协作物联网技术文档

1. **技术简介**

针对传统物联网非同源设备难以互联互通的问题，协作物联网技术通过改进网络终端的数据管理和通信机制，将网络中的通信业务管理由应用层转移至协议层底层，通过定义统一的应用数据封装格式，设计高效的节点间通信协作策略，有效避免不同通信业务的竞争，实现对带宽资源利用效能的最大化。

基于边云协同的思想，协作物联网同时支持本地自组网和接入云端两种模式。当存在联网需要时，节点通过智能网关计算最佳路由，选择合适的链路，与云端节点进行通信。而对于无需联网的本地应用，节点只在局域网中进行数据的获取、传输和处理，并针对设备行为做出智能化决策。具体来说包括两个方面：

1. 物端自主：各个节点可以自主进行环境中的数据的发现、获取、传输和处理。

该过程中，节点实时感知所处环境及网络状态的变化，并依据所设计的信号处理机制作出动态响应，在有限状态机中完成状态跳转，无需人工干预。

1. 物间协作：物间协作体现在物物协作、端云协作、人机交互三个层次。
2. 物与物协作.

节点间出现业务竞争时，基于资源请求大小、实时性要求、使用时长等业务特性设置优先级，通过时间片轮转、最短作业优先、最高响应比优先等策略实现竞争业务的智能化排序及调度，在资源受限时实现利用效率最大化。

1. 端与云协作.

节点复杂度通常较低，数据处理能力有限，对于高计算复杂度的业务，需要网络中空闲节点配合实现分布式计算。针对分布式处理仍不能满足要求或者远程数据采集监控任务，节点可通过网关接入云端网络，将数据上传到云端完成高性能计算。

1. 物与人协作.

人机交互是物联网用户体验是否良好的关键因素。没有良好的人机接口、交互显示界面，较低的语音指令识别准确率、延迟率，即使网络通信性能再好，也无法带给用户良好的物联网使用体验。

基于协作物联网技术统一报文数据格式，标准化用于通信的软硬件接口，可以设计制造适用于不同厂商产品的模组。对于物联网终端设备厂商而言，无需自行设计设备软硬件接口标准，只需在产品中采用标准化模组，即可消除同源壁垒，实现非同源物联网设备的互联互通。对于物联网新产品开发者而言，只需要采用标准的应用编程接口，可以专注于构思应用场景和实现业务逻辑，省去了底层开发的繁文缛节，有效解决了非同源节点之间的通信协作和数据共享问题。对于产品用户而言，可以不必局限于特定厂商特定型号的产品，扩大了产品选择范围。

1. **网络拓扑结构**

常见的无线局域网拓扑结构包括由中心控制节点的星形网络和无中心控制节点的mesh网格网络，如图2.1所示。

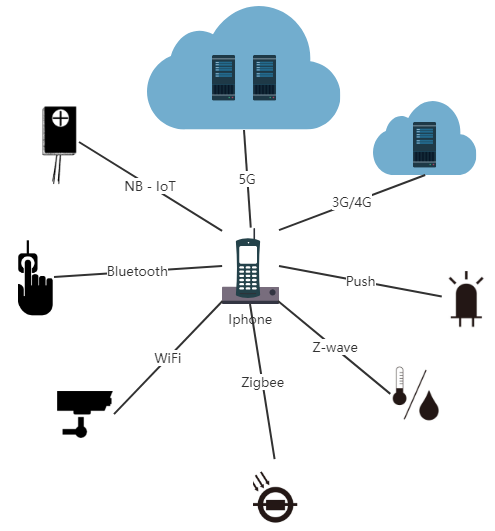
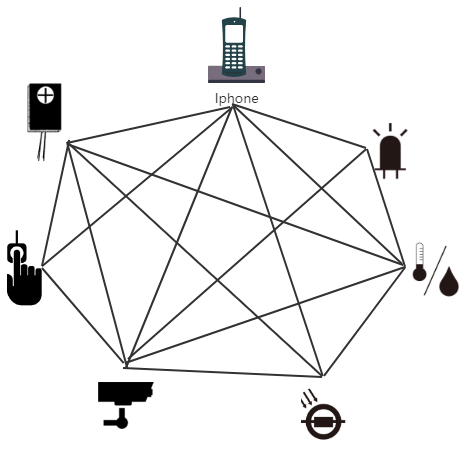
**** ****

图2.1 (a) 星形网络-有中心控制节点 图2.1 (b) Mesh网格网络-无中心控制节点

现有的大多数物联网/无线局域网应用可以看做是星形网络的模式，手机或其他控制终端作为中心节点，控制其他节点执行动作。虽然可以实现非同源节点的通信，任何两个非中心节点进行通信需要中心节点进行中继转发（交换），导致通信灵活性低，不同节点对同一事件驱动难以有序连贯地执行动作。为了有效解决该问题，可采用组网灵活，易于部署的Mesh网络。在Mesh网络各非同源终端节点内部，嵌入数据格式和用户接口一致的标准化模组，即可实现非同源终端节点间的互联互通。此外，通过将数据分散存储于Mesh网络各个节点，可在保障数据安全性的前提下实现数据的分布式可扩展存储方案。

1. **协作物联网关键技术**

建立一个协作物联网生态需要解决节点相互识别与信息交换以及通信管理与数据管理两大关键问题。

**3.1 节点相知相识算法**

为了实现节点之间的高效互联，提出节点间相知相识机制nSIIC (Nodes’ Sense, Identification and Inter-Collaboration, nSIIC)，该算法主要包括三个流程：环境节点探测、身份信息识别、交互协同操作。

1) 环境感知阶段

首先介绍环境感知阶段的整体流程。当物联网系统首次启动并完成初始化后，各节点按照一定频率广播和扫描包含基本地址信息（IP或MAC地址）的数据报文。交互数次后，各节点完成对其它节点基本地址信息的获取，将网络所有节点的信息记录在网络节点状态信息表（可由数组或者链表实现）中。在数据交换开始前，各节点对网络节点通信链路质量进行评估。评估依据包括广播和扫描过程的通信时延以及所接收信号的强弱等物理参数。通常情况下会发生多次交互，可以计算这些物理参数的均值、方差等统计特征，根据这些统计特征评估网络节点间通信链路质量是否足够好，足够稳定。若是，既可以采用TCP协议建立面向端到端可靠传输的通信连接，也可以使用UDP面向无连接的通信机制。否则应使用UDP协议进行节点间通信。需要说明的是，网络层通信协议（TCP或UDP）的选择并不是必须步骤，实际使用时可以根据实现方便与否、模组特性、通信协议栈、用户习惯等其他因素自行决定选用哪一种网络层通信协议。若采用TCP协议，则在本阶段（环境感知）需要通过三次握手机制建立连接。若采用UDP协议，则节点向其它节点发送数据时需启动超时重传定时器，若在规定时间内未收到应答报文，则重发一次，直至达到规定次数。

下面介绍发生故障重启后或新加入网络的节点（简称作新入网节点）重新加入已有网络时的过程。新入网节点Q通常不了解网络状态（节点数、节点间链路通信质量）以及各节点的基本地址信息，但网络中的其它节点已经完成了本阶段（环境感知）的交互，均已将网络所有节点的基本信息记录在网络节点状态信息表中，这与物联网刚建立起来时所有节点均为新入网节点的情况有所不同。在这种情况下新入网节点发出广播数据报文后，网络节点对其进行响应和应答时，可以采用以下两种策略中的一种：

(i)在请求回送报文中，网络中既有节点不仅要发送节点自身的地址信息，也要发送所掌握的网络中其它节点的状态信息。发出请求的Q节点收到报文后根据报文时间戳觉得是否更新。

(ii)在请求回送报文中，网络中既有节点除了发送节点自身状态信息外，增加传送本地记录的网络节点数信息。发出请求的Q节点收到报文后记录网络节点数并缓存，后续收到其它报文则根据其时间戳决定是否更新。

两种策略的选择取决于网络环境状态：当节点数目少或节点间通信链路质量较好时，选择前者；若节点数目多或节点间通信链路质量较差时，选择后者。节点数目多少的阈值可根据实际场景设定，节点间通信链路质量的判断可由前述介绍网络通信协议选择时的准则（通信时延、接收信号强弱等参数的均值、方差）进行评估。

2) 身份识别阶段

第一阶段完成后，各个节点仅获知网络中其他节点的最基本的IP地址或MAC地址信息。然而仅仅有这最基本的信息是远远不够的。由于数据格式标准的不同，节点无法解析收到的来自其他节点的数据报文，发送给其他节点的数据报文也无法被解析。为了解决这一问题，需要节点彼此之间有更为充分的认识，即节点需要掌握其他节点更多的信息。本发明利用“名片文件”对这一部分信息进行格式化规范描述。

名片文件是将节点通信所需的源地址、目的地址、校验位、数据包长度、节点设备类别、节点优先级和节点支持的功能操作信息等字段按照一定顺序封装为类或者结构体形式的数据集合，包括内涵和外延两部分。其内涵为用于实现节点间相识和可互操作的数据信息集合，包括节点的源地址、目的地址、校验位、数据包长度、节点设备类别等信息，即节点的身份页；而外延为节点常用功能，包括节点本身支持的功能列表及对应的控制编码、物理量及赋值范围、感知数据列表及数值范围。通常情况下可将常用的物联网设备按照类别分组，然后确定每组的常用功能及对应指令。为每个设备提供的指令编码包括mesh类网络通信指令码、basic\_func类设备功能指令码、device类设备自身状态指令码三类。每个节点接收来自其他节点的名片文件，也向其他节点发送自身的名片文件。网络每个节点都存储有一张网络节点状态信息表，保存中网络其它节点的名片文件所包含的节点身份信息，每当收到新的报文时，按照表中的名片文件对报文进行解析，提取报文中的有效信息。

3) 交互协同阶段

前两个阶段完成了节点智联过程中“联”的部分，共同为第三个阶段提供了通信基础设施，而节点智联过程中的“智”体现在第三个阶段，即节点与节点之间的交互协同上。根据环境物理参数的动态变化，自适应、实时动态作出序贯的动作决策.基于实际应用场景、业务逻辑以及用户需求，以及网络各节点消息通信记录利用统计分析方法确定动态响应及交互的原则，按照if-else式逻辑代码实现。

另外一种是利用多层感知机网络或脉冲神经网络等网络行为模型，将多路传感器数据作为输入，执行动作或决策作为输出，先训练后推理，根据传感器采集到的外界环境参数物理量的变化，动态实时交互，自适应地进行负反馈调节，直至系统各项指标参数达到要求的范围内，从而保持整个系统处于稳态。

若收到其它节点的数据请求，则采集本地传感器的实时数据并解析，将解析后的数据推送至发出请求的节点。此外，当传感器采集数据发生变化时，系统缓存的名片文件会随之更新，并根据设计好的规则，将更新后的数据提前推送至目标节点。

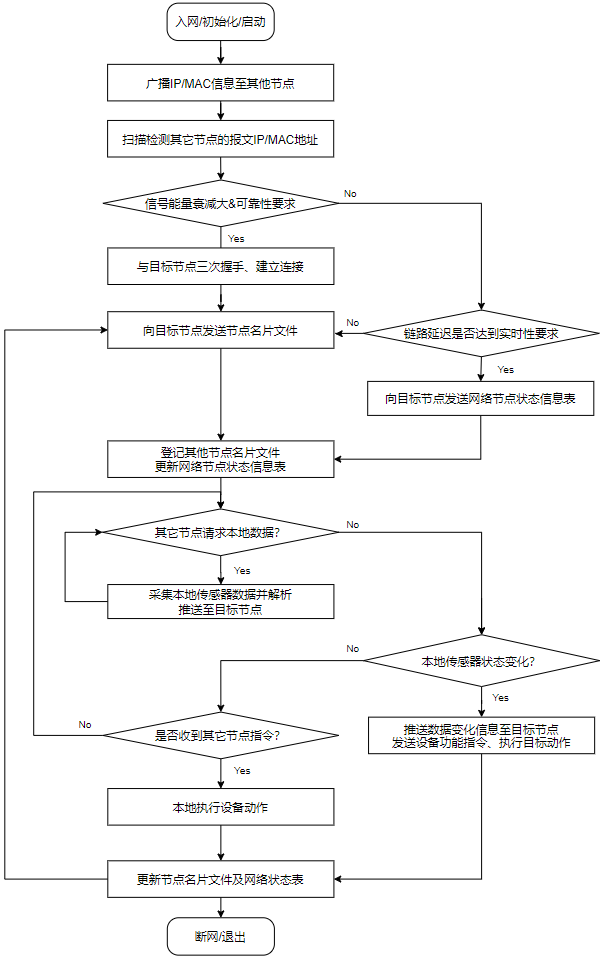


图3.1节点相知相识算法的流程设计

**3.2 通信管理与数据管理策略研究**

除了相知相识流程中的名片文件更新，数据和通信报文的管理还体现在数据的智能获取方面。本节提出基于统计信息的消息预取及推送策略。

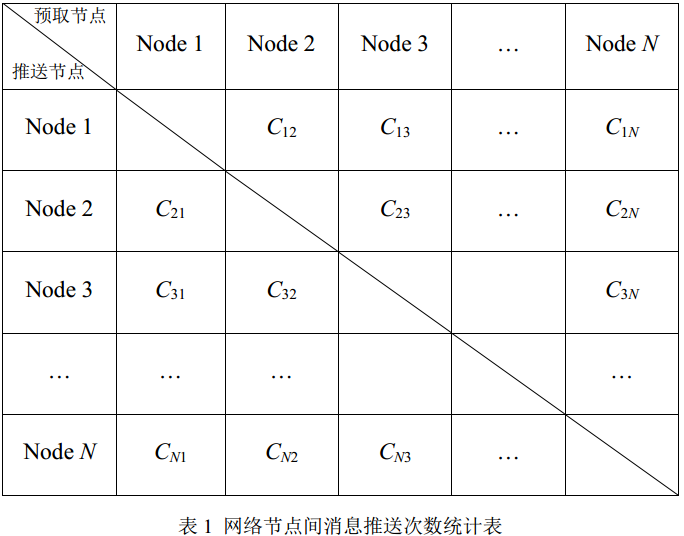
在通信芯片协议层操作系统内部，常见的消息推送模式包括基于用户请求式推送和服务侧主动推送两种方式。基于用户请求的消息推送策略属于传统信息交互方式，节点计算过程中产生数据需求后，向拥有该数据的目标节点发出请求报文，目标节点收到报文后对该请求进行应答，原节点收到应答报文后进行解析，提取有效数据，即完成了一次数据请求-推送过程。该策略思路直观，实现简单，适合计算复杂度较低的嵌入式设备，但未考虑带宽利用情况、网络节点工作状态，若所请求数据非当前任务所需，实时性要求不高，且目标节点有大量任务待处理时，会给目标节点增加更多负担，消息推送过程缺乏灵活性。

上述做法存在着一定的问题。若有多个用户请求某一个用户或者某一种类型的消息报文，则出现网络拥塞导致推送效率降低的概率较大。而在主动推送数据策略中，主要存在两种角色的节点：有数据需求的需求方节点和拥有数据的服务方节点。该策略将网络节点的传感器感知到物理环境或自身状态的改变定义为“事件”，事件发生会触发服务方节点将数据（环境或自身状态变化生成的信息）主动推送至有此数据需求的需求方节点。无论需求方节点此时需不需要此数据，均接收并存储。这种推送策略仍然不够灵活，一个比较显著的缺陷是若网络环境或服务方节点自身状态发生较频繁波动，将会产生大量主动推送的数据，占用较多的网络带宽，同时服务端节点侧发起大量数据推送过程，而需求方节点接收到数据后进行存储或者对原有数据的替换，加重了工作负载。若需求方节点暂时不需要此数据，数据的存储也可能导致大量存储空间被占用。这种不考虑需求方节点需求的推送策略极有可能导致需求方节点CPU计算资源和网络带宽资源的浪费，因此，设计一种新型的兼顾节点数据需求和网络带宽利用效率的消息预取及推送策略是非常有必要的。

基于统计信息的消息预取及推送策略以传感装置等低功耗低复杂度、功能较单一的设备为应用对象。该策略与主动推送数据策略的出发点不同。主动推送策略基于拥有数据的节点的视角，一旦感知到数据更新便会触发这类节点发起主动推送数据的过程。而基于统计信息的消息预取及推送策略则有所不同，对于网络中的某个具体节点而言，基于数据生产者的角度，节点通过自身所配置的传感器感知环境数据，是数据的提供者，可以基于统计信息将数据推送至其它节点；基于数据消费者的角度，节点执行动作或者做出决策任务还依赖从其他节点获取到的数据，可以基于统计信息进行提前预取。本发明中设计了根据消息预取及推送的历史记录有针对性地向网络中的节点预取或推送数据的策略。目标是让各个节点提前获取到接下来的计算任务中可能会用到的数据、指令，存储在本地设备上，等到使用时无需发出请求便可即时使用，有助于提高节点工作效率，降低任务处理时延。若接下来使用的数据未被预取的数据命中，则采用传统的基于用户请求的方式发出请求报文，从服务方节点获取数据。

物联网节点行为数据在系统中最简单的存在形式就是日志。系统在运行过程中都产生大量原始日志，每一条记录表示一次用户行为和对应的服务。比如光照强度、温度、湿度、节点工作状态、节点移动速度等，每一次服务请求和应答都会生成一个展示日志，其中记录了查询和返回结果。不同的节点对不同的业务信息请求频率和响应频率是不同的，必然有高频率的请求响应和低频次的请求响应，即不同节点间的联系紧密程度是不同的，如果各节点能够对自身的请求和被请求记录进行阶段性记录和分析，就能够发现不同节点之间的行为联系。据此可建立网络节点关联度列表，通过分析过去一段时间内网络各节点的历史消息推送记录，计算节点之间的关联度列表。当节点向网络中某服务方节点发出数据请求后，通过分析网络服务方节点关联度，也会向网络中与该服务方节点关联度较高的其它服务方节点发出数据请求。同理，当节点收到网络中某需求方节点发来的数据请求后，通过分析网络需求方节点关联度，向从而选择相似度高的消息请求，预取出来，实现消息的预取推送。

实现基于统计信息的消息预取及推送策略的第一步是建立网络需求方节点关联度列表和网络服务方节点关联度列表。这两个关联度列表以时间长度T为周期进行更新，列表中数值的计算任务由网络中的节点按照节点序号轮流执行。第n个周期的节点关联度列表由m号节点计算完成，则n+1个周期的节点关联度列表由n+1号节点计算完成。在第n个周期开始时，网络各节点将与自身进行数据预取和推送的节点标号及交互次数以报文的形式发送给m号节点。m号节点计算网络需求方节点关联度列表和网络服务方节点关联度列表，计算完成后将列表中与各节点相关的一部分计算结果分发给的对应的节点（每个节点会收到特定的一行和一列）。



若网络中有N个节点，则可统计网络节点间消息推送次数统计表如表1所示。表中推送节点指的是拥有数据的服务方节点，即数据的生产者；预取节点指的是有潜在数据请求的需求方节点，即数据的消费者。表中各统计值均以特定周期为单位，*Cxy*指的是某一个特定周期内节点*x*向节点*y*推送数据的次数。以节点1和节点2为例，计算二者同为需求方节点是的关联度，需要考虑网络其它节点(Node 3 - *N*)对节点1的数据推送记录与对节点2 的数据推送记录的相似程度，即*C*31, *C*41,…,*CN*1与*C*32, *C*42,…,*CN*2两个序列的相似程度。利用余弦相似度公式求出*D*12为



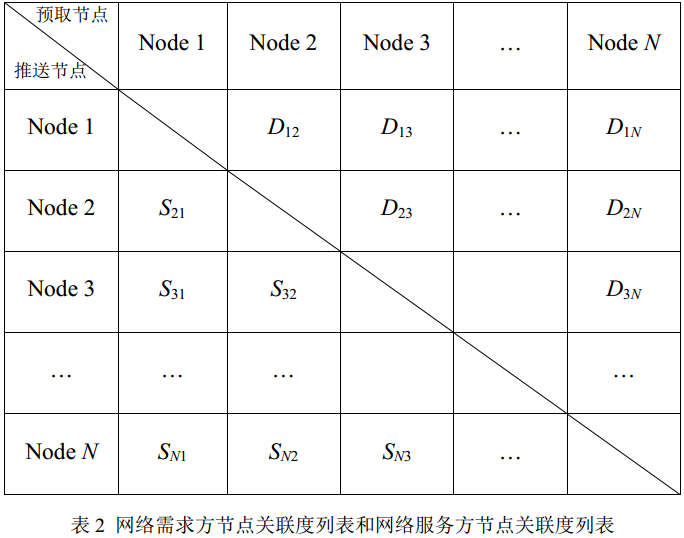
一般地，对可计算节点*x*和节点*y*需求关联度为



同理，计算节点*x*和节点*y*的服务关联度时，应考虑二者向网络其它节点推送数据记录的相似程度，可计算节点*x*和节点*y*服务关联度为



由上述需求关联度和服务关联度的计算公式计算得到网络需求方节点关联度列表和网络服务方节点关联度列表如表2所示，以对角线为分割线，右上侧的三角部分为各节点之间的需求关联度，左下侧的三角部分为各节点的服务关联度。该表中数值的有效期为周期T，每隔T时间更新一次，表格数值由下一个节点重新计算。



计算得到表2所示网络需求方节点关联度列表和网络服务方节点关联度列表后，完成该计算任务的节点将节点*t*所需的计算结果表格第*t*行以及第*t*列发送给节点*t*，从而使每个节点均获取到网络中其它节点数据的需求关联度和推送关联度。进行数据的预取时使用服务关联度，当节点向网络中某服务方节点发出数据请求后，也会向网络中与节点的服务关联度最高的节点发出数据请求，得到请求的数据后会向网络中与节点的服务关联度次高的其他节点发出数据请求，直至达到一定的次数后停止。同理，当节点收到网络中某需求方节点发来的数据请求后，不仅向该需求方节点推送数据，也会将同样的数据推送给与该需求方节点需求关联度最高的其他节点推送同样的数据报文，之后继续向与该需求方节点需求关联度次高的其他节点推送同样的数据报文，直至达到一定的次数后停止。需要注意的是，预取和推送的次数并不是无限制的。因为计算所得到的关联度衡量的是节点之间的推送或者预取行为的相关性，若关联度数值过小，说明两个节点在推送或者预取行为上没什么关联，不需要执行预取或推送操作。因此需求关联度和服务关联度需设置一定的阈值*D*，对于低于此值的节点不进行预取或者推送数据操作。此外，在网络节点趋同性强，大部分节点彼此之间关联度较高，若节点需要针对阈值以上的所有节点都进行预取或推送数据操作会占用大量带宽资源，只对关联度最高的一定比例的节点（例如*p*=20%）执行对应操作即可。实际应用时，节点对其他节点执行预取或推送数据的操作时须同时满足阈值*D*及比例*p*的限制条件。

总结以上策略流程。首先根据消息预取及推送的历史记录计算节点间的需求关联度和服务关联度。当节点向网络中某服务方节点发出数据请求后，通过分析网络服务方节点关联度，也会向网络中与该服务方节点关联度较高的其它服务方节点发出数据请求。同理，当节点收到网络中某需求方节点发来的数据请求后，通过分析网络需求方节点关联度，向从而选择相似度高的消息请求，预取出来，让网络节点可以将一部分数据和指令提前获取，当后续任务执行过程中产生真正的数据或指令需求时既可直接使用，降低任务处理耗时，提高工作效率。

1. **典型应用案例**

图4.1为协作物联网技术的一个实际应用案例。

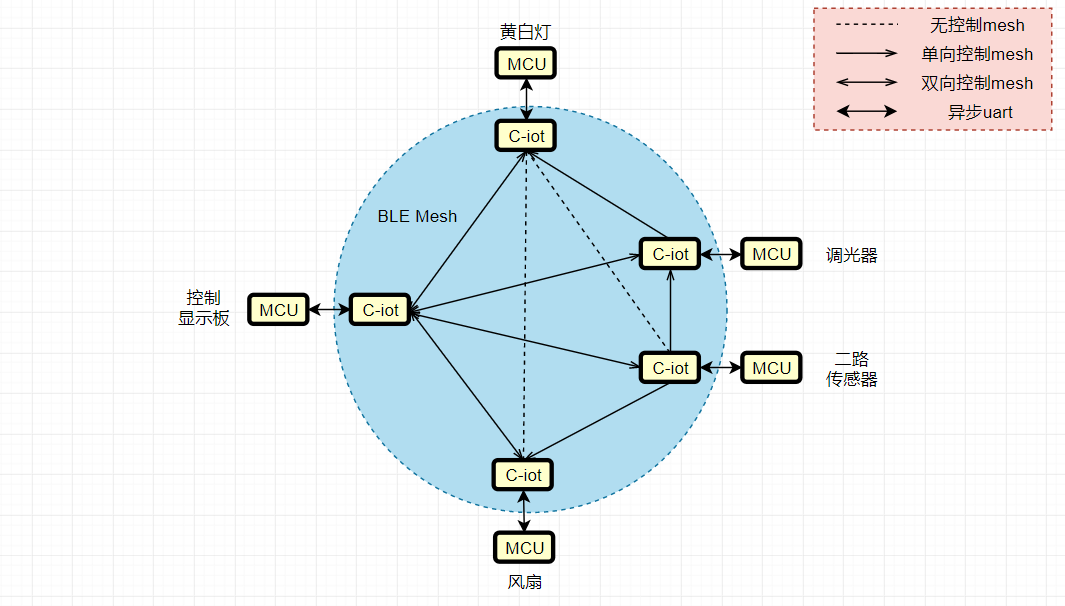


图4.1 协作物联网技术典型应用案例

该网络包括黄白灯、调光器、二路传感器、风扇、控制显示板等5个节点。在各节点内部，设备厂商在基本功能模组上嵌入C-iot协作物联网蓝牙模组，基本功能模组与C-iot蓝牙模组间以异步串口Uart方式在设备内部进行通信。各节点内部的C-iot蓝牙模组将各节点组成无线 Mesh网络。当节点申请加入网络时，发送广播报文，收到广播包后，首先验证密钥，确认无误后解析报文。按照前述节点相互识别与信息交换策略完成收-发-收的过程，确认加入网络。其它节点更新网络节点状态信息表。各节点周期广播自身的状态信息，并接收来自其他节点的广播包，对网络全部节点的状态信息进行动态更新。若在规定的时间周期阈值内没收到某节点的广播包则视作该节点脱离网络，其它节点检测到这一事件后将该节点信息删除。

节点每40ms发送一次广播报文，包含节点信息和mesh字段。其中节点信息包括设备类别、状态、地址标识、消息编号、生存周期以及密钥信息，mesh字段根据节点及网络状态按照SIG发布的Mesh Model和Mesh Profile协议配置即可。而扫描报文包括src/dest/info/msg/ttl等字段，分别表示源地址、目的地址、报文信息、消息编号和生存周期（转发次数），报文信息info字段进一步划分为cmd/len/INFO/crc等字段，其中cmd为指令，指令码从前述标准化命令编码mesh类网络通信指令码、basic\_func类设备功能指令码、device类设备自身状态指令码三类中选取，len字段表示报文长度，INFO字段为配合指令的操作数（设备执行某些指令需要cmd操作码和INFO操作数，全0表示无具体操作数），crc为校验字段。当网络所处边境变化较快时，报文通信量急剧增加，容易出现丢包现象，可根据前述消息预取和分级缓存策略进行冲突避免。