

[2019 年 10 月 24 日]



WAVEMESH 2.x

无线移动自组网协议综述

Email: eric8136@gmail.com | Tel: 13366662606 | QQ: 20206070

目录

WaveMesh 无线移动自组网协议综述	4
WaveMesh 协议族简介	7
WaveMesh-PRO 专业版	7
WaveMesh-STD 标准版	8
WaveMesh-AMR 抄表版	9
WaveMesh-BAS 基础版	9
WaveMesh-TNY 精简版	9
WaveMesh-STR 入门版	9
WaveMesh-LIT 轻量版	9
WaveMesh-MIN 迷你版	10
WaveMesh-SMP 简单版	11
WaveMesh-FRE 免费版	12
WaveMesh 其他版本	12
WaveMesh 不同本版功能对比	12
WaveMesh 休眠策略	14
异步休眠	14
异步休眠唤醒	14
同步休眠	15
混合休眠	16
自主休眠	17
被动休眠	17
强制休眠	18
WaveMesh 全网集抄	18
WaveMesh 报文解析	20
WaveMesh 串口握手	20
WaveMesh 碰撞算法	21
WaveMesh 路由协议	22
常见问题	25
不同固件类型是否互通	25

不同固件类型不同射频芯片是否互通	25
是否支持固件二次开发	25
是否支持 LoRa 调制方式.....	25
为什么 470MHz~510MHz 频段体验不如 433MHz 频段.....	25
是否支持 868MHz 和 915 MHz 频段	25
是否支持 2.4GHz 频段.....	25
对射频芯片有什么特殊要求	25
客户 ID 如何指定.....	25
固件如何烧写	25
实际部署网络规模和什么有关.....	26
模块复位或者重新上电为什么会重新收到上一次广播报文	26
模块为什么会收到少量重复报文.....	26
固件烧写数量如何授权	26
为什么硬件需要逐个出厂测试.....	26
硬件测试有什么方式.....	26
现场如何诊断网络.....	26
免责声明	28

WaveMesh 无线移动自组网协议综述

WaveMesh 无线移动自组网协议是为低成本、低功耗、移动设备设计的轻量级、分布式无线移动自组网协议。该协议定义了完备的链路层（MAC）和网络层（NWK）协议规范，但物理层可以采用多种无线信号调制方式，因此可以运行在多种射频芯片上。该协议采用市场通用的高性价比 ISM 频段射频芯片和 MCU 芯片，并且开放硬件设计，目的在于将无线模块的成本降至最低，降低各行业物联网应用的门槛。WaveMesh 协议经过了十多年的发展完善和市场验证，目前已经发展到第二代，相比上一代在功能和性能上有了质的飞越，可以满足绝大部分物联网的应用需求。WaveMesh 无线移动自组网协议的特点有：

- 分布式网络
 - 没有中心节点，所有节点都平等，网络扁平不分等级；
 - 所有节点各自计算自己的路由，不需要中心节点掌控全网路由；
- 移动节点
 - 所有节点都可以任意移动，在移动的同时保持数据流连续传输；
- 多种网络拓扑
 - 网状网络（在任何节点之间建立路由进行双向通信）；
 - 树形网络（在节点和网关之间建立路由进行双向通信）；
- 网络规模大
 - 协议栈资源开销与网络节点数量无关，网络规模没有理论上限；
- 网络覆盖范围大
 - 单一网络可以通过部署多个网关进行无限扩展；
 - 其中每个网关可以管辖最大 255 级中继路由的网络范围；
- 可靠数据传输
 - 单播采用 5 次握手方式可靠传输；
 - 可靠广播采用 4 次握手方式点对点可靠传输；
 - 网络不会主动丢弃数据报文，不休眠情况下可以实现 100%可靠传输；
- 协议开销极小
 - 数据传输不需要事先建立路由，网络不需要初始化，上电可立即工作；
 - 网络不需要心跳、信标报文，没有维护开销；
 - 协议握手开销固定而且很小（~10%）和无线传输速率无关；
- 网络健壮
 - 只需要相邻节点能够通信即可，不强调无线的传输距离，和单跳 LPWAN 窄带低速网络在设计理念上有本质的不同；
 - 路由协议可以动态寻找尽可能多的路由节点，路由的切换不需要额外时间，部分节点的离开对网络剩余部分没有影响；
- 吞吐量大
 - 采用私有智能碰撞算法和路由协议，可以在多路径、多信道并行发送数据；
 - 网络可以有多个网关，因此网络吞吐量可以远大于无线物理带宽；
 - 协议开销小，单物理信道有效传输速率可以达到物理带宽的 90%；

- 由于不要求无线传输的距离，可以采用比 LPWAN 窄带低速网络更高的无线传输速率，从而获得更大的网络吞吐量；
- 网络集抄
 - 采用全网集抄的方式代替逐点轮抄，能够在数秒钟之内采集成百上千点的数据（采集时间和数据量、物理带宽有关）；
 - 在非低功耗应用中，全网集抄保证 100%的成功率；
 - 在低功耗应用中，全网集抄会优先确保功耗，单次集抄接近 100%的成功率；
 - 网络集抄相对逐点轮抄具有更高的实时性、更低的功耗、更少的占用无线资源的特点；
- 智能碰撞算法
 - 不依赖于信号强度，没有远近效应，在强干扰下仍然有效；
 - 充分考虑网络拓扑的非均匀分布，优化网络瓶颈，确保网络最佳吞吐量；
- 精确路由
 - 在节点和节点之间、节点和网关之间均可以建立精确路由；
 - 会优先选择单播的方式按照路由进行精确转发；
 - 路由信息会在数据交换的过程中被时时刻刻更新和优化，不需要额外开销；
- 低功耗
 - 支持多种休眠方式：自主休眠、异步休眠、同步休眠和混合休眠等，可以满足绝大部分的低功耗应用；
 - 全新的异步休眠唤醒算法，节点监听时间片可以短至~1ms，按照 2 秒休眠时间片长度计算，节点异步休眠待机电流可以低至 5uA；
 - 分布式网络不需要初始化，节点上电可以立即进行数据传输，对比集中式网络需要长时间建立全网路由过程消耗的功耗就可以足够 WaveMesh 网络工作很长时间；
 - 采用全网集抄的方式，每次全网数据采集仅需要数秒即可完成。对比逐点轮抄全网数据采集所消耗的功耗是 WaveMesh 网络单次全网集抄的上百倍以上；
 - 采用比 LPWAN 网络更高的无线传输速率，用时更短，功耗更低；
 - 对于单播传输（单点控制）采用精确路由，只需要唤醒沿途中继节点，剩余节点不受影响；
- 安全性
 - 每个节点有不可被篡改的 6 字节客户 ID 和 6 字节全球唯一号 UUID；
 - 每个节点有可以配置的 3 字节网络 ID 和 128 个无线信道编号；
 - 采用物理层比特流动态加密，其密钥采用上述 3 种 ID 和无线信道编码信息的结合，无线信号被破解的概率极低，确保不同客户 ID、不同网络 ID 和不同信道之间的设备完全不能互通；
 - 严格限制不同厂商的设备采用不同的客户 ID，并且客户 ID 不能被修改，因此可以保障不同客户设备之间完全隔离，确保无线数据的安全；
 - 采用 CRC-32 替代上一代的 CRC-16 进行报文校验，极大降低了报文误码率；
- 开发成本低廉
 - 协议栈设计和实现完备，不需要应用层参与，直接向无线模块发送应用层数据报文即可，无需二次开发；

- 仅仅需要配置参数就可以完成开发工作，几分钟内就可以完成；
- 批量生产时可以将参数写入烧写器和固件一起烧入，不需要逐个配置参数；
- 生产成本低廉
 - 协议栈仅需要 1-2k RAM 和 8-32K Flash 空间，可以运行在 8 位 MCU 上；
 - 网络中所有的设备采用相同的软硬件设计，大大降低了网关设备的成本；
 - 硬件采用市场上性价比极高的通用射频芯片和 MCU，并且硬件开放给客户自己生产，将硬件成本降至最低；
 - 直接开放协议栈编译好的固件给客户使用，客户可以自行完成硬件测试、固件烧写，将生产成本降至最低；
- 维护成本低廉
 - 固件可以通过串口进行升级；
 - 固件也可以通过无线进行全网设备固件快速升级；

WaveMesh 协议族简介

WaveMesh 无线移动自组网协议栈为了满足不同的应用需要，同时减小协议栈的体积，进一步降低软硬件成本，因此设计了不同的协议版本，称之为 WaveMesh 协议族。

WaveMesh-PRO 专业版

WaveMesh-PRO 专业版网络拓扑结构是全连接的网状网络，所有的节点都是对等的，任意节点之间都可以建立路由进行双向数据交换。网络中任何节点都可以作为网关，也就没有网关的概念。低功耗支持自主休眠、异步休眠、被动休眠和强制休眠，但不支持同步休眠。数据传输可以由任意节点发起，支持广播和单播报文。其网络的拓扑示意图如下所示：

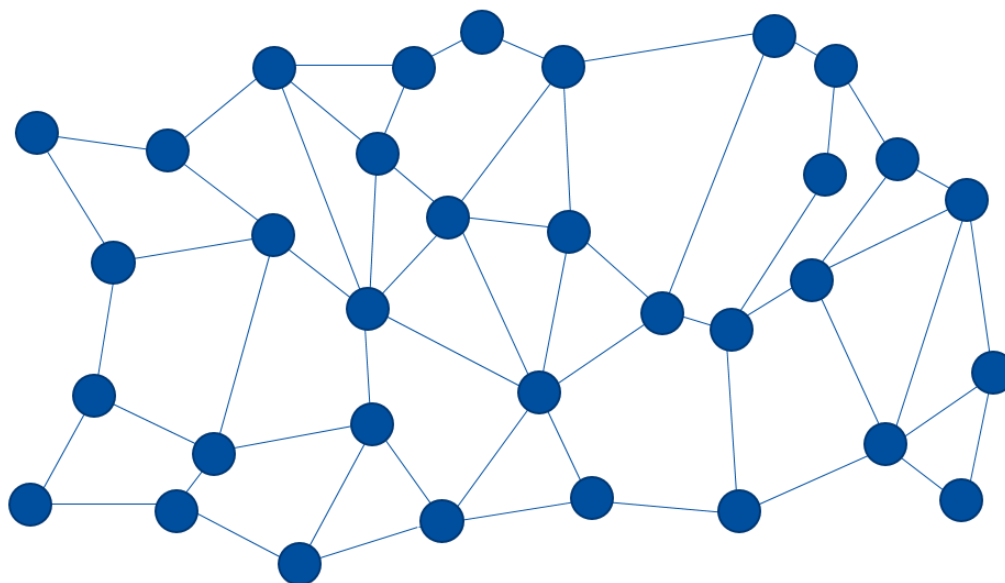


图 1 WaveMesh-PRO 网络拓扑示意图

说明：

1. 路由表项的数量跟 MCU 的内存大小有关，路由表的数量不代表网络容量上限；
2. 广播报文可以进行可靠传输和即时传输两种模式，可以通过模块参数进行选择；
3. 可以限定节点的最大路由级数（1- 255），从而控制网络中各节点的通信范围；
4. 可以在任意节点进行全网数据采集，具备树形网络的功能；
5. 采用异步休眠时，全网所有节点都可以休眠，数据发送前会进行休眠唤醒再发送；
6. 自主休眠的节点不参与路由，需要配合不休眠的路由节点组网；
7. 可以通过网络进行网络 RTC 授时，精度秒级；
8. 可以通过无线和串口进行固件升级；
9. 支持碰撞算法；

使能 WaveMesh-PRO 路由功能的条件：

1. 应用层数据帧中需要包含源地址和目的地址两个地址字段；

2. 需要配置参数使能协议栈对应用层数据帧进行解析，自动从应用层数据帧中获得源地址和目的地址；
3. 如果需要进行广播，则需要设置广播和多播通配符参数，使协议栈能够识别出广播地址和多播地址；

WaveMesh-STD 标准版

WaveMesh-STD 标准版网络拓扑结构是树形网络，由节点（NODEs）和网关（ROOTs）两种设备组成。节点可以和网关之间进行双向数据通信，节点之间可以进行数据中继转发和路由但是不能直接通信。低功耗支持自主休眠、异步休眠、同步休眠、混合休眠、被动休眠和强制休眠。数据传输可以由任意节点或者网关发起，支持广播和单播报文。由网关向节点发送方向的传输称之为下行，由节点向网关发送方向的传输称之为上行。其网络的拓扑示意图如下所示：

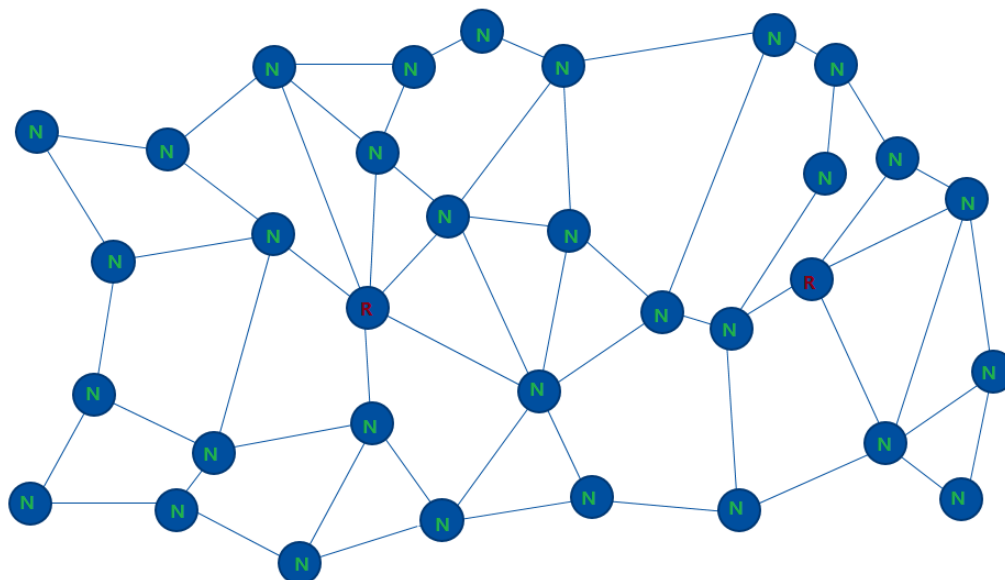


图 2 WaveMesh-STD 网络拓扑示意图

说明：

1. 路由表项的数量跟 MCU 的内存大小有关，路由表的数量不代表网络容量上限；
2. 下行可以采用可靠广播或者单播（精确路由）的方式进行传输；
3. 可以限定节点到网关的最大路由级数（1- 255），从而控制网关的通信范围；
4. 可以在网关进行全网数据采集；
5. 采用异步休眠时，全网所有节点都可以休眠，数据发送前会进行休眠唤醒再发送；
6. 自主休眠的节点不参与路由，需要配合不休眠的路由节点/网关组网；
7. 网关负责全网同步休眠的时间同步；
8. 可以通过网络进行网络 RTC 授时，精度秒级；
9. 可以通过无线和串口进行固件升级；
10. 支持碰撞算法；

使能 WaveMesh-STD 下行路由功能的条件：

1. 应用层数据帧中需要包含至少一个地址字段，上行报文需要提供源地址，下行报文需要提供目的地址；
2. 需要配置参数使能协议栈对应用层数据帧进行解析，自动从应用层数据帧中获得源地址和目的地址；
3. 如果需要进行广播，则需要设置广播和多播通配符参数，使协议栈能够识别出广播地址和多播地址；

WaveMesh-AMR 抄表版

WaveMesh-AMR 抄表版本是在 WaveMesh-STD 标准版本的基础之上，取消了部分功能，同时针对抄表的行业应做了优化，其它功能保持不变：

1. 取消了网关和节点同步休眠功能；
2. 取消了网关自动重发的功能；

WaveMesh-BAS 基础版

WaveMesh-BAS 基础版本是在 WaveMesh-STD 标准版本的基础之上，取消了部分功能，其它功能保持不变：

1. 取消了网关和节点同步休眠功能；
2. 取消了网关自动重发的功能；

WaveMesh-TNY 精简版

WaveMesh-TNY 精简版本是在 WaveMesh-STD 标准版本的基础之上，取消了部分功能，其它功能保持不变：

1. 取消了网关和节点异步休眠和异步唤醒的功能；
2. 取消了网关和节点同步休眠功能；
3. 取消了网关自动重发的功能；

WaveMesh-STR 入门版

WaveMesh-STR 入门版本是在 WaveMesh-STD 标准版本的基础之上，取消了部分功能，其它功能保持不变：

1. 取消了节点自主休眠的功能；
2. 取消了网关和节点异步休眠和异步唤醒的功能；
3. 取消了网关和节点同步休眠功能；
4. 取消了网关自动重发的功能；

WaveMesh-LIT 轻量版

WaveMesh-LIT 轻量版网络拓扑结构是全连接的网状网络，所有的节点都是对等的，底层采用可靠广播的方式进行双向数据交换，不支持路由和单播。节点之间可以进行数据报文的中继和转发。网络中任何节点都可以作为网关，也就没有网关的概念。低功耗支持异步休眠和被动休

眠，不支持强制休眠、自主休眠和同步休眠。数据传输可以由任意节点发起，其网络的拓扑示意图如下所示：

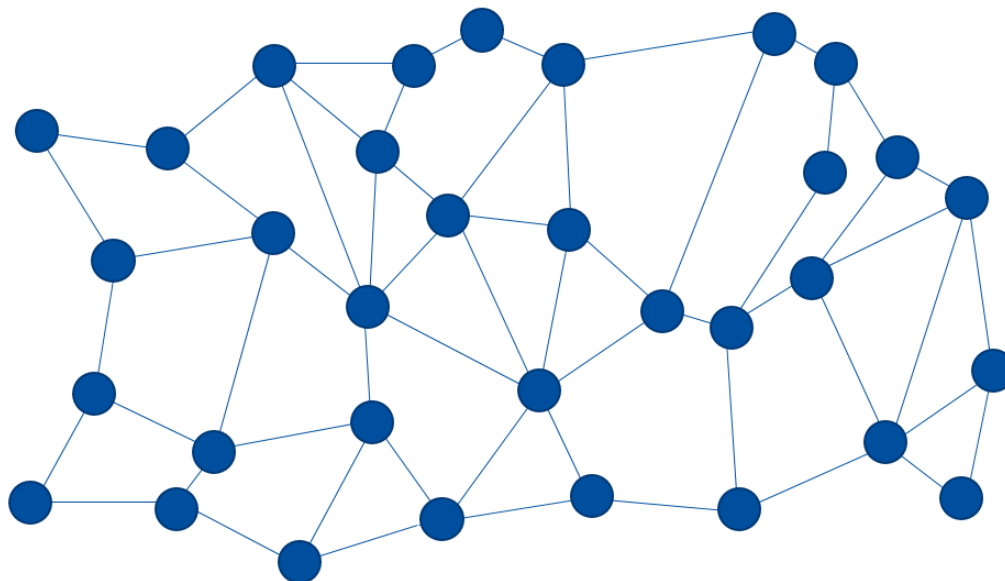


图 3 WaveMesh-LIT 网络拓扑示意图

说明：

1. 底层采用可靠广播的方式进行数据交换，没有路由表项；
2. 可以限定节点的最大广播中继级数（1- 255），控制网络中各节点的通信范围；
3. 不能进行单命令全网数据采集，只能逐点轮询；
4. 采用异步休眠时，全网所有节点都可以休眠，数据发送前会进行休眠唤醒再发送；
5. 不支持网络进行网络 RTC 授时；
6. 可以通过串口进行固件升级，但不支持无线固件升级；
7. 支持碰撞算法；

WaveMesh-MIN 迷你版

WaveMesh-MIN 迷你版网络拓扑结构是全连接的网状网络，所有的节点都是对等的，底层采用即时广播的方式进行双向数据交换，不支持路由和单播。节点之间可以进行数据报文的中继和转发。网络中任何节点都可以作为网关，也就没有网关的概念。低功耗支持被动休眠，不支持异步休眠、强制休眠、自主休眠和同步休眠。数据传输可以由任意节点发起，其网络的拓扑示意图如下所示：

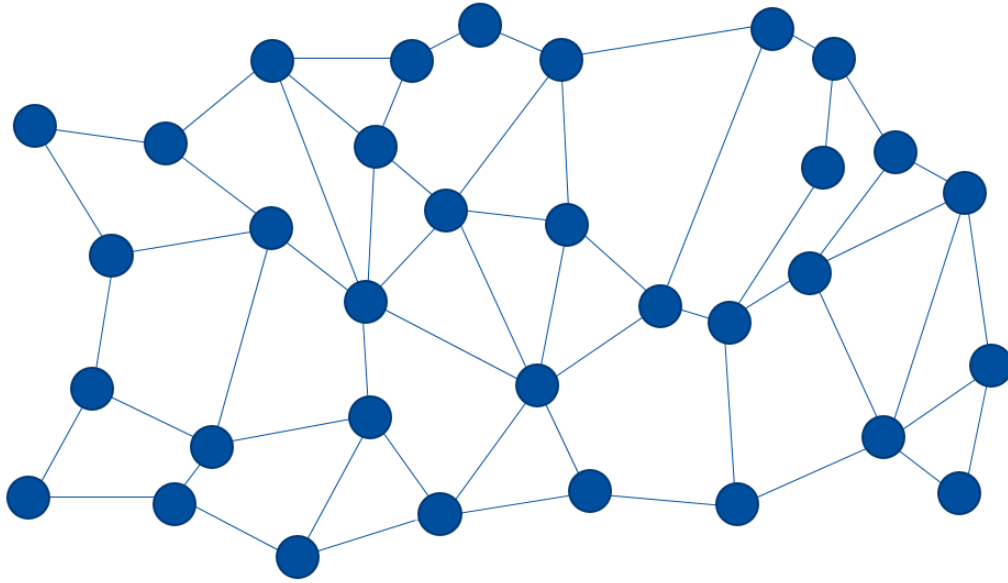


图 4 WaveMesh-MIN 网络拓扑示意图

说明：

1. 底层采用即时广播的方式进行数据交换，保证实时性但不保证可靠性，没有路由表项；
2. 不能限定节点的最大广播中继级数，不能控制网络中各节点的通信范围；
3. 不能进行全网数据采集；
4. 不支持网络进行网络 RTC 授时；
5. 可以通过串口进行固件升级，但不支持无线固件升级；
6. 不支持碰撞算法；

WaveMesh-SMP 简单版

WaveMesh-SMP 简单版网络拓扑结构是单跳网状网络，所有的节点都是对等的，底层采用即时广播的方式进行双向数据交换，不支持中继和路由。节点之间不可以进行数据报文的中继和转发。网络中任何节点都可以作为网关，也就没有网关的概念。低功耗支持被动休眠，不支持异步休眠、强制休眠、自主休眠和同步休眠。数据传输可以由任意节点发起，其网络的拓扑示意图如下所示：

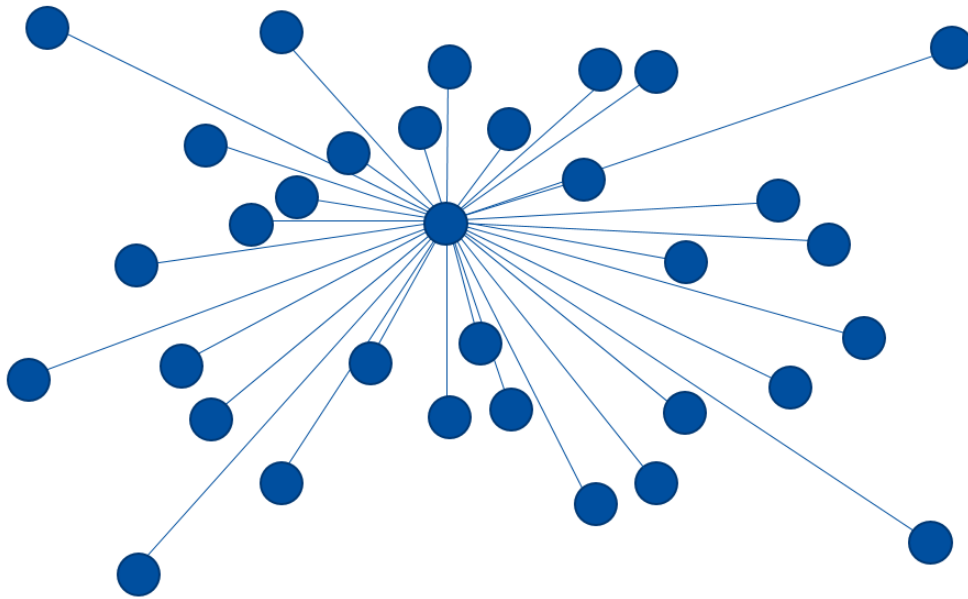


图 5 WaveMesh-SMP 网络拓扑示意图

说明：

1. 底层采用即时广播的方式进行数据交换，保证实时性但不保证可靠性，没有路由表项；
2. 不支持数据中继；
3. 不能进行单命令全网数据采集，只能逐点轮询；
4. 不支持网络进行网络 RTC 授时；
5. 可以通过串口进行固件升级，但不支持无线固件升级；
6. 不支持碰撞算法；

WaveMesh-FRE 免费版

WaveMesh-FRE 免费版网络是在 WaveMesh-SMP 简单版本的基础之上，限制最大的报文长度，其它功能保持不变。可以用于硬件测试和点对点的應用。

WaveMesh 其他版本

除了上述的模块固件版本之外，WaveMesh 还会为某些行业定制功能更全面的协议固件版本，这些版本不在此文档赘述，请参见别的文档。

WaveMesh 不同版本功能对比

为了方便对不同版本协议进行选择，WaveMesh 无线移动自组网协议栈不同版本之间功能对比如下表所示（数据部分仅对 2K RAM MCU 有效）：

功能\版本	PRO	STD	BAS	TNY	STR	LIT	MIN	SMP	FRE	AMR
网络拓扑	网状网络	树形网络	树形网络	树形网络	树形网络	可靠广播网络	即时广播网络	即时单跳网络	即时单跳网络	树形网络
精确路由	支持	支持	支持	支持	支持	不支持	不支持	不支持	不支持	支持
碰撞处理	支持	支持	支持	支持	支持	支持	不支持	不支持	不支持	支持
报文解析	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
地址过滤	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
自主休眠	支持	支持	支持	支持	不支持	不支持	不支持	不支持	不支持	支持
异步休眠	支持	支持	支持	不支持	不支持	不支持	不支持	不支持	不支持	支持
异步休眠唤醒	支持	支持	支持	不支持	不支持	不支持	不支持	不支持	不支持	支持
同步休眠	不支持	支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持
自动间隔重发	不支持	支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持
被动休眠	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
强制休眠	支持	支持	支持	支持	支持	不支持	不支持	不支持	不支持	支持
单播/路由	支持	支持	支持	支持	支持	不支持	不支持	不支持	不支持	支持
中继/转发	支持	支持	支持	支持	支持	支持	支持	不支持	不支持	支持
全网数据集抄	支持	支持	支持	支持	支持	不支持	不支持	不支持	不支持	支持
路由表项	512	1024	1024	1024	1024	0	0	0	0	1024
可靠广播	支持	支持	支持	支持	支持	支持	不支持	不支持	不支持	支持
即时广播	支持	不支持	不支持	不支持	不支持	不支持	支持	支持	支持	不支持
串口固件升级	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
无线固件升级	支持	支持	支持	支持	支持	不支持	不支持	不支持	不支持	支持
最大跳数	255	255	255	255	255	255	不限	1	1	255
限制网络范围	支持	支持	支持	支持	支持	支持	不支持	不支持	不支持	支持
最大报文长度	231字节	174字节	174字节	174字节	174字节	249字节	249字节	255字节	16字节	174字节
硬件测试	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持

图 6 WaveMesh 协议功能对比

WaveMesh 休眠策略

为了满足各种低功耗的应用场景，WaveMesh 协议栈提供了多种休眠策略：异步休眠、同步休眠、自主休眠、混合休眠、被动休眠和强制休眠。本节简单说明各种休眠的工作方式，方便对不同协议版本进行选择。

异步休眠

异步休眠模式下节点会在网络空闲后自动按照设定进入“休眠-监听-休眠-监听...”的循环过程，协议栈在休眠时间片结束时会打开射频监听网络中的无线信号，进入监听时间片。如果在监听时间片之内识别出特定的无线唤醒信号则会结束监听过程进入正常工作模式；否则会在监听时间片结束时进入下一个休眠的时间片。休眠时间片和监听时间片的长度可以根据实际需要设定。异步休眠待机过程的电流消耗典型示意图如下所示：

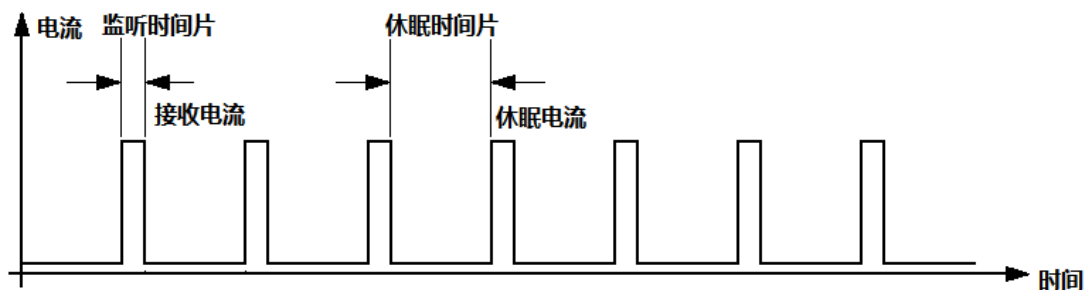


图 7 异步休眠待机电流示意图

以 WM7139-49N 模块为例，该模块无线速率 100kbps，其休眠时间片的长度最大为 4 秒，监听时间片的长度为 1.28ms 或者 2.56ms，接收电流 < 10mA，休眠电流 < 1.5uA。如果设置监听时间片为 1.28ms，则理论上待机电流为：~14.3uA@1 秒休眠时间片；~8uA@2 秒休眠时间片；~5.8uA@3 秒休眠时间片；~4.7uA@4 秒休眠时间片。

说明：

1. 所有节点和网关都可以进行异步休眠；
2. 无论是否使能休眠，全网所有节点均需要设置相同的异步休眠时间片参数；
3. 异步休眠节点之间通信，需要在交换正式数据之前进行无线唤醒；
4. 异步休眠待机后节点不需要发送任何报文，全网保持静默；
5. 异步休眠节点和非休眠节点可以进行数据通信，但需要非休眠节点设置相同的异步休眠参数以便能够唤醒异步休眠节点；
6. 异步休眠节点在被串口的数据报文唤醒后会立即结束休眠时间片进行突发数据传输；
7. 监听时间片的长度和无线的速率相关，无线速率越大监听时间片越短，反之越长；
8. 异步休眠数据传输功耗优先级大于可靠性，单次全网数据集抄不保证 100%可靠性；

异步休眠唤醒

对于异步休眠节点的无线唤醒通常采用前导的方式，然而这种唤醒方要求报文的前导长度大于休眠时间片的长度才能基本保证能够唤醒休眠节点。前导唤醒方式的报文耗时很长，被误唤醒

的概率很大，在唤醒时基本上没办法进行碰撞避免，没办法区分单播和多播，更没有办法进行休眠网络的集抄。因此 WaveMesh 协议并没有采用简单的前导唤醒方式。

由于 WaveMesh 协议的握手报文非常短并且加上优异的碰撞避免算法，上一代 WaveMesh 协议直接采用握手报文的方式进行唤醒，达到非常理想的唤醒效果。握手报文唤醒不需要报文持续整个休眠时间片，基本没有被误唤醒的可能，可以进行碰撞算法并且容易区分广播报文和单播报文，可以在唤醒过程中交换路由和广播信息，并且达到极高的唤醒成功率。但是握手报文唤醒方式在网络规模比较大的网络会因为持续的数据报文会导致反复唤醒休眠节点，休眠节点的监听时间片长度也会比较长。

在第二代 WaveMesh 协议中对握手报文唤醒方式进行了改进。简化握手报文过多信息，将报文的长度缩短到最短；充分利用不同射频芯片的特点，采用时间精度更高、更智能的碰撞算法。将唤醒过程独立出来，不再和数据交换过程同时进行，避免被反复唤醒的情况。经过改进的唤醒算法保留了第一代握手报文唤醒优点的同时，将唤醒过程所需的时间进行了严格的控制，有着确定的唤醒延时和网络节点密度以及网络的规模无关。第二代协议唤醒方式会优先确保功耗，而第一代协议会优先保证成功率，这是两代休眠唤醒算法的本质区别。由于采用更短的唤醒报文可以使监听时间片进一步缩短至~1ms 以内。对于单播仅需要唤醒下一跳中继节点，对于广播可以进行全网唤醒。异步唤醒和数据交换的过程如下图所示：

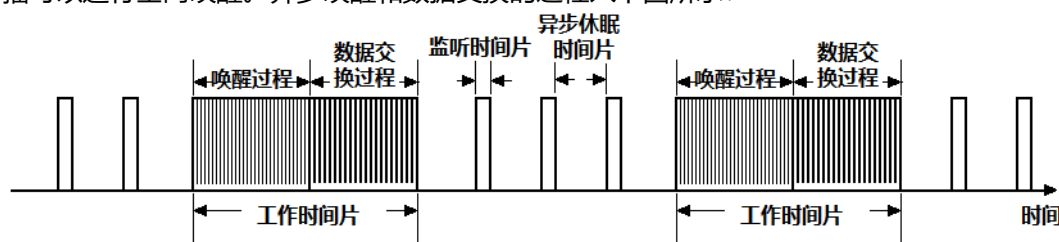


图 8 异步休眠唤醒过程示意图

同步休眠

同步休眠模式下，网络所有节点的休眠时间片是严格同步的，协议会尽可能将相邻节点之间的时间片误差降至最小，可以实现所有节点同时工作同时休眠。网络中需要指定某个网关负责时间片的同步，通过广播同步休眠报文实现，广播同步休眠报文携带本次同步休眠时间片的长度。因此同步休眠时间片长度不是固定值，可以根据需要进行随时调整。网络中若没有数据传输时可以立即进入下一个休眠时间片而不需要等待当前工作时间片结束。网关会根据静默超时判断工作时间片的结束时刻，因此工作时间片长度也随着网络数据量、节点数量的变化而变化。同步休眠模式下各节点时间片的误差很小，因此在工作时间片内进行数据传输不需要唤醒过程。同步模式下的时间片的使用如下图所示：

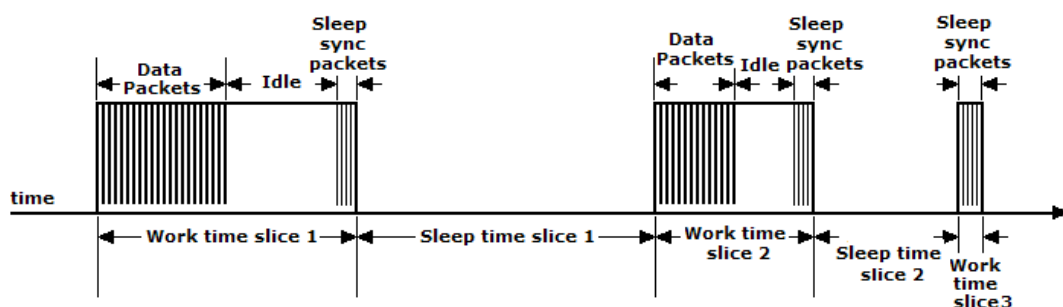


图 9 同步休眠时间片示意图

说明:

1. 为了降低硬件成本和功耗，模块休眠时采用的是 MCU 片内的低速 RC 振荡器，频偏比较大；
2. 协议软件会对低速 RC 振荡器进行校准，使其频偏在可控的范围内；
3. 协议充分考虑了校准后的 RC 振荡器的频偏带来的误差；
4. 综合考虑 MCU 的性能和低速 RC 振荡器的误差，目前允许的最大同步时间片的长度为~17 分钟；
5. 同步休眠参数仅仅需要对负责同步休眠的网关设置即可；
6. 同步休眠时间片无法提前结束，也无法进行突发数据传输；
7. 同步休眠数据传输功耗优先级大于可靠性，单次全网数据集抄不保证 100%可靠性；

混合休眠

单纯采用同步休眠会在节点收到干扰收不到正确的同步休眠报文而无法进入休眠的异常情况，这时节点就处于接收状态，就会产生比较大的功耗。为了增加同步休眠的健壮性，可以搭配异步休眠一起使用，称之为混合休眠模式。在同步时间片结束时刻，节点不会立即进入工作时间片，而是进入异步休眠。因此在下一个工作时间片开始之前需要进行异步休眠唤醒，在唤醒结束时才会进入工作时间片。采用混合休眠的节点在受到干扰接收不到同步休眠报文时，会进入异步休眠模式，这时待机功耗要远低于工作时间片，确保功耗优先的原则。采用混合休眠另一个好处是降低节点对工作片同步的精度要求，因为只要时间片的误差小于一个异步休眠时间片的长度均可以正常工作。混合模式时间片的使用如下图所示：

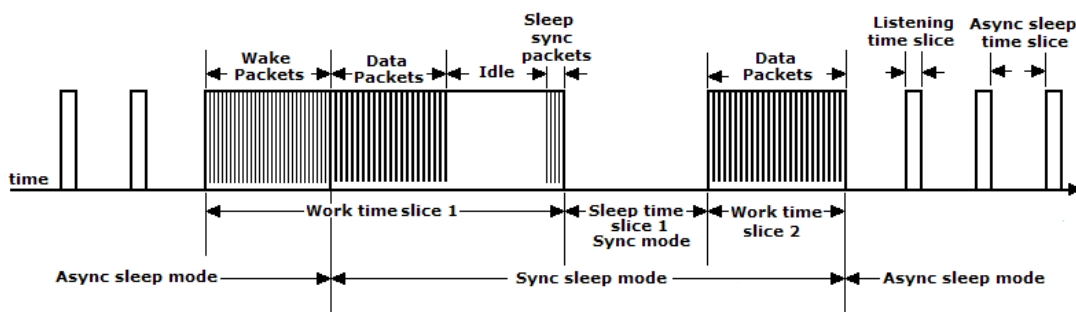


图 10 混合休眠时间片示意图

说明:

1. 异步休眠在混合休眠里起着异常情况处理的作用，仅仅是辅助作用；

2. 需要平衡正常工作和异常处理下的功耗平衡，推荐异步休眠时间片长度为 100ms~500ms；

自主休眠

对于不需要进行数据中继和路由转发的叶子节点（终端节点），可以仅仅在需要进行收发或者询问属于自己的数据报文时才会打开射频进行数据传输，在数据传输完毕可以立即进入休眠，称之为自主休眠。自主休眠节点可以和不休眠节点之间进行突发数据传输，但自主休眠节点之间无法进行数据交换。自主休眠节点和不休眠节点之间不存在绑定关系，因此自主休眠节点可以在网络中任意移动，甚至脱离网络。自主休眠的节点按照预设的休眠时间片进行休眠，在休眠时间片结束后会主动发送报文询问广播报文，以及探测不休眠节点的存在。因此，不休眠节点无需主动唤醒自主休眠节点。在接收到串口数据报文时，可以立即结束自主休眠时间片，进行突发数据传输。自主休眠时间片的使用如下图所示：

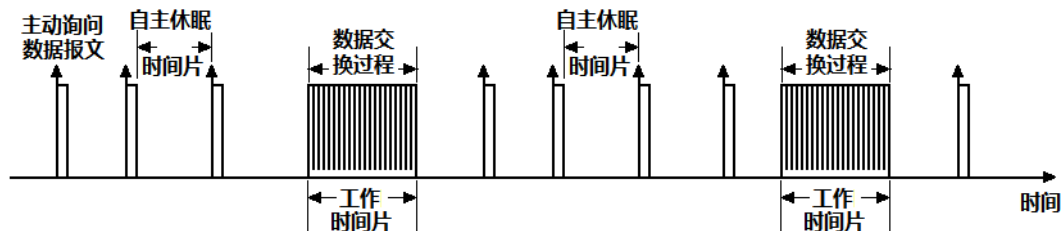


图 11 自主休眠时间片示意图

说明：

1. 自主休眠节点没有路由和数据中继转发的能力，网络需要有不休眠的路由/网关节点和自主休眠节点配合才能进行数据传输；
2. 不要求自主休眠节点的休眠时间片都相同，可以任意设定；
3. 自主休眠节点可以主动醒来探测网络和询问广播报文，其休眠时间片可以设置为 1 秒~4.5 分钟；
4. 如果不需要自主休眠节点主动醒来探测网络，可以设置自主休眠时间片为无穷大，这样自主休眠节点只有在收到串口数据报文时才打开射频进行数据传输，没有数据传输时会一直休眠状态；
5. 自主休眠节点可以获得最低的待机功耗，节点功耗通常比异步休眠低 1-3 个数量级，可以用于低功耗传感节点、人员定位卡等应用；
6. 在允许部署少量不休眠节点做路由的情况下，自主休眠是最佳选择；
7. 自主休眠节点在主动醒来探测网络时采用碰撞算法，在遇到网络拥塞或者无线报文错误时，可能会导致此次数据交换失败立刻重新进入休眠时间片，但数据报文不会丢失，并且在休眠时间片结束后会继续尝试；
8. 因此只要给足够多的时间，自主休眠单次全网数据集抄保证 100%可靠性；
9. 自主休眠的网络稳定性、抗干扰能力和网络规模都要优于异步休眠网络；

被动休眠

外部 MCU 可以通过串口发送 ATTS 指令，使节点立即进入休眠状态，其休眠策略而并非来自协议本身，称之为被动休眠。可以指定被动休眠的时间片长度 1 秒~4.5 分钟或者永久。在节点

休眠时间片结束之后或者接收到串口的数据报文后，协议栈会重新接管对硬件的控制权，恢复休眠之前的工作模式。这种休眠方式可以用来满足以下应用需求：

1. 硬件的休眠电流测试；
2. 设备在安装使用之前，长期存放和运输过程中；
3. 更加灵活多变的休眠策略，如可以只是在每天的某个时间段通过无线模块进行数据收发，其它时间可以休眠，这种休眠不规律的情况；

说明：

1. 被动休眠不需要设置任何参数，直接发送 AT 指令即可；

强制休眠

在网络个别节点硬件出现异常如不停复位等情况出现后，可能会和网络中相邻正常节点反复进行数据交换，导致正常节点无法休眠的极端情况出现。为了降低这种异常情况下的正常节点的功耗，可以使能强制休眠。强制休眠的休眠策略是：如果节点的工作时间片超出~8 分钟会强制节点休眠~17 分钟。这样做会使正常节点主动断开异常节点的数据请求，网络会处于比较长时间的共同静默状态。这段静默时间很可能会迫使异常节点放弃数据传输请求。强制休眠策略相当于飞机跑道最后的防护网，仅仅为了应对硬件的异常情况。

说明：

1. 需要确保正常情况下节点的工作时间片小于 8 分钟，才能使能强制休眠；
2. 强制休眠工作时间片超时时间为固定值，无法通过参数配置；

WaveMesh 全网集抄

无线自组网的大部分应用就是收集网络中海量节点数据，同时伴随着实时性和低功耗的要求，例如无线抄表、传感器采集等应用。如何快速实时获得全网所有节点的数据并且满足功耗的要求，WaveMesh 协议给出的答案就是全网集抄，这也是该协议的精髓。全网数据集抄只需要网关发送一条广播抄读报文，网关在数秒之内即可以获得全部节点的数据。对于不休眠的应用如电表抄表协议可以保证单次集抄 100%的成功率；对于自主休眠低功耗网络协议保证单次集抄 100%成功率；对于同步休眠和异步休眠低功耗网络，本着功耗优先的原则，协议可以提供单次集抄接近 100%成功率。

为什么集抄能在如此短的时间内获得全网数据？我们可以看一下集抄的工作过程：

1. 网关将广播抄读报文以可靠广播的方式文发送给其相邻节点或者第一级路由节点（可是很多个）；
2. 第一级路由节点在收到广播抄读报文后会继续中继转发给自己的相邻节点或者第二级路由节点（数量很可能比第一级多很多），同时将广播抄读报文通过串口发送给外设 MCU 以获取返回数据报文；

3. 第二级路由节点在收到广播抄读报文后会继续中继转发给自己的相邻节点或者第三级路由节点（数量很可能比第二级多很多）并且获得外设数据，此时第一级路由节点会将反馈的数据报文陆续发送给网关；
4. 第三级路由节点在收到广播抄读报文后会继续中继转发给自己的相邻节点或者第四级路由节点（数量很可能比第三级多很多）并且获得外设数据，此时第二级路由节点会将反馈的数据报文陆续发送给第一级路由节点，第一级路由节点会陆续中继转发第二级路由节点的数据报文给网关；
5. 依次，第四级、第五级... 直至采集完毕全网数据；

在网关看来，反馈的数据报文会从网络远处的各个节点源源不绝的汇聚过来，理论上单个网关的吞吐量可以达到物理层带宽的 80%到 90%（实际上需要根据射频芯片的能力，需要计算收发切换时间、校准时间和不同的报文长度，吞吐量可以达到物理层带宽的~70%）。整个网络的数据集抄时间可以根据总的数据量和吞吐量进行简单计算就可以得出。如果网络中有多个网关，可以使网络的吞吐量成倍的增加，这样整个网络的数据集抄时间可以成倍减少。

对全部节点处于异步休眠状态的网络进行全网集抄时，需要逐级唤醒所有节点，其中每级唤醒需要 1-2 倍异步休眠时间片长度。异步唤醒整个网络所需要时间跟网络最大跳数相关，跳数越多其唤醒时间就越久。然而全网异步唤醒过程和全网数据集抄过程可以并行进行，因此全网异步唤醒过程给集抄过程带来的额外时间开销并不多。按照通常情况如果每一级节点数量达到几十个以上，异步唤醒过程会比数据传输过程耗时要短，这样以来异步唤醒过程就不会额外增加时间开销。因此，通常情况下全网异步唤醒过程给整个集抄过程总时间仅仅增加 1-2 倍异步休眠时间片长度。鉴于异步休眠时间片为 0 - 4 秒，因此对于异步休眠的网络集抄用时仍然可以达到秒级。

说明：

1. 集抄的过程会自动建立网关到节点的双向路由，因此整个网络不需要事前建立路由，网络上电可以立即工作；
2. 如果节点需要上报的数据量大于协议支持的最大报文长度，则需要将数据打包成多个报文进行发送；
3. 集抄的过程各节点向网关发送的数据报文个数不限，如果需要发送多个报文则需要与无线模块进行串口握手，以确保在网络拥塞的情况下报文不被串口丢弃；
4. 对于成百上千点密度适中的网络来说，每节点均上报一包数据，射频速率采用 100kbps，全网集抄可以在数秒内完成；
5. 对于包含自主休眠的节点网络，全网集抄可以保证 100%可靠性，但是需要等待 1 到多个自主休眠时间片长度（时间片无限长除外）才能得到全部数据。这是因为休眠节点为了保证功耗优先，在拥塞和碰撞的时候很可能会放弃数据传输立即休眠，但会在休眠时间片结束再次尝试；

WaveMesh 报文解析

为了减小应用层开发的工作量甚至免除协议栈二次开发的必要，WaveMesh 协议栈可以直接接受客户的应用层的数据帧，并且协议栈的行为不需要应用层进行干预。在应用层看来，WaveMesh 无线模块按照透明传输的方式收发应用层的数据报文，在使用上和没有协议栈的 UART-RF 透明传输模块基本没有区别。但是，简单的透明传输应用层的数据报文无法保证数据帧的完整性，也无法从数据帧中获得地址信息也就无法进行精确路由。因此，WaveMesh 协议栈设计了非常灵活的应用层数据帧解析功能，该功能可以通过参数灵活配置，基本上可以适配何种数据帧格式。WaveMesh 数据帧解析过程如下所示：

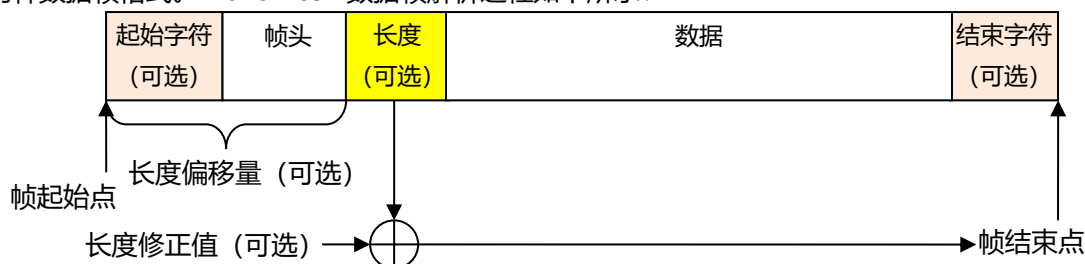


图 12 WaveMesh 数据帧解析

WaveMesh 的数据帧解析参数仅有 4 个参数而且都是可选的：帧起始字符、帧结束字符、长度偏移量、长度修正值。可以解析以下数据帧：

1. 不做任何解析，真正透明传输；
2. 解析按照固定字符开始、固定字符结束的数据帧；
3. 固定长度数据帧；
4. 帧中含有帧长度信息的数据帧；
5. 其中 2 可以和 3、4 项联合检测；

说明：

1. 数据帧解析可以过滤垃圾字节，特别是模块在使用串口休眠唤醒时容易产生垃圾数据；
2. 确认数据帧的完整性，特别是需要确认数据帧的开始位置，这是获取正确地址信息进行路由转发的基石；
3. 数据帧解析和串口握手报文搭配使用，确保串口数据不丢失，是实现网络可靠传输的关键；
4. AT 指令解析模式和应用层数据帧解析模式会同时进行，因此不需要主动切换这两种帧解析模式，但是 AT 指令识别优先于应用层数据帧；

WaveMesh 串口握手

WaveMesh 协议在不休眠情况下可以保证 100%数据可靠性。在休眠情况下会因为优先保证功耗，存在极小无线数据报文发送不成功的可能性。但是在无线发送失败的情况下也绝不会丢弃任何数据报文，等待网络恢复或者干扰消除后会继续发送之前未发送的报文。因此 WaveMesh 协议可以被认为是一种可靠传输网络协议。

在连续发送多个报文的应用中仍然会发现无线模块丢包的现象，其实数据包是被无线模块的串口丢弃而不是无线。无线模块的片内的数据缓存数量有限，在无线网络拥塞的情况下，模块缓存中的数据报文还没来得及通过无线发送，这时如果向模块串口持续发送新的报文会导致缓冲区满。在模块缓冲区满之后会临时关闭串口，这时再向模块串口发送数据报文就会被丢弃。只有等待数据报文发送成功或者失败之后，才会再次打开模块串口接收新的报文。

与无线模块进行简单的串口握手，就可以保证模块的串口数据报文的可靠性。模块提供参数可以设置 3 个串口握手报文：

1. 串口确认报文：在模块串口接收到完整的数据帧后会立即回会通过串口发送该报文通知外设 MCU 确认串口已经准确收到刚刚的数据帧；
2. 发送成功报文：在模块通过无线发送成功一包单播数据帧后（广播无效），会通过串口发送该报文通知外设 MCU 数据报文被无线成功发送；
3. 缓冲区空报文：在模块串口缓冲区由满到空时，会通过串口发送该报文通知外设 MCU 可以向模块发送新的数据报文；

说明：

1. 串口握手是可选项，在需要连续发送多个数据报文时或者实际应用中会发现丢包现象时需要使能该功能；
2. 串口握手各报文内容由客户自己定义，长度最大仅为 15 字节；
3. 串口握手各报文设置后就为固定内容，目前无法回应序列号等其它信息；

WaveMesh 碰撞算法

WaveMesh 协议底层是分布式异步网络，即使采用同步休眠的节点底层也是个纯异步的网络。并且该协议采用完全扁平化设计，多种类型的设备如路由中继节点和叶子节点会共享相同的物理信道，其数据传输和路由完全依赖碰撞避免算法，因此碰撞避免算法的效率和策略对该协议而且显得异常重要。WaveMesh 碰撞避免算法经过了很多年的改进，无论网络的疏密、节点类型的不同，新一代的算法目前的有效物理层带宽利用率可以达到 90%以上，基本上已经达到理想极限。

相对于上一代碰撞避免算法做了以下改进：

1. 信道空闲检测不再依赖信号强度，远处节点和近处节点具有几乎相同算法优先级，同时减小了 MCU 的开销，降低了对射频芯片的依赖程度；
2. 对非均匀分布的网络做了优化，对于网络瓶颈上的中继节点会有更高优先级，确保网络最佳吞吐量。以自主休眠网络为例，数量众多的叶子节点需要经过某个路由节点中继转发数据报文，这就要求担任路由的节点要求更高的发送优先级；以抄表应用为例，新的居民楼盖的越来越高，在同一栋楼里的表相对位置比较近、无线信号比较强，而和相邻楼或者单元里的表往往信号比较差，很可能楼和楼之间只有几个边缘节点担任路由和中继，这就要求碰撞避免算法需要感知路由关系，确保担任楼间路由的节点具有更高的发送优先级；

3. 算法能够时时刻刻感应和预测网络吞吐量的变化和同时进行传输的相邻节点数量的变化。以数据集抄为例，汇聚节点输出的数据流吞吐量从集抄的开始到结束几乎没有变化。而集抄的整个过程中，同时想要发送数据的邻居节点数量随着时间的推移在逐步下降，这就要求碰撞避免算法的效率不随着节点的数量变化而变化；

说明：

1. 新一代碰撞避免算法根据实际应用场景和资源开销综合设计，要求同时发送数报文的相邻节点的数量尽可能控制在 250 点以内；
2. 不过在实际使用中基本上不可能所有相邻节点会同时发送数据，因此对相邻节点的数量限制不是个硬性指标；
3. 对于非常密集的网络，相邻节点数量远远大于 250 的情况下，可以通过降低射频发射功率的方法减小相邻节点数量；
4. 对于生产出厂测试来说，往往节点数量非常多和密集，这就需要按照分批测试和逐个测试的原则，不能上千个节点放在一起测试；
5. 相邻节点的数量和网络节点容量是完全不同的概念，对相邻节点的数量限制不是对网络节点容量的限制；

WaveMesh 路由协议

无线移动自组网是多个具有路由功能的移动节点组成的多跳网络，数据的传输需要多个节点的协作才能完成，因此路由协议是网络实现至关重要的一部分。与传统有线网络相比，无线移动自组网有自己的特点，如分布式控制、动态变化的网络拓扑结构、无线传输带宽和节点能力有限、安全性差、路由生存时间短等。

理想的无线移动自组网路由协议应该具有以下特点：

1. 分布式路由算法；分布式算法更适合于无中心的分布式控制网络；
2. 自适应能力强；可适应快速变化的网络拓扑结构；
3. 无环路；无环路是任何路由协议的基本要求，可以避免路由错误和带宽浪费；
4. 路由计算与维护控制开销少；用最小的控制开销做到最完整、最强大的功能是所有路由协议共同努力的目标；
5. 适应于大规模网络；
6. 健壮性、可扩展性好；

无线移动自组网中从任何一个源节点到目的节点的路径通常会有多条，而且节点具有随机移动性，整个网络的拓扑结构经常变化。WaveMesh 协议采用私有按需轻量动态多径路由协议，该协议是针对硬件资源条件苛刻的移动自组网设计的，适用于移动速度快、拓扑结构变化快的无线网络。该路由协议可以最大限度减小路由建立和维护过程的开销，能够在多条路径并行进行数据报文的发送，可以感知网络拓扑结构的变化并对路由进行更新不需要进行洪泛，在不同路由之间无缝切换。

WaveMesh 路由协议特点有：

1. 每个节点维护自己的路由信息，没有中心节点负责全网的路由；
2. 路由不需要事先建立，如果之前没有路由会采用可靠广播的方式发送，在数据报文发送的过程中同时建立路由，没有路由建立延时；
3. 在建立路由之后，在数据通信的同时能够充分利用无线信号的冗余，时时刻刻进行路由的维护和更新，并且没有额外开销；
4. 不论多大的规模，完全没有路由回路，路由稳定性好；
5. 路由选择算法权衡了很多因素如距离矢量、信号能量、链路质量和电池电压等；
6. 支持移动节点，对网络拓扑结构的变化敏感，路由能够动态迅速达到最优；
7. 支持最大 255 级路由，网络规模大；
8. 可以限制最大的路由级数（1-255），从而可以控制数据发送的网络范围；

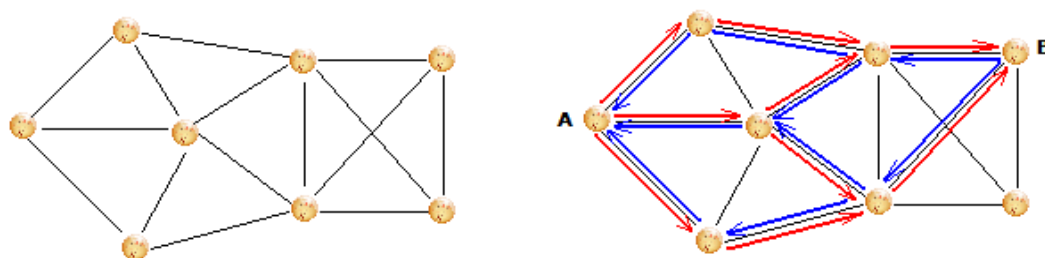


图 13 WaveMesh 多径路由示意图

WaveMesh 多径路由示意图如上所示，网络拓扑结构图如左侧所示，黑线代表节点间的链路。右侧为的 A 和 B 两点建立的数据链路，红线为从 A 到 B 点的路径，蓝线为逆向路径。路由的建立通过可靠广播的方式实现，多条路由间没有闭环回路、允许多条路径相交。每个节点都会选择尽可能多的节点作为自己的下一跳路由，数据报文可以在多条路径之间动态切换，能够并行传输。失效路由检测、新路由发现、网络拓扑结构的变化通过监听相邻节点间的握手报文来感知，不需要进行洪泛也不需要额外的开销。包括源节点在内的所有节点仅需要寻找自己的下一跳中继节点，而不需要确定整条路径，因此该路由协议开销很小，适合拓扑结构快速变化的移动网络，能够迅速发现即时最佳路由，支持 255 级路由的超大规模网络。

WaveMesh 路由协议会综合多种选择算法进行路由的筛选，包括距离矢量、信号质量和节点剩余电量。距离矢量算法根据目的地的远近来决定的路径，每个节点都会维护一张矢量表，表中列出了当前已知的到每个目标的最佳距离。节点可以根据这张矢量表，选择比自己更接近目的地的节点作为转发路由。根据距离矢量算法可以找到两个节点间的最近路径，但不一定是最佳路径。

不同于有线的网络，对于无线移动自组网来说无线信号容易受外界干扰的影响，造成数据链路生存时间短、稳定性差的特点。路由协议必须能够正确选择信号质量好、链路稳定的路径才能保证网络的稳定性、实时性、可靠性和抗干扰能力。WaveMesh 路由协议能够迅速探测多条路由的即时链路质量，能在极短时间内选择出最佳链路质量的路径做路由，并且在必要时可以选

择次最近路径作为路由。WaveMesh 链路状态算法路由选择示意图如下所示，A 可以通过 B 中继到 C 既 A-B-C 但该条路径受到干扰为不稳定链路，同时 A 到 C 有另外一条链路质量好的路径 A-D-E-C。选择 A-B-C 这条路径虽然距离更近，但是由于链路不稳定性报文的接收成功率很低，会大大增加报文的重发概率耗费大量时间。而如果选择 A-D-E-C，虽然距离会远一些，但是能保证报文传输的可靠性和实时性。

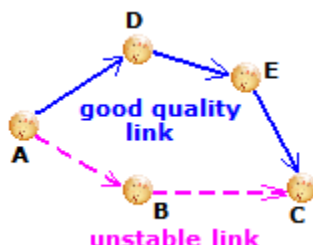


图 14 WaveMesh 链路状态算法路由选择示意图

另外对于低功耗无线移动自组网来说，路由选择需要充分考虑节点电池的电量，应尽可能避开电池电量低的节点进行路由。在距离矢量和信号质量都相同的条件下，WaveMesh 路由协议会自动选择剩余电量相对大的节点做路由。WaveMesh 节点剩余电量路由选择算法示意图如下所示，A 可以通过 B 中继到 C 既 A-B-C 但 B 节点的剩余电量低，同时 A 到 C 有另外一条稍远的路径 A-D-E-C，路径中的节点剩余电量较高。选择 A-B-C 这条路径虽然距离更近，但是会很快耗尽 B 节点的电量。如果选择 A-D-E-C，虽然距离会远一些，但是增加整个网络的使用寿命减小系统的维护费用。

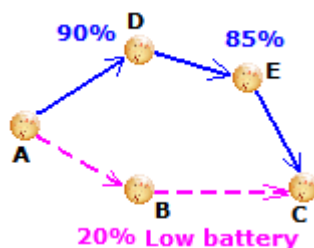


图 15 WaveMesh 节点剩余电量路由选择示意图

常见问题

不同固件类型是否互通

不同固件类型的底层握手报文定义不同，因此无法互通。即使固件的参数完全相同，也不能通信。

不同固件类型不同射频芯片是否互通

不同厂家的射频芯片性能不同甚至调制方式不同。协议栈的设计原则是性能至上，为了获得最佳的性能，并没有为了不同的射频芯片做折中设计，因此不保证不同射频芯片之间的互通性。

是否支持固件二次开发

不支持。

是否支持 LoRa 调制方式

支持，采用 SX1278 LoRa 射频芯片固件已经发布。

为什么 470MHz~510MHz 频段体验不如 433MHz 频段

随着很多城市陆续开通大功率地面波数字电视信号，其发射频段包含 470MHz~510MHz，会对低功率无线自组网信号造成很大的影响，因此体验会不如 433MHz 频段。推荐首选使用 433MHz 频段。

是否支持 868MHz 和 915 MHz 频段

支持，A7139-86N 模块可以支持上述频段。

是否支持 2.4GHz 频段

支持。

对射频芯片有什么特殊要求

WaveMesh 协议栈主要采用射频的直接模式，因此对于支持直接模式会优先考虑。除此之外，对无线芯片没额外特殊要求。

客户 ID 如何指定

客户 ID 为 6 字节，公司客户可以选择喜欢的 ID 编码进行登记，只要别的公司没有使用过即可。

固件如何烧写

通过 WaveMesh 专用离线脱机烧写器进行烧写。

实际部署网络规模 and 什么有关

虽然 WaveMesh 网络规模几乎没有理论限制，但是实际中需要根据网络的吞吐量、功耗和实时性要求来设计网络的规模。

模块复位或者重新上电为什么会重新收到上一次广播报文

这是 WaveMesh 可靠广播的特点，节点不会因为错过广播的时间点而错过广播报文，在系统设计时需要考虑节点刚上电、复位时接收到的广播报文。

模块为什么会收到少量重复报文

WaveMesh 单播采用 5 次握手的方式进行可靠传输，如果第 5 次握手被干扰产生错误就会造成发送端和接收端同时拥有该重复报文。WaveMesh 协议会纪录最近几次通过无线接收到的单播报文的校验值并进行重包过滤，这种方式可以过滤掉绝大部分的单播重包。但是，由于 WaveMesh 路由协议的设计，发送失败的单播报文被发送端再次发送时会采用别的路径发送，这样重包过滤就无法奏效，就造成会收到极少量的单播重复报文。

固件烧写数量如何授权

通过授权文件网络发送，不需要快递烧写器硬件实物。

为什么硬件需要逐个出厂测试

分布式自组网中某个节点的硬件异常可能会导致周边正常节点甚至整个网络工作不正常。相对集中式自组网，分布式自组网对硬件有更严格的要求，不允许硬件异常的节点参与组网。因此，需要对每个出厂的硬件设备进行逐个出厂测试。

硬件测试有什么方式

首先可以烧写免费版本的 WaveMesh-FRE 固件，通过 AT 指令可以通过频谱仪、信号源、电源等设备测试无线的发射功率、频偏、接收灵敏度、发射电流、接收电流、休眠电流等指标。还可以在烧写正式版本固件之后，可以通过测试吞吐量的工具，测试无线模块的真实吞吐量，判断无线模块的优劣。

现场如何诊断网络

可以使用 ATRI 指令来获取全网所有节点的无线信息：信号强度、电压、路由、温度、复位原因等信息，用来直观了解各节点的工作状况，甚至是全网节点拓扑结构。

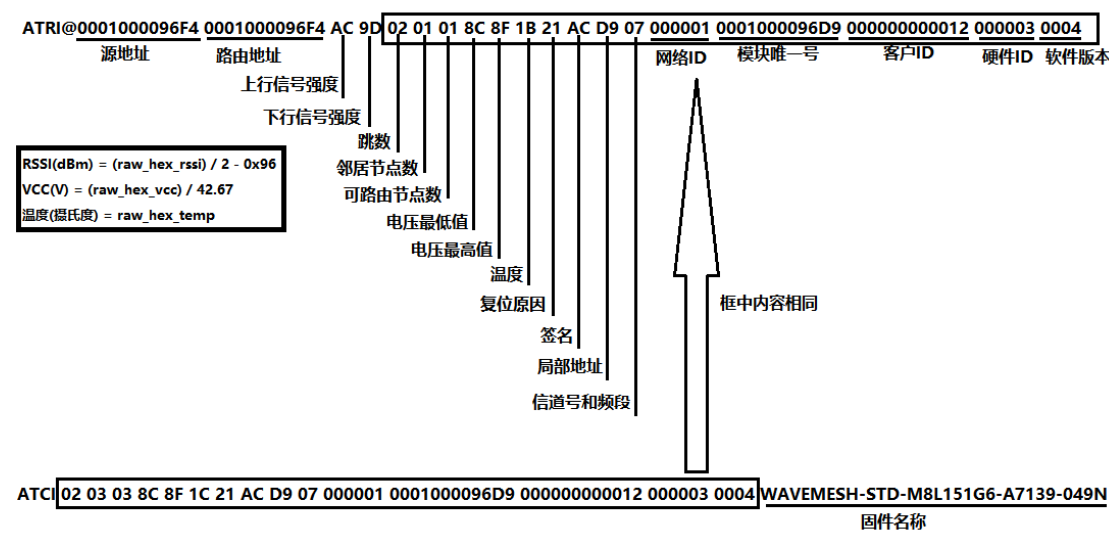


图 16 ATRI/ATCI 报文解析

免责声明

1. WaveMesh 协议为私有协议，只能用在免费 ISM 频段中使用，需要符合当地的法律法规。对于非法使用的情况，本协议开发人员免责。
2. WaveMesh 协议编译后的固件均为公开下载的方式，客户需要自行负责烧写的固件类型和版本。对于烧写错误固件版本的情况，本协议开发人员免责。
3. 客户需要充分测试 WaveMesh 固件的功能和性能之后才能做商用，本协议开发人员会积极解决客户发现影响其使用的固件功能性问题。但对于已经商用的固件缺陷问题并且造成的损失，本协议开发人员免责。
4. 本协议开发人员会对客户提供免费基本的远程技术支持和问题解答，如果需要更优质的服务需要付费。
5. 如果需要本协议开发人员去异地（北京市城区之外）现场解决问题的情况，需要客户付费并负责车旅、食宿费用，保证开发人员人身安全。