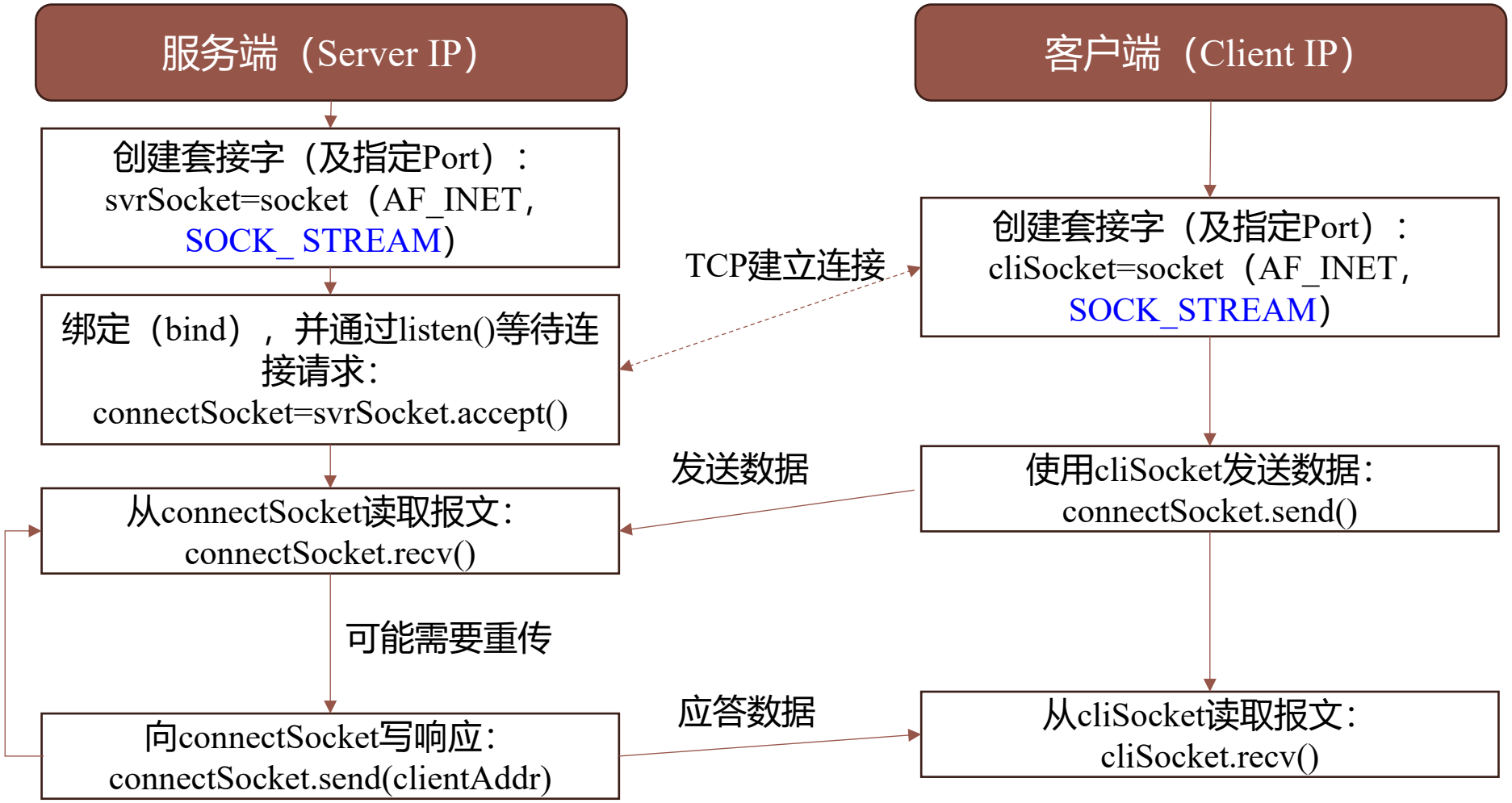
连续ARQ：TCP报文的丢失-超时和重传（续）

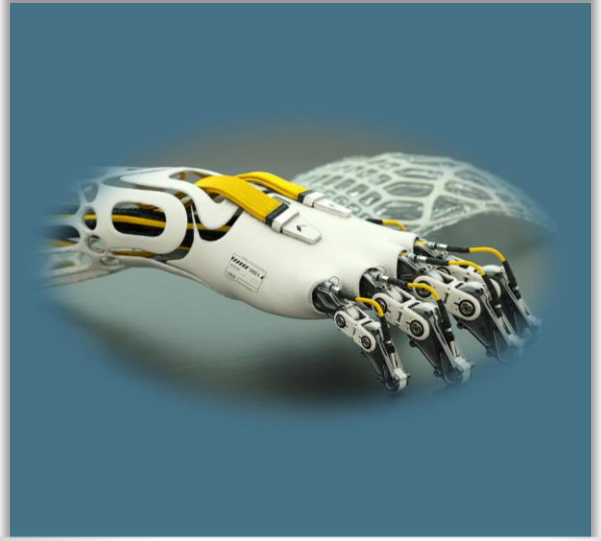
3. 有超时但不重传



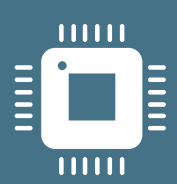
- 超时但不重传
  - （1）发送端A连续发送92号字节和100号字节两个报文。
  - （2）两个报文都被正确接收。
  - （3）92号报文对应的ACK=100号报文丢失。
  - （4） 100号报文对应的ACK=120号报文在ACK=100号报文超时前抵达。
  - （5）发送端A由所收取的ACK报文知道120字节前的所有字节都被正确接收，因此不重传92号报文。

TCP套接字编程简介


服务端（Server IP）
客户端（Client IP）
创建套接字（及指定Port）：
svrSocket=socket（AF_INET，
SOCK_ STREAM）
创建套接字（及指定Port）：
cliSocket=socket（AF_INET，
SOCK_STREAM）
TCP建立连接
绑定（bind），并通过listen()等待连接请求：
connectSocket=svrSocket.accept()
发送数据
使用cliSocket发送数据：
connectSocket.send()
从connectSocket读取报文：
connectSocket.recv()
可能需要重传
应答数据
向connectSocket写响应：
connectSocket.send(clientAddr)
从cliSocket读取报文：
cliSocket.recv()



第二章
物联网接入与组网




2.1 无线网络基础

2.2 TCP/IP协议简介

2.3-2.4 近距离和中远距离无线通信

基于802.15协议族的非IP协议

协议号：802.15.x	协议名	特性
802.15	无线个域网（WPAN)	低功耗、小覆盖范围、（大部分情况）低速率和低价格
802.15.1/2/3	基本蓝牙协议（Bluetooth）	定义了蓝牙的物理层和MAC协议，用于多媒体高速数据（55Mbit/s+）
802.15.4	Zigbee协议的PHY和MAC层	定义了低功耗、低成本、短延时、低速率（802.15.t：2Mbit/s）网络的物理层、MAC层和网络层协议
（其他）802.15.6	无线体域网（WBAN）用于医疗和娱乐的人体局域网	定义了低功耗、低成本、短延时、低速率（10Mbit/s），应用于人体穿戴式传感器、植入装置等场景
（其他）802.15.7	短距离可见光通信协议	超宽带协议，应用于照明、车载系统等场景。

蓝牙协议简介

蓝牙工作在2.4GHz频段（2.4GHz-2.4835GHz），即工业-科学-医疗（ISM）非授权（公开频段）。

当前常见蓝牙设备协议版本

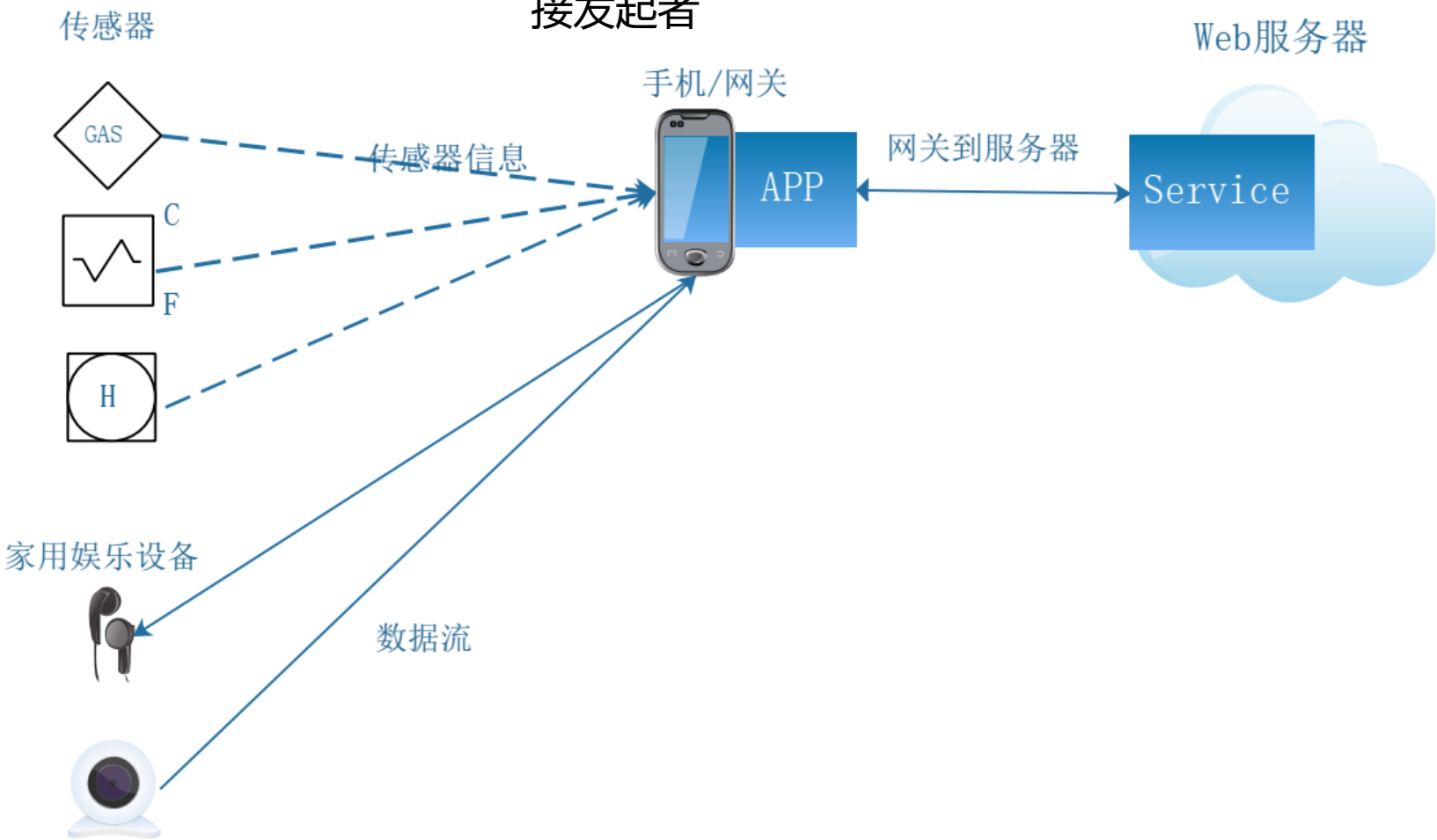
- 蓝牙4.x（4.0、4.1、4.2）：
  - 引入低功耗（Low Energy，BLE）模式；
  - 引入速率双模式：基本速率/增强速率（Basic Rate/Enhanced Data Rate, BR/EDR）模式；
  - 引入ATT（Attribute Protocol）与GATT（Generic Attribute Protocol）协议和配置，定义了设备的角色、服务的特性等内容；
  - 引入带AES加密的安全管理。
- 蓝牙5.x （5.0、5.1）：
  - 引入时隙可用掩码（Slot Availability Mask，SAM）改进媒体接入层（MAC层）接入管理，优化功耗和数据传输；
  - 引入低功耗远距离模式；
  - 引入网状网络结构拓扑。

蓝牙协议在消费者物联网（Consumer IoT）中的典型应用场景

蓝牙的主-从连接模式

从节点：广播者

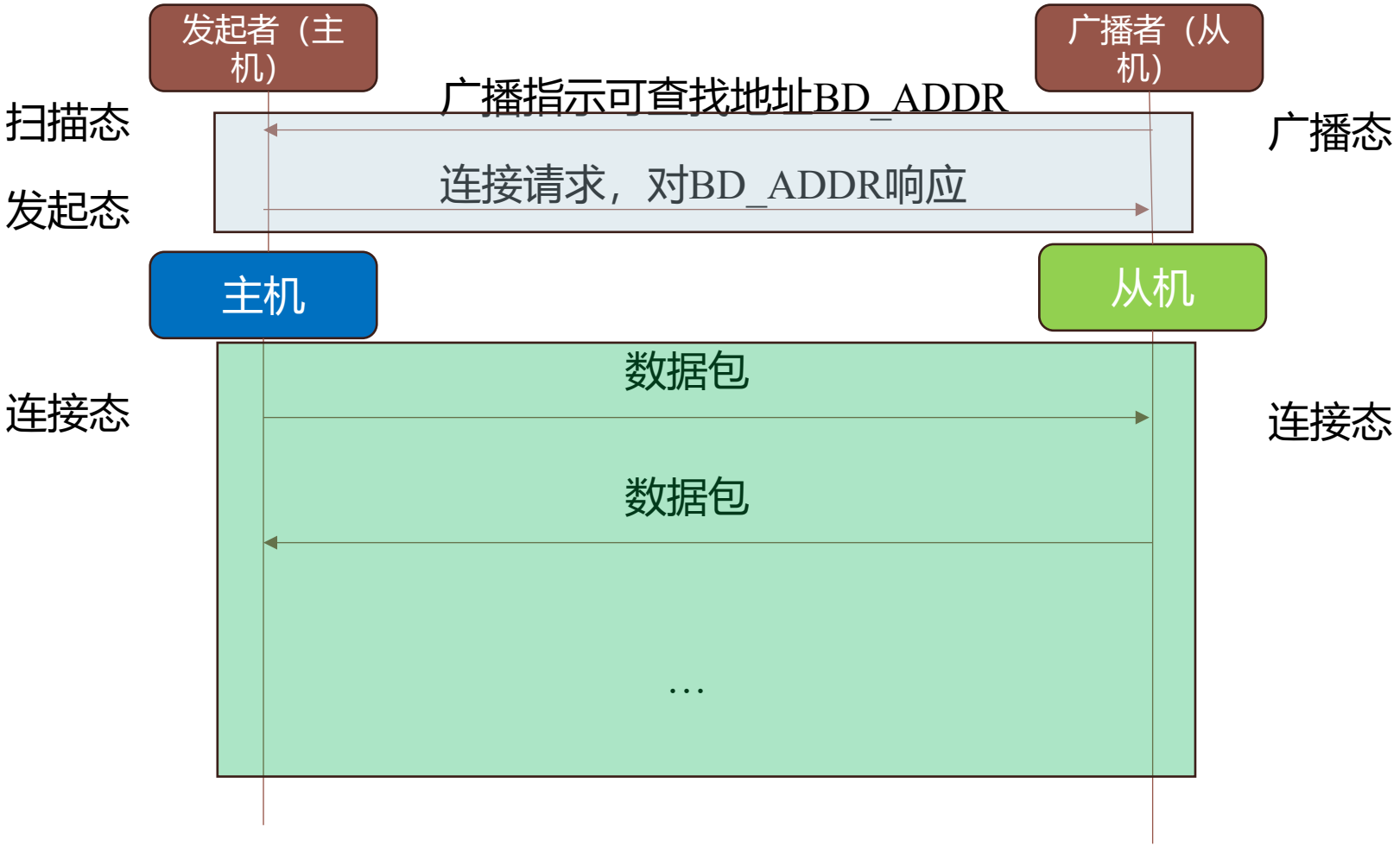
主节点（接入点）：连接发起者



（BLE）蓝牙连接中的节点身份和事件/状态

- 蓝牙WPAN中节点/设备的身份：
  - 广播者：设备发送广播数据包（多用于**从机**请求连接），并监听数据包产生的响应；
  - 扫描者：设备接收无连接意向的广播包；
  - 发起者：设备尝试建立一个连接。
- 蓝牙WPAN中的可能事件：
  - 广播：由广播者到扫描者发起的过程，提醒扫描者连接请求或在广播包中传递消息；
  - 连接：由发起者（**主机**）到广播者（**从机**）配对建立连接的过程；
  - 周期性广播（蓝牙5）：允许广播设备通过信道跳转在非主要信道（37个信道）周期性广播：
  - 扩展广播：允许扩展**协议数据单元**（Protocol Data Unit，PDU）支持广播连接或大PDU负载。
- 蓝牙WPAN中数据链路的状态：
  - 就绪态（默认）：节点待机，不发送或接受任何数据包，等待事件引发的下一个状态发生；
  - 广播态：节点发送广播信道的数据包并监听响应，对应广播事件；
  - 扫描态：节点监听其他节点设备发送的广播信道数据包；
  - 发起态：对特定节点进行监听和响应；
  - 连接态：节点与其坚挺到的设备连接，在此状态下，两个节点分别称为主机和从机。

蓝牙微微网（PicoNet）连接建立的握手过程



每个蓝牙设备都有一个唯一的48位地址码（前24位由制造商向IEEE购买，后24位由制造商分配）

广播信道

数据信道

BLE模式下，蓝牙频段被分为40个信道，每个信道相隔2MHz。其中3个信道用于广播，37个信道用于数据传输。

蓝牙接入的多址方案：FDMA+TDMA

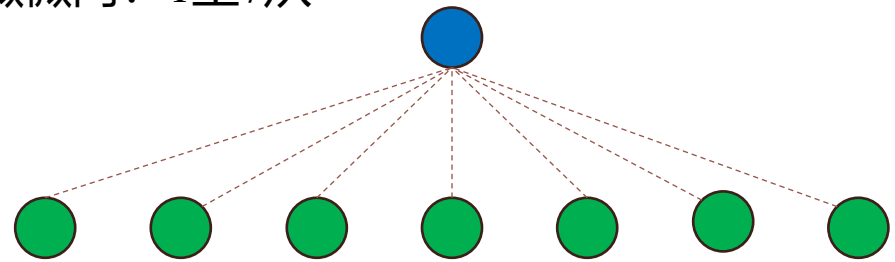
- 频分多址（FDMA）：
  - 低功耗（BLE）模式下采用40个信道划分，每信道间隔2MHz（左图：从0到39）；
  - 基本速率/增强速率（BR/EDR）模式下采用79个信道划分；
  - 跳频：蓝牙信道通过伪随机数选择，以1600跳/s的速度在上述频率间切换。
- 时分多址（TDMA）：
  - SAM（Slot Availability Mask，时隙可用掩码，低功耗（BLE）模式不可用）：由主设备生成，并广播给从机，每一个二进制位（bit）对应一个特定的时隙——“1”表示该时隙可用。
  - 在协同通信中，SAM允许两个设备指示相互的收发时隙，用来优化BR/EDR的时隙使用效率。

Frequency	Band	
2402 MHz	37	Advertise
2404 MHz	0	Data
2406 MHz	1	Data
2408 MHz	2	Data
2410 MHz	3	Data
2412 MHz	4	Data
2414 MHz	5	Data
2416 MHz	6	Data
2418 MHz	7	Data
2420 MHz	8	Data
2422 MHz	9	Data
2424 MHz	10	Data
2426 MHz	38	Advertise
2428 MHz	11	Data
2430 MHz	12	Data
2432 MHz	13	Data
2434 MHz	14	Data
2436 MHz	15	Data
2438 MHz	16	Data
2440 MHz	17	Data
2442 MHz	18	Data
2444 MHz	19	Data
2446 MHz	20	Data
2448 MHz	21	Data
2450 MHz	22	Data
2452 MHz	23	Data
2454 MHz	24	Data
2456 MHz	25	Data
2458 MHz	26	Data
2460 MHz	27	Data
2462 MHz	28	Data
2464 MHz	29	Data
2466 MHz	30	Data
2468 MHz	31	Data
2470 MHz	32	Data
2472 MHz	33	Data
2474 MHz	34	Data
2478 MHz	35	Data
2480 MHz	39	Data
2480 MHz	39	Advertise

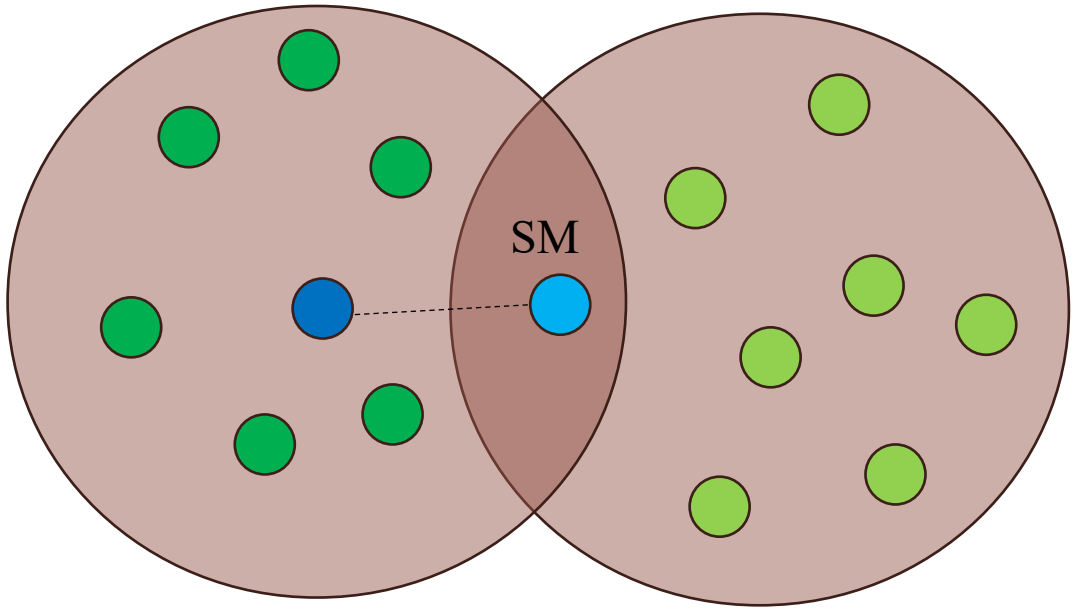
Piconet：微微网

- **BR/EDR（传统）模式**：
  - 基本拓扑结构：星型；
  - 扩展结构：**散布式网络**，即一个微微网可以通过一个从-主节点附加管理辅助网络；
  - 特点：3位寻址，即一个微微网最多支持一个主节点和7个从节点；
  - 主从节点共享一个公共信道。
- **BLE（低能耗）模式**：
  - 特点：24位寻址，即一个微微网最多支持一个主节点和（理论上）百万个从节点；
  - 每个微微网可以在不同信道上，但每个微微网同一时刻只有一个从节点和主节点连接。

传统微微网：1主7从



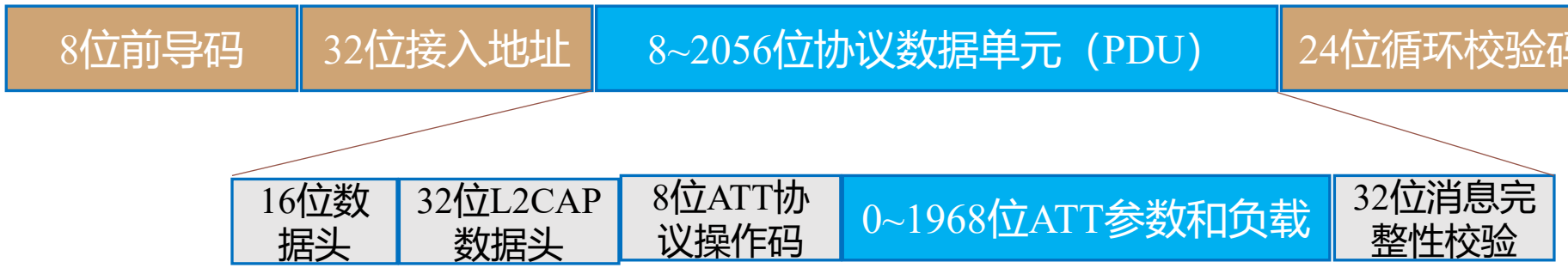
散布式网络：从-主节点作为桥接节点



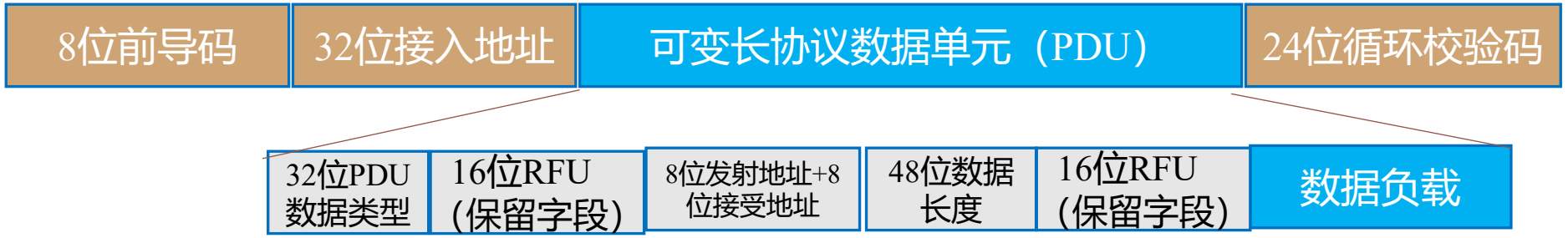
蓝牙BLE模式下的常见包结构

1. BLE广播包

2. BLE数据包

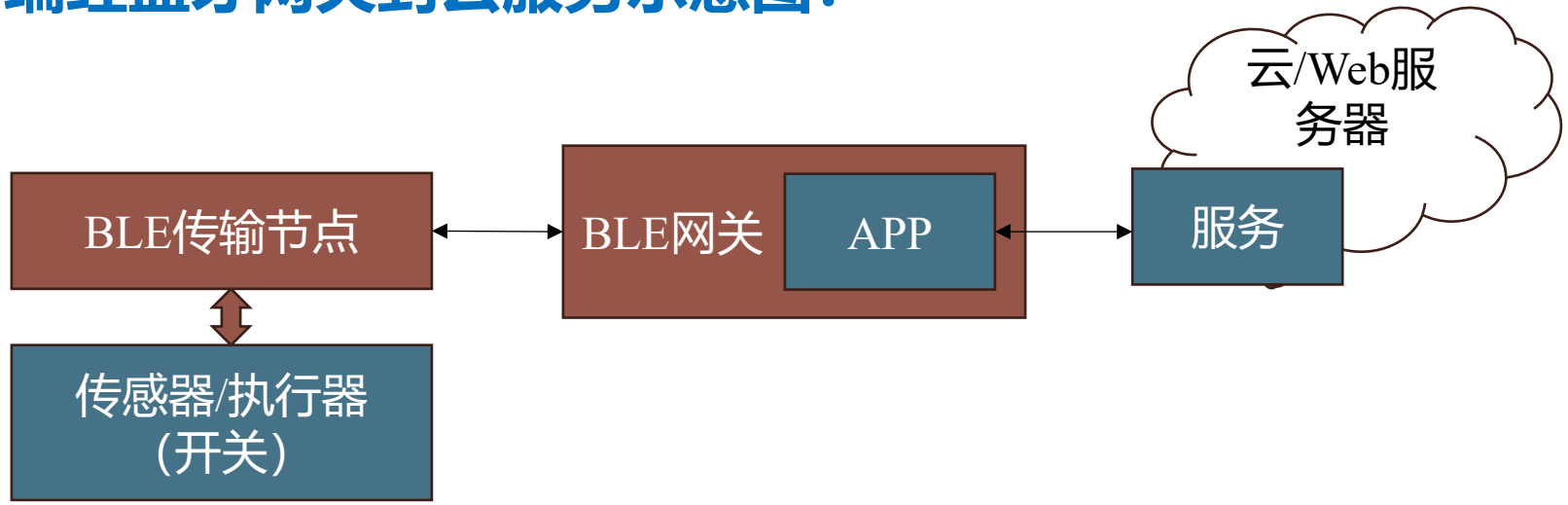


3. BLE广播数据包



蓝牙物联网结构设计

- **蓝牙物联网设计的一般化步骤**：
  - 传感-执行设备选型，蓝牙模块选型，蓝牙模式选型；
  - 蓝牙组网（网格拓扑）的设计；
  - 主节点应用程序的开发-部署，云服务的开发-部署；
  - 网关（主节点）配置和设备组网（如，BLE设备接入）；
  - 功能模块测试等；
  - 网络运行和维护。
- **数据从终端经蓝牙网关到云服务示意图：**



蓝牙设备选型：基于设备分类的功率级别和覆盖范围

- 基于应用场景选择蓝牙设备及功率等级：

功率等级类别	最大输出功等级率（dBm）	最大输出功率（mW）	最大传输距离	应用场景
1	20 dBm	100 mW	100m	USB适配器，网络接入点
1.2	10 dBm	10 mW	30m （典型用例5m）	可穿戴设备，定位信标
2	4 dBm	2.5 mW	10m	移动设备，智能读卡器，蓝牙适配器
3	0 dBm	1 mW	10cm	蓝牙适配器

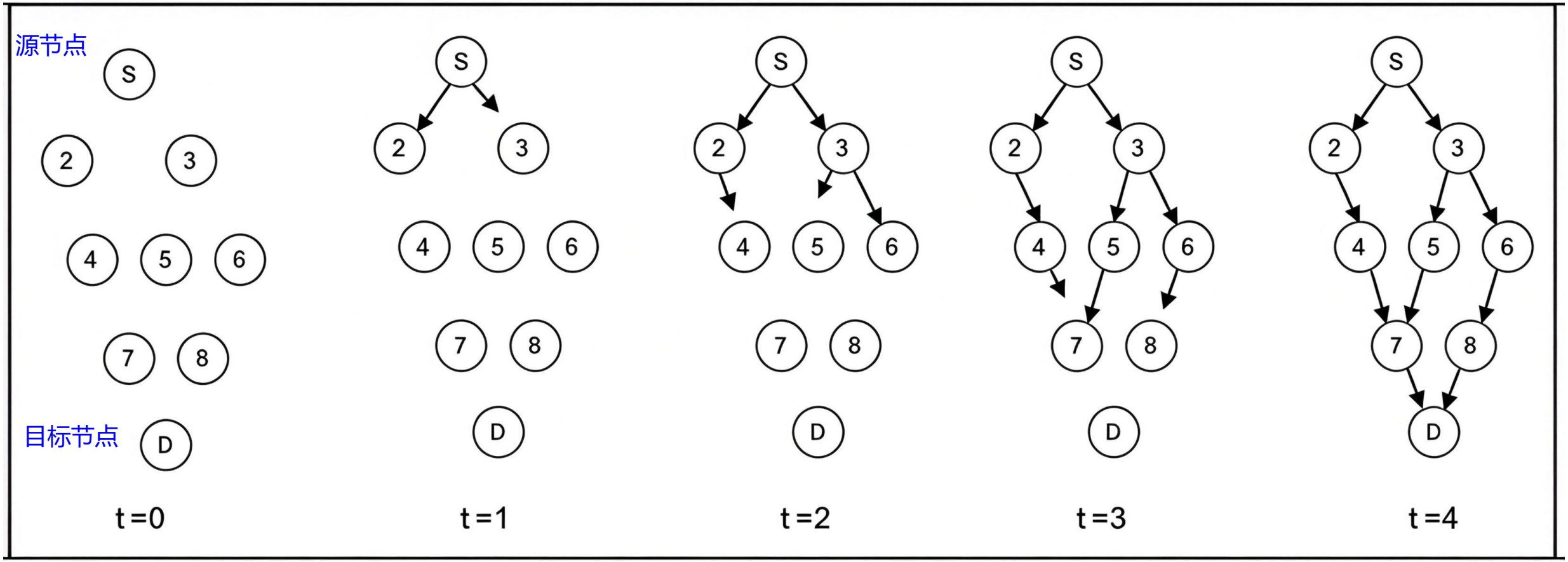
注意：$dBm = 10 \times \log_{10}(P / 1\ mW)$

蓝牙设备组网：多跳网格（Multi-hop Mesh）

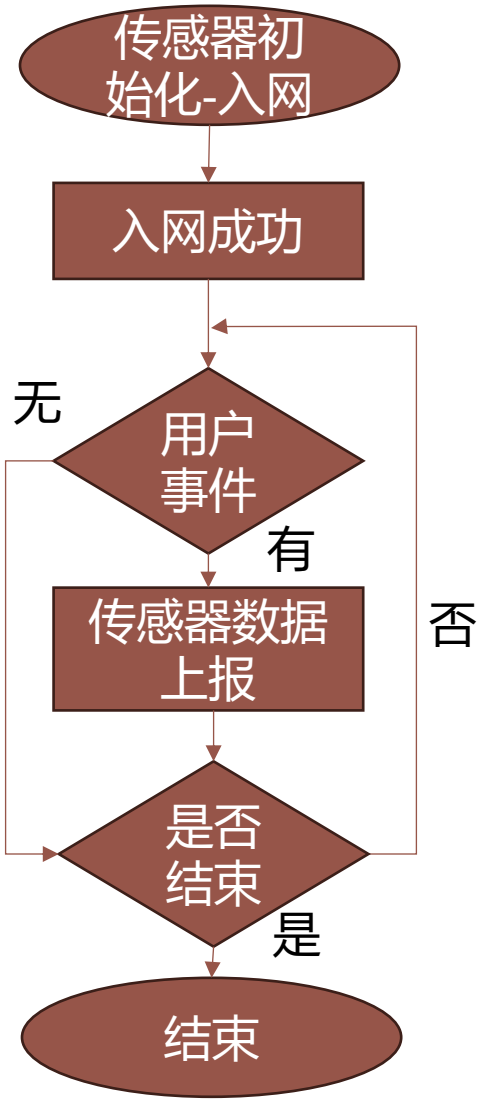
- **当蓝牙设备产生的数据需要多于一个节点（一跳）转发时**
  - 蓝牙节点组成多跳网格。
  - 节点通过启动配置，以中心化服务器秘钥分发的形式接入网络：启动后的中心节点（如网关）将基于秘钥（NetKey）哈希的方式将唯一地址（单播地址）分发给未启动设备。
- **多跳蓝牙网络中的节点**
  - 未配置节点（Unprovisioned Devices）：未加入网络的潜在设备。配置后，这些设备可以转化为下述类型的节点；
  - 中继节点（Relay）：加入网络并支持接收消息转发的节点；
  - 网格网关（Mesh Gateway）：在蓝牙子网和非蓝牙网络/连接间桥接消息的节点；
  - 代理节点（Proxy）：低版本LE节点（如蓝牙4.x）可能不支持Mesh，代理通过GATT（Generic Attribute Protocol）接口与这些BLE节点连接，BLE节点通过GATT（Generic Attribute Protocol）和代理协议读写代理消息，代理再将消息转译成Mesh网络可接受的PDU数据包。
- **泛滥式（Flooding）数据转发机制**
  - 蓝牙节点并不维护路由表；
  - 泛洪网络（Flood Network）：中继节点向所有其邻居节点（消息源节点除外）转发其受到的消息。

蓝牙多跳网格（Multi-hop Mesh，续）

- 泛洪网络中的消息转发示意图：



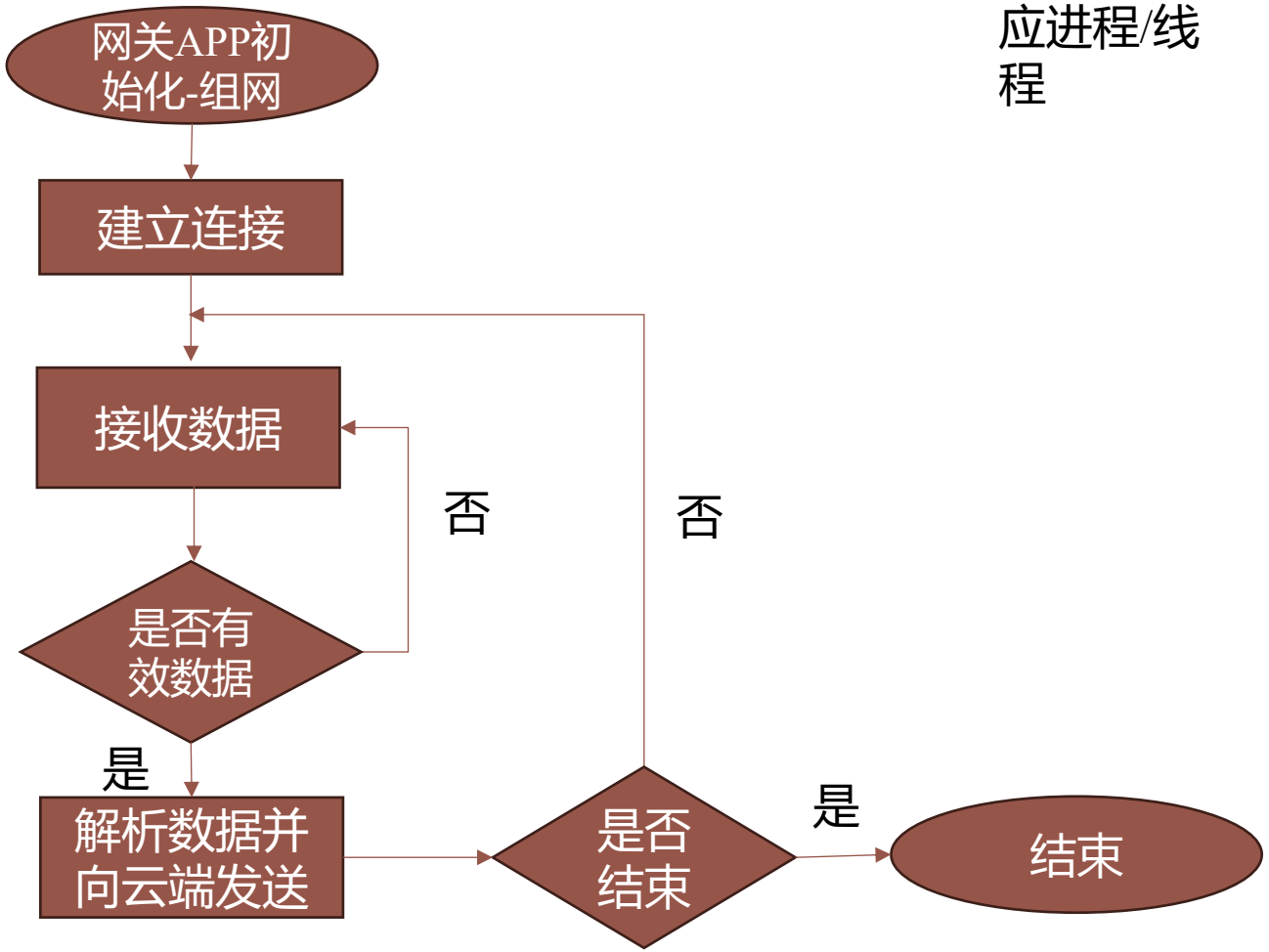
基于BLE的IoT数据推送简易执行逻辑框架



传感器端/从节点端进程

注：网络连接建立和维护细节由BLE协议栈API封装

网关端/主节点端对应进程/线程



基于Python的蓝牙服务/应用开发：PyBlueZ

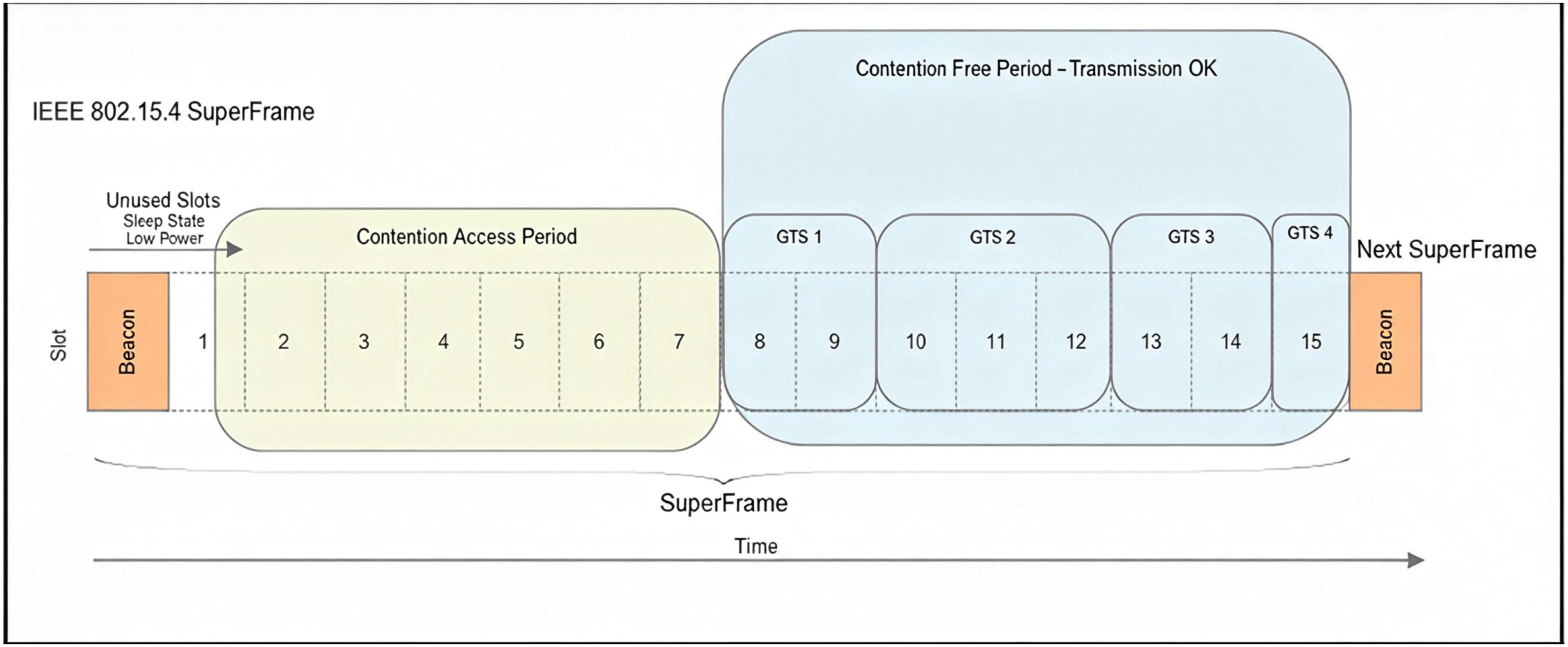
- PyBlueZ简介：跨平台蓝牙套接字开源库
  - 支持平台：Linux，Windows，MacOS， Raspberry Pi。
  - 优点：兼容套接字格式。
  - 缺点：已停止开发。
- BluetoothSocket类简介：
  - 同TCP/UDP套接字类似， BluetoothSocket使用bind()，connect()，accept()，close()，等成员函数维护蓝牙连接状态。
  - 使用recv()，send()，sendto()等函数处理数据发送和接收。
- DeviceDiscover类简介：
  - 核心成员函数：find_devices()，cancel_inquiry()等。
- 其他类似的Python开源库：
  - Bleak：相对PyBlueZ有更好的BLE支持，但并不以Socket形式封装。

IEEE 802.15.4 Zigbee（非IP协议）

- **严格意义上：802.15.4 和Zigbee有所不同**
  - 802.15.4协议（WPAN）只提供物理层（PHY）和媒体接入层（MAC）协议定义。
  - Zigbee在802.15.4的协议层基础上进一步提供了网络层（如，**组网、路由**等服务）定义。
- **802.15.4的物理层：**
  - 频段占用（中国地区）：2.4GHz （2405到2480MHz）的ISM频段。
  - 网络信道：总频段分为16个信道（Channel），信道号11（0x0b）-26（0x1a）。
  - 典型的发送端功率：3dBm~15dBm；对应传输范围30m~200m。
- **802.15.4的MAC层：**
  - 随机多址接入：采用载波侦听多路访问，即CSMA/CA方式。
  - 在基于信道竞争-Backoff的随机接入外，还提供有保证时隙（GTS，Guaranteed Time Slot），即由中心化的网络协调器指定时隙分配给特定的设备。
  - 引入超帧：多个时隙（分别用于接入竞争和预分配GTS）组成一个**超帧**（Super Frame）。

（补充了解）802.15.4的MAC层超帧（Super Frame）结构

一个超帧（Super Frame）包含一段由多个普通时隙组成的接入竞争时期（Period）和一段由多个GTS组成的无竞争时期。

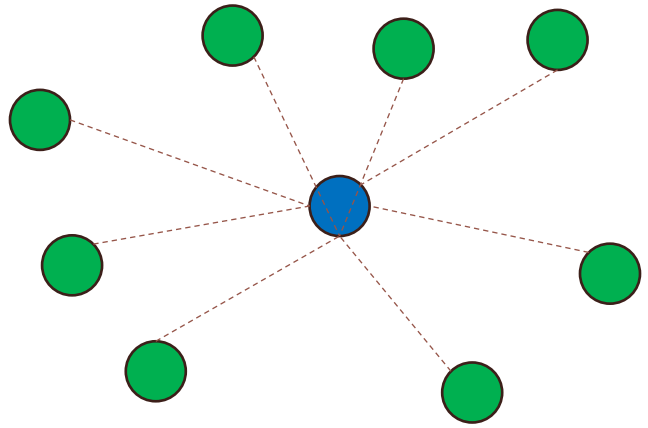


Zigbee的网络层定义

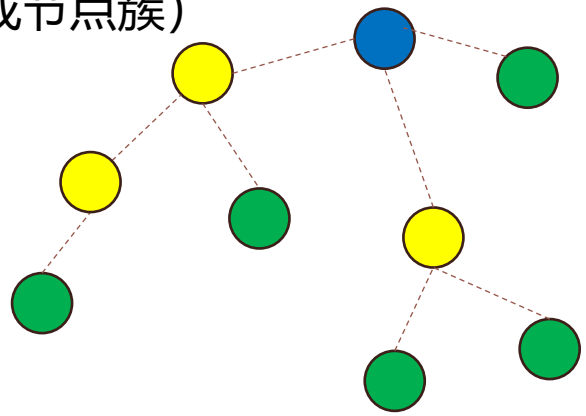
- **802.15.4定义的设备类型**
  - **全功能设备（Full Function Device，FFD）**：支持各种网络层拓扑结构中的节点。
  - **简化功能节点（Reduced Function Device，RFD）**：只能作为网络边缘/终端节点。
- **Zigbee中网络层定义的节点类型**
  - **协调器**（Coordinator）：一个网络中只能存在一个协调器。它负责选择在网络初始化时信道频率，建立网络并将其他节点接入网络，提供路由等其他服务。协调器必须是FFD（**全功能设备**）。
  - **路由节点**（Router）：当Zigbee网络组成多跳（Multi-hop）网络，负责对接收数据进行转发并允许子节点通过它接入其他网络。路由节点必须是FFD（**全功能设备**）。
  - **终端节点**（End Device）：终端节点只能向协调器或路由节点发送（或接受）数据，无法转发或通过它将其他节点接入网络。终端节点一般是RFD（**简化功能节点**）。

Zigbee网络层协议支持的网络拓扑结构

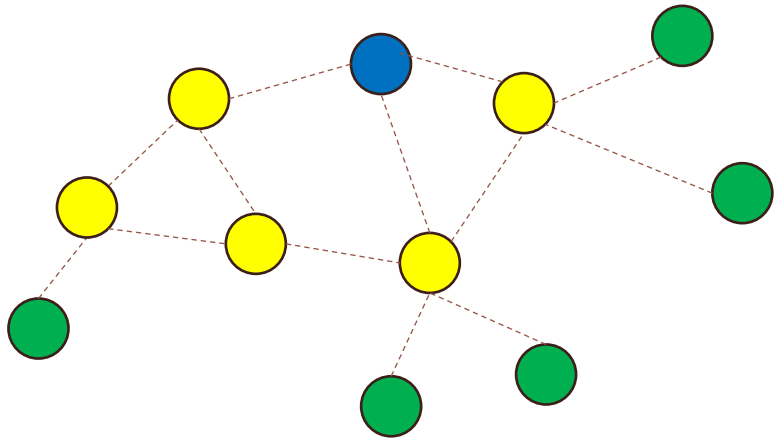
（1）星形网络：单跳中心化网络



（2）树状网络：多跳网络（根节点是协调器，非叶节点是路由节点，子树组成节点簇）



（3）网状网络：多跳网络（多个FFD组成骨干网，骨干网节点连接RFD或FFD形成子网）

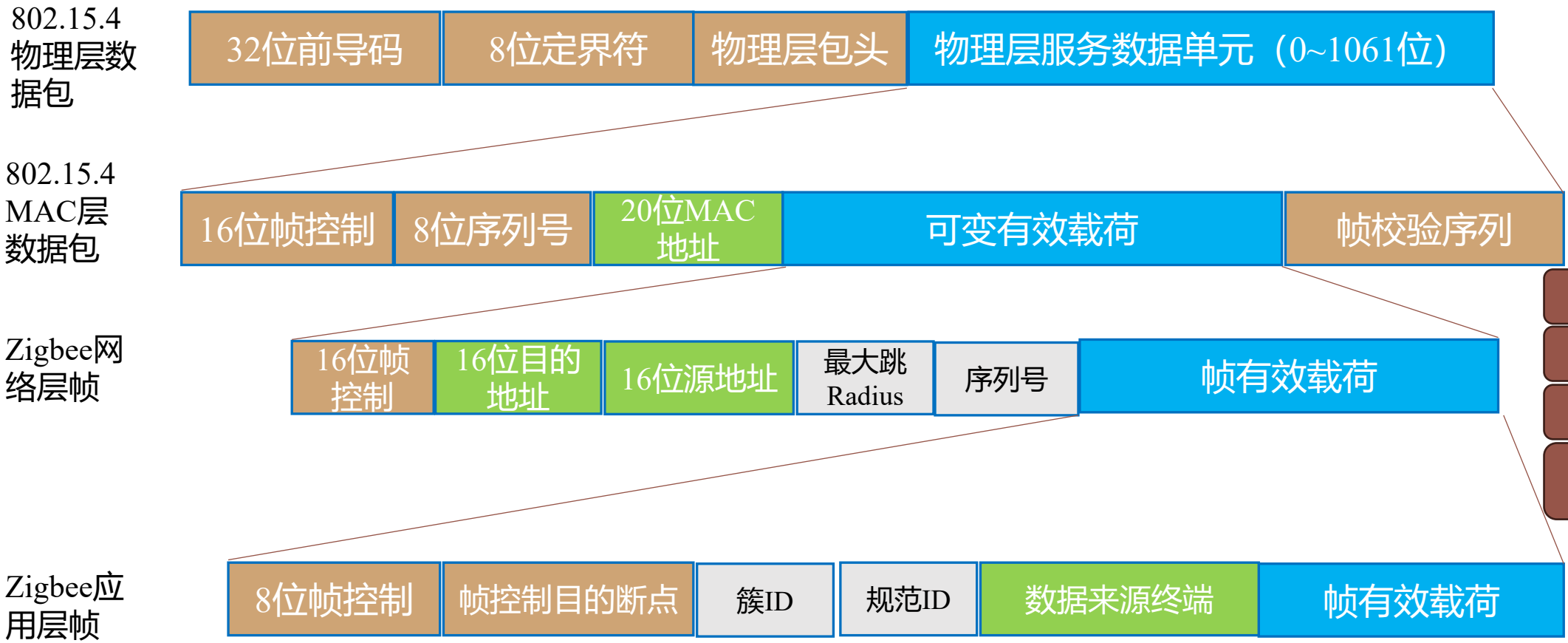


协调器FFD

路由节点FFD

终端节点RFD

Zigbee寻址和数据包结构

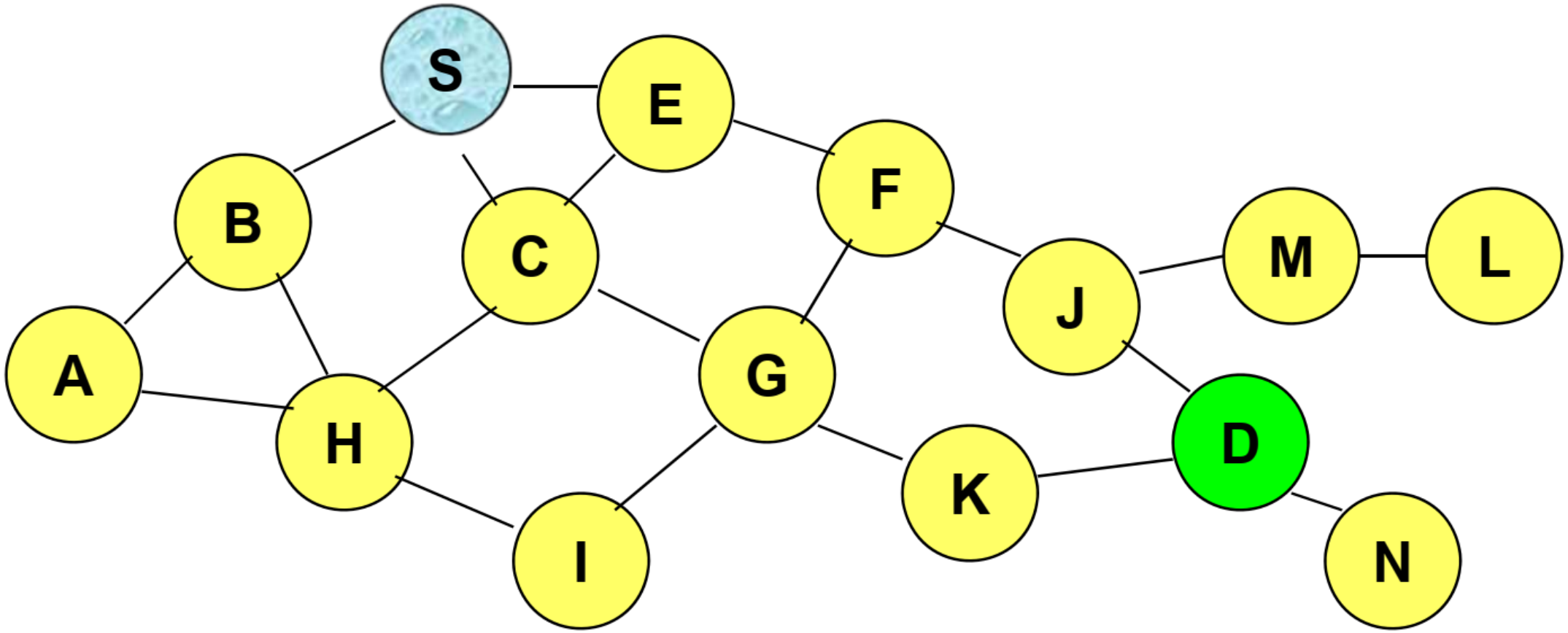


Zigbee路由算法

- 基于AODV：Ad-hoc On-demand Distance Vector （自组织按请求路由矢量）路由
  - RREQ（Route-REQ路由请求帧）：从源节点向目的节点通过广播<broadcast ID, addresses>到邻居节点请求建立路由（ Path Table）。
  - RREP（Route-REP，路由回复帧）：收到路由请求的节点向源节点沿反方向建立反向路由表（Reverse Path Table）。
  - 路由节点只处理最早一次收到的RREQ帧，其余舍弃。
  - 在Zigbee网络层，**只有具有路由能力的节点（FFD）**才发起路由请求RREQ并建立路由表。
  - 当路由点收到来自多条路径的RREP时，选择最小路径损的路径作为到达目的节点的路由。

Zigbee路由算法示意（AODV）

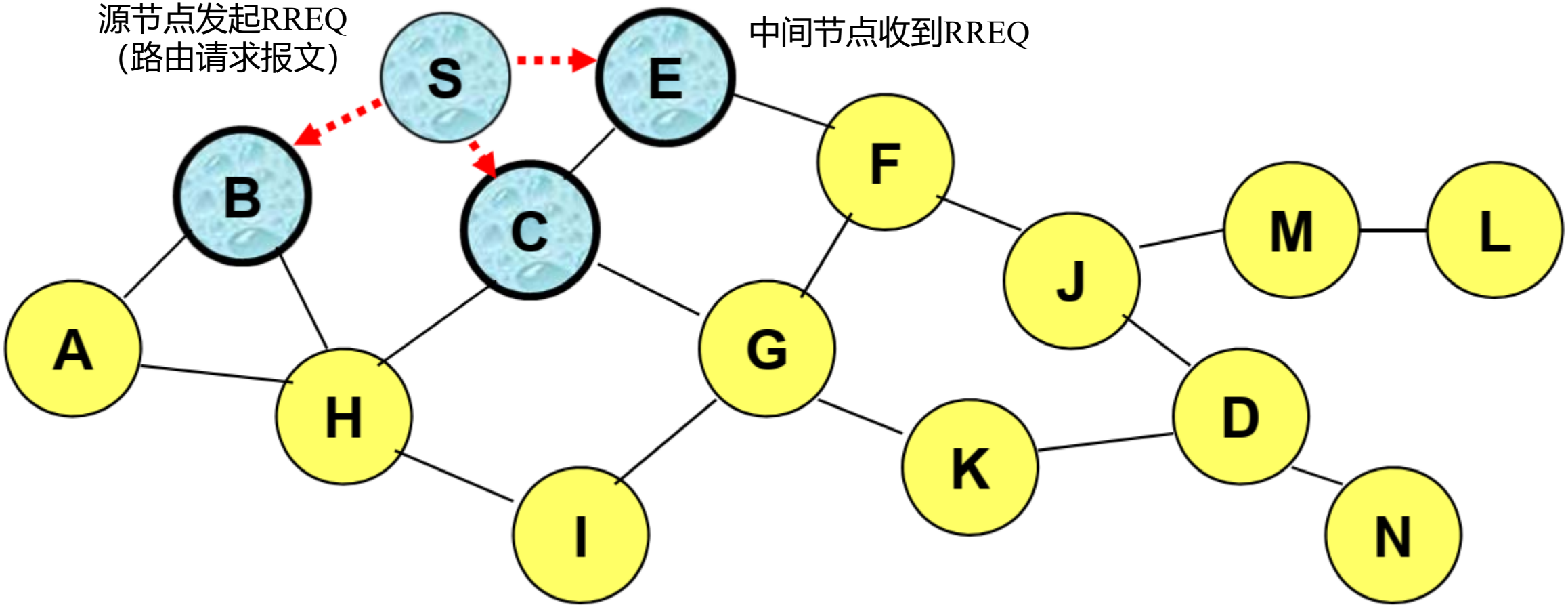
在下图节点组成的Zigbee骨干网中，寻找S-D间的路由



Zigbee路由算法示意（AODV）

在下图节点组成的Zigbee骨干网中，通过泛洪广播，寻找S-D间的路由

喊话询问（广播RREQ）

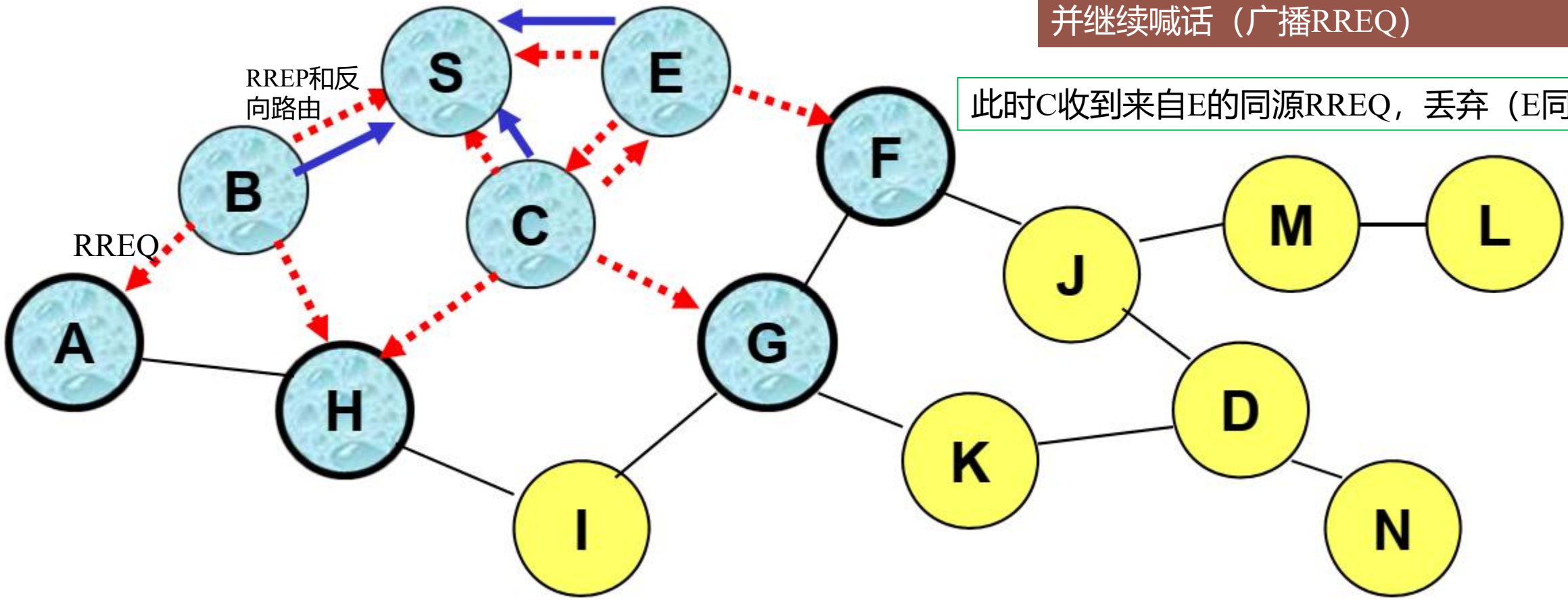


Zigbee路由算法示意（AODV）

在下图节点组成的Zigbee骨干网中，寻找S-D间的路由——中间节点收到RREQ后：

- 建立或更新一个指向源节点的反向路径（记录哪个邻居发来了这个RREQ），以便后续将回复传回。
- 如果它没有到目的节点的有效路由，则继续广播转发这个RREQ。

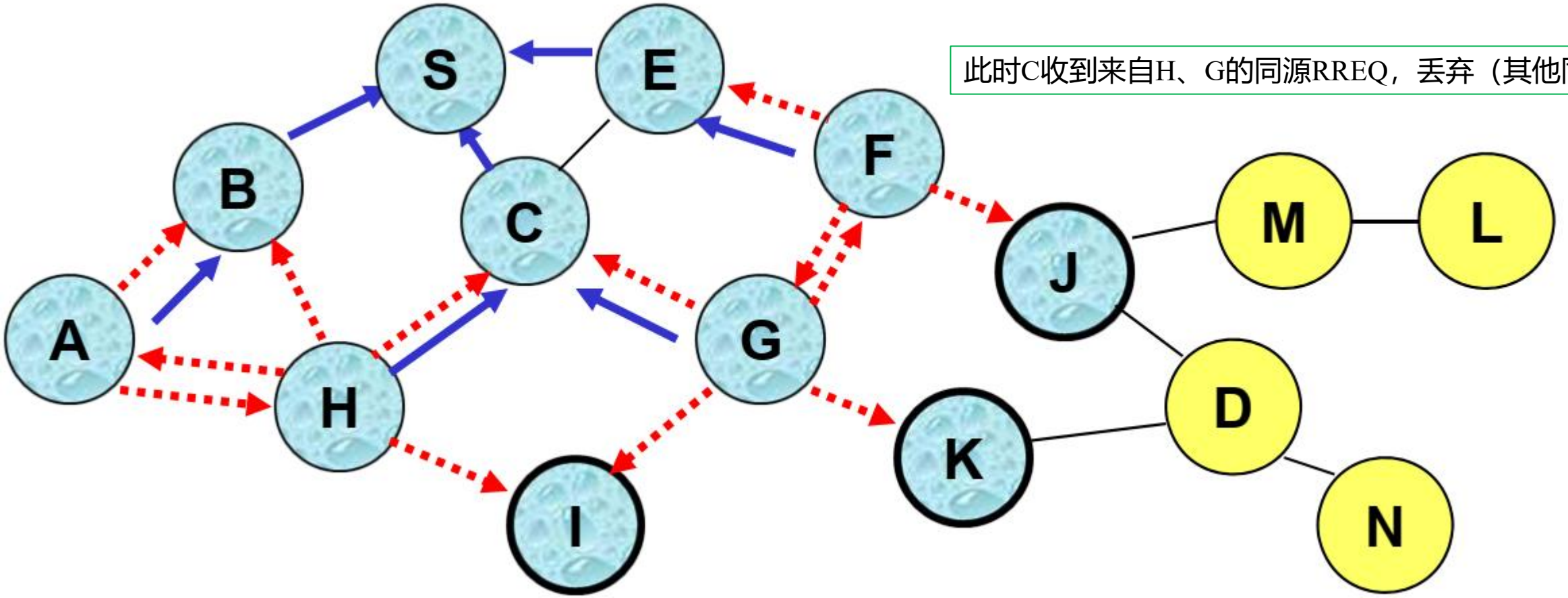
中间节点记忆来路（建立反向路径）
并继续喊话（广播RREQ）



此时C收到来自E的同源RREQ，丢弃（E同）

Zigbee路由算法示意（AODV）

在下图节点组成的Zigbee骨干网中，寻找S-D间的路由：继续泛洪广播



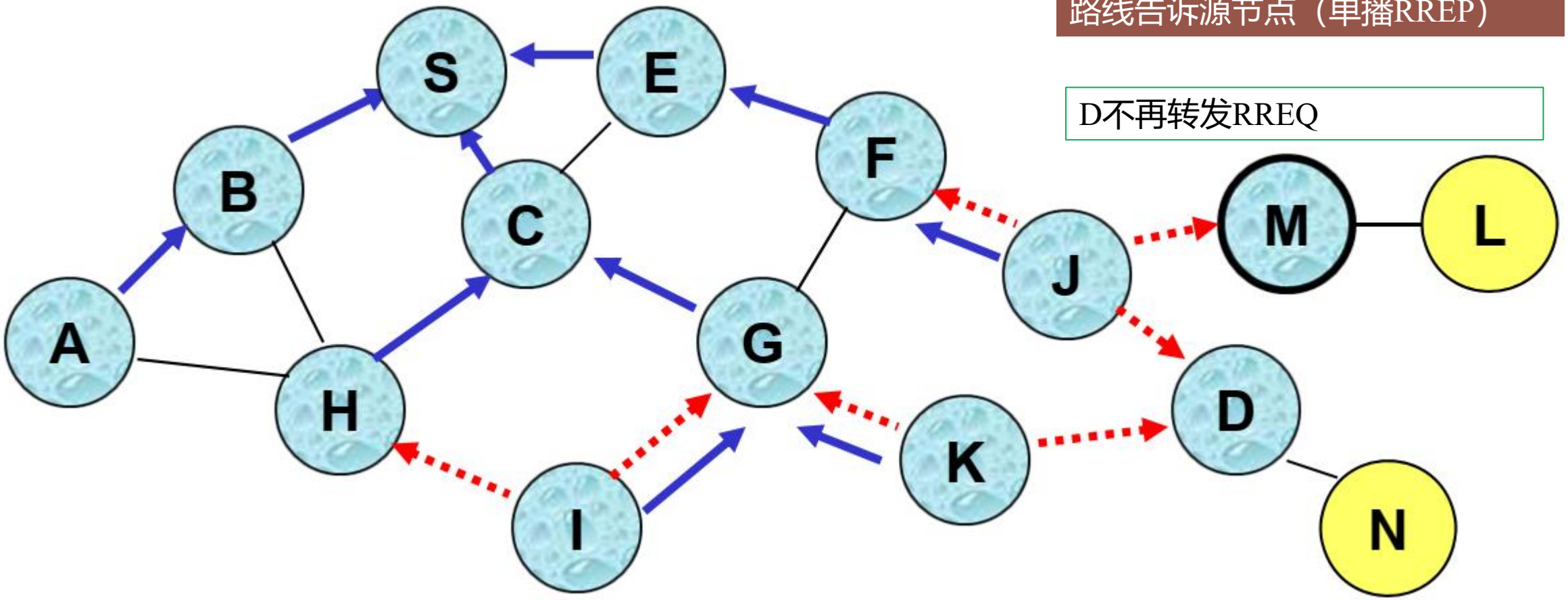
此时C收到来自H、G的同源RREQ，丢弃（其他同）

Zigbee路由算法示意（AODV）

在下图节点组成的Zigbee骨干网中，寻找S-D间的路由（继续）：目的节点回复RREP

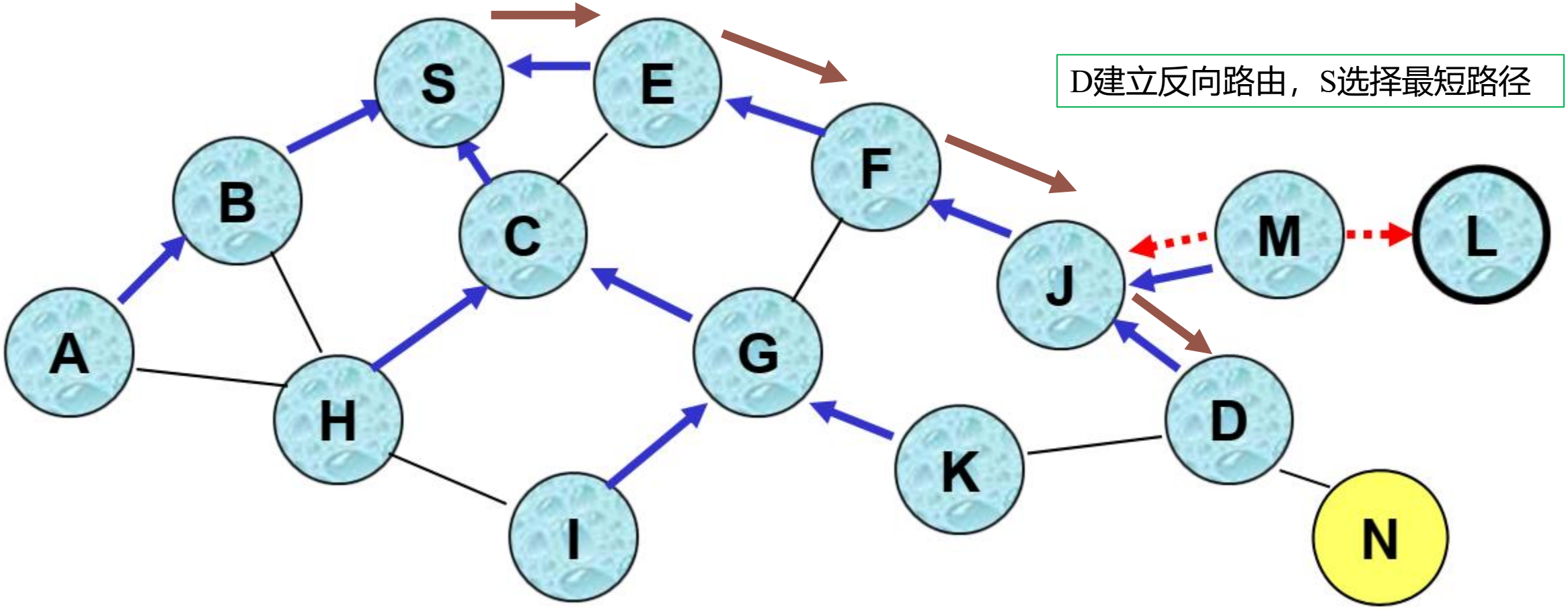
目的节点沿着大家记住的来路把路线告诉源节点（单播RREP）

D不再转发RREQ



Zigbee路由算法示意（AODV）

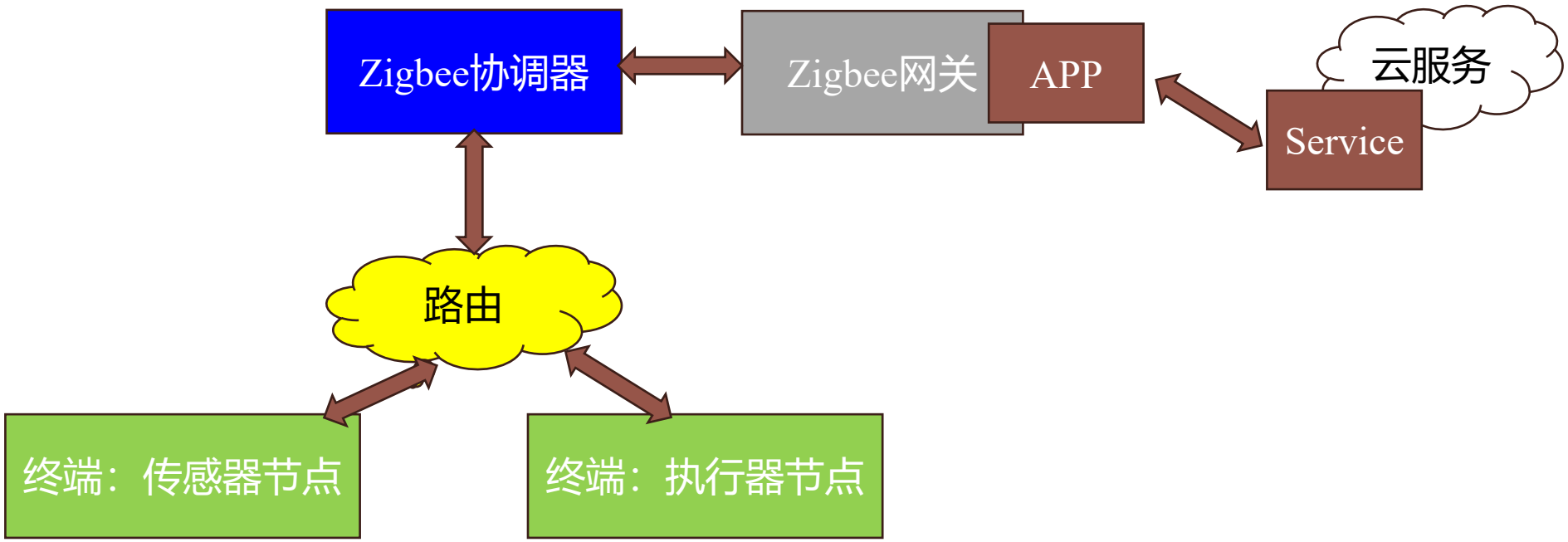
在下图节点组成的Zigbee骨干网中，寻找S-D间的路由（继续）



D建立反向路由，S选择最短路径

基于Zigbee协议的IoT网络的应用系统开发

基于Zigbee协议的传感-执行IoT网络应用框架

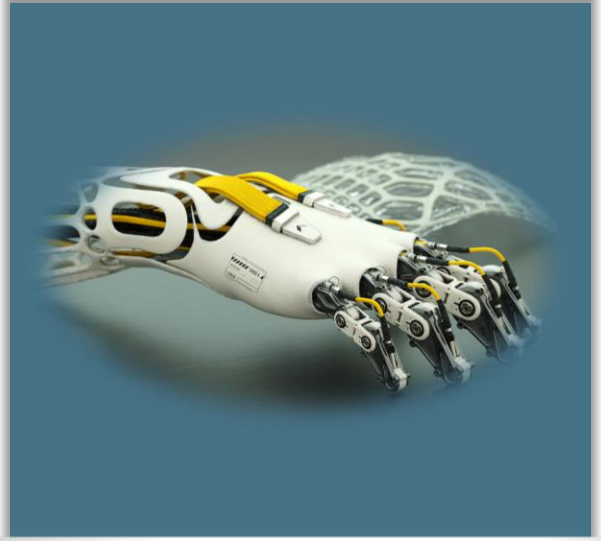


基于Python的Zigbee服务/应用开发

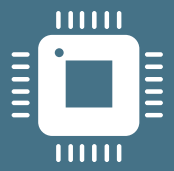
- 由于硬件厂商的私有实现不同，Zigbee的开发除依赖协议接口库外，还需要通过额外的厂商Device Handler获取底层射频固件（firmware）支持，如
  - 德州仪表的Z-Stack官方库，Digi Xbee官方库、 Silicon Labs的官方库等。
- zigpy： 较受欢迎的开源Python库
  - Hardware Independent。
  - 支持zigbee模块到USB或GPIO经由UART串口协议的数据转换。

讨论：物联网的应用场景1

- **机电工程学院的会议预约系统需求**
  - 机电楼的房间最大直线距离>100m；
  - 预约系统通过会议室的屏幕显示预约情况（以小时为单位）；
  - 预约系统接入由预约人从网络或专门的实体终端接入；
  - 会议室大约有10个，由于硬件条件所限，预约控制设备和显示设备有供电，但不能接入校园网。
- **系统设计讨论**



第二章
物联网接入与组网




第二章 作业

1. 无线通信基础

(1) 信道容量是如何定义的？决定信道容量的三要素是什么？

(2) 已知某语音信道的带宽是4kHz，如果要求信道的信噪比为30dB，试求信道容量C。如果信道上的最大数据传输速率是19.2kb/s，试确定信道所需的最小信噪比是多少？

(3) 已知，网络中某节点待发送100Mb的数据。设其工作信道带宽为8kHz，信噪比为20dB，则发送这些信息最少需要多少时间？

(4) 请通过查询资料回答：在无线通信系统中，采用差错控制（如FEC编码）的目的是什么，牺牲了什么指标？

2. 无线通信网络协议栈

(1) 试借助图示简述CSMA/CA中的退避（backoff）机制?

(2) 试根据课上所述时隙ALOHA协议的传输效率分析方法，给出纯ALOHA协议的期望最优传输效率值，试通过查阅资料给出基于泊松分布的解?

(3) 在CDMA多址协议下，假设通信连接A的8位时隙编码序列为00011011，连接B的8位编码序列位00101110，C的编码序列为01011100，D的编码序列为01000010。检验上述编码是否符合CMDA多址编码规则？（根据惯例）在传输中将编码序列中的0表示为-1，现各连接的接收端收到一个码片序列（-1, +1, -3, +1, -1, -3, +1, +1），则哪些连接发送了数据，发送的数据值是什么?

3. TCP/IP协议

(1) 某IPv4地址的二进制形式为：10001011.01010110.00001101.00100011，请将其转换为10进制表示格式并判断它是哪类IP地址?

(2) 简述UDP和TCP协议的特点?

(3) 编程：利用python中的socket库和thread库，在一个进程中分别建立一个TCP服务端线程和一个tcp客户端线程。要求：在建立TCP连接后，从客户端线程通过给定套接字和接口向服务端线程逐个发送字符串“I love KUST！”中的单个字符，在服务线程收到逐个后输出在命令行窗口。

- 指定：本地虚拟地址127.0.0.1；端口号8080。
- 要求1：编程作业以A4纸打印形式提交。
- 要求2：客户端收到空数据后关闭连接（客户端对回复确认信息的处理不做统一要求）。