

WaveMesh 远程网络划分

2014. 08

现场零配置，将维护成本降低至最低

WaveMesh Working Group

内容

为什么要进行远程网络划分	1
如何实现远程网络划分	2
如何实现跨网互通	3
相关参数	4
相关 AT 指令	8
建议设置过程	19
实例说明	19
常见问题解答	26

远程网络划分可以实现安装零配置、设备维护无需去现场，极大节省设备安装和维护成本，避免人为操作错误。

为什么要进行远程网络划分

WaveMesh 网络可以支持几万点甚至更大规模的网络，但有时候实际应用中并不允许采用这么大规模的网络，其原因有以下几种可能：

1. 节点虽然在地理位置上有交叉，但应该属于不同的网络。如：对于电表来说，在不同的台区下的表应该属于不同的网络；对于燃气表、水表、热表来说，在不同供应管道下的表应该属于不同的网络。如果同一个小区同时采用多个的供应商，这样就绝对不可能让不同的供应商的仪表组成一个混杂的网络。
2. 需要控制网络的规模，便于后台管理。如果不对 WaveMesh 网络加以控制，可能会形成超大规模的网络甚至没有边界，这样网络中往往会有很多个网关并且数据上行报文会自动选择网关，这样后台处理就产生不确定性。

除此之外，实际应用还需要考虑一下因素：

1. 功耗和安全方面的考虑。网络的规模很大以后，网络中报文转发的次数变多，会导致远处节点的延时很大，路由节点的功耗上升。另外，如果同一张网络中存在不同的网关却隶属于不同的管理后台，就很可能存在安全隐患。
2. 维护成本的考虑。目前，对无线模块的参数修改或者网络划分基本无一例外地采用在安装时现场采用红外、点对点等其它手段对指定的节点进行网络 ID 等参数的修改。这种需要在现场才能对节点参数进行手工修改的方式，给网络的维护带来了诸多不便，现场必需保证一次性参数设置正确。如果后续有新节点加入，新老设备替换，也必需事先知道需要设置节点参数，进行现成设置。这种设计的维护成本实现实在太高。允许远程进行网络划分可以实现现场安装时不需要对节点参数进行任何设置，彻底避免出错的可能。另外，可以实现在后续的使用中对网络进行重新划分，远程重新

划分网络使得费时费力的人工维护，变成简单的鼠标点击。远程划分网络的引入会大大降低系统的维护成本。

远程网络划分并非“先有鸡还是先有蛋”的悖论，可以通过多种方式去实现。

如何实现远程网络划分

远程网络划分的目的是为了将庞大的网络划分成多个较小的网络，在划分后的小网络之间正常情况下是不允许再进行无线通信了。这时如果需要对网络进行再次远程划分，只能通过无线来实现，然而之间既然已经不允许不同网络之间进行无线通信了，那又如何实现呢？这个“先有鸡还是先有蛋”的问题“看似是无解的。其实，这种类似的问题在无线通讯中比比皆是，还是不难找到解决方法的。

WaveMesh 协议解决该问题有两种方式：

1. 节点主动方式

- a. 每个网关上电后随机选择网络 ID，如果探测到网络中存在不止一个网关设备，则网关会自动选择别的网络 ID。也可以通过远程指定每个网关设备有独立的网络 ID。确保所有网关的网络 ID 都不同。
- b. 向网关设备下达所管辖的所有节点设备的地址列表（网络的划分和再划分的工作，简化到向网关设备下达节点设备的地址列表即可）。
- c. 节点设备在上电或者认为断网后主动搜索周围存在的网络（不同网络 ID），如果扫描到网络存在则向网关发送注册报文，如果收到网关的确认报文，则认为注册成功。再收到网关的否认报文或者响应超时后，节点设备会继续搜索下一个网络 ID，一直到注册成功为止。

2. 节点被动方式

- a. 同节点主动方式，通过远程指定每个网关设备有独立的网络 ID，并确保所有网关的网络 ID 都不同。

“节点被动方式”在响应时间、确定性、功耗、设备成本、系统隐蔽性等方面优于“节点主动方式”。推荐使用“节点被动方式”进行远程网络划分。

- b. 通过特殊的报文（广播网络 ID）实现跨网互通，通过该报文指定任意节点设备的网络 ID。（网络的划分和再划分的工作，简化到向网关设备下达修改任意节点设备的网络 ID 命令）。

对比远程网络划分的两种实现方式，“节点主动方式”要求节点设备不停的检测网络是否发生变化，以便在最短的时间内做出响应，其响应时间具有不确定性。如果网关出现故障，后者网关设备安装迟于节点设备安装，可能会导致节点设备不停的搜索网络，白白消耗功耗。而“节点被动方式”则不要求节点设备在没有数据传输时时刻监听网络的变化，也不会带来任何额外的功耗开销，也不存在响应时间的不确定性问题，很适合于低功耗的仪表组网应用。另外，对于“节点主动方式”要求网关设备记录所管辖的节点设备的地址列表，需要额外的存储空间，这样也增加了设备的成本。因此，“节点被动方式”是 WaveMesh 网络实现远程网络划分的推荐方式，该方式已经简化到仅仅需要 1-2 个 AT 指令就可以实现，这也是本文内容的重点。

跨网互通是实现远程网络划分、再划分的基础。其实现的原理是允许无线模块同时监听和接收来自 2 个不同的网络 ID 的数据报文。

如何实现跨网互通

采用“节点被动方式”实现远程网络划分的关键是如何实现子网间的跨网无线互通，是保证远程网络划分、再划分的关键。WaveMesh 协议网络 ID 长度为 3 个字节，该信息会在所有链路层报文中携带。协议默认（未使能跨网互通）只接收网络 ID 完全匹配的数据报文，对于不同的网络 ID 报文则视为无线干扰。

在 WaveMesh AMR 等协议 1.10 版本之后的固件增加了一个选项是否“使能跨网互通”，在使能该功能之后，可以允许无线模块同时接收来自两个网络 ID 的数据报文。其中一个网络 ID 为子网网络 ID，另外一个为大网网络 ID（所有允许互通的子网需要使用相同的大网网络 ID）。如果一个数据报文中携带的网络 ID 为子网网络 ID，那么该报文仅仅会在子网内被转

跨网互通的实现过程由无线模块自动实习，用户仅仅需要发送 1-2 条 AT 指令即刻完成。

迅速切换路由是跨网互通实现难点，WaveMesh 协议切换路由不需要等待。

发；但如果一个数据报文中携带的网络 ID 为大网网络 ID，那么该报文会在子网之间转发，也就没有子网的边界限制。我们可以利用这个特性实现远程网络划分：

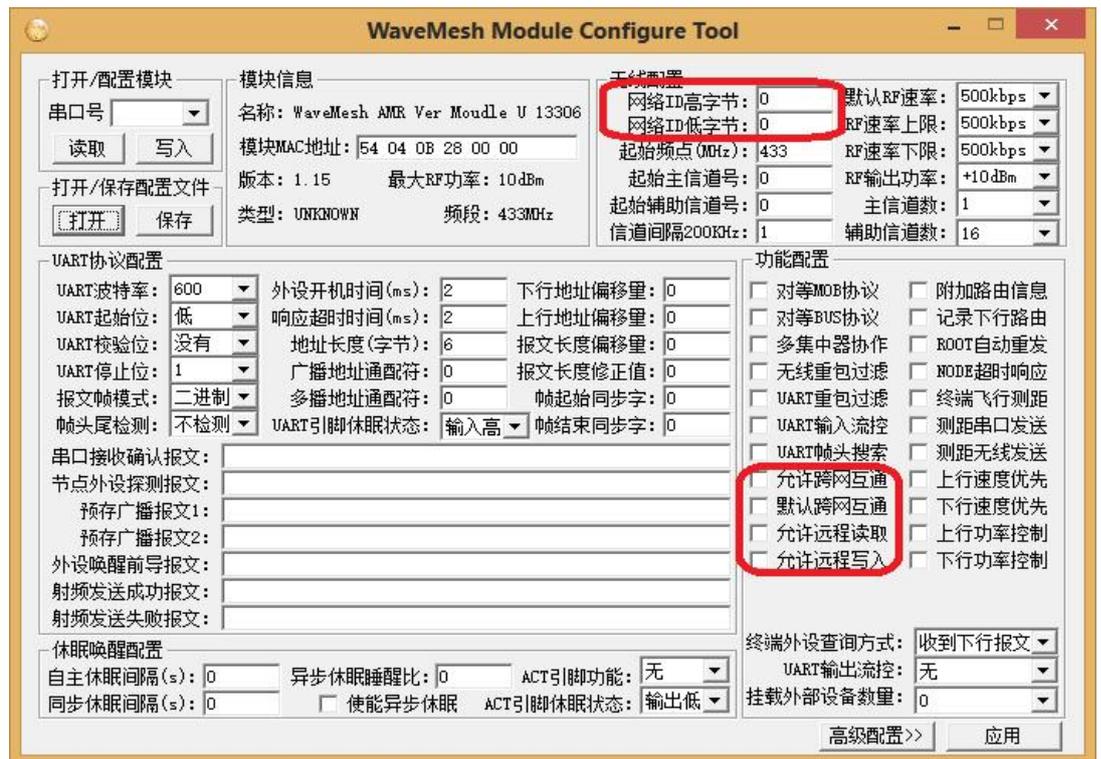
1. 将需要统一划分网络的节点设备、网关设备设置为相同的大网网络 ID、采用相同的起始频点和信道编号，并且使能“跨网互通”；
2. 正常的数据传输采用子网的网络 ID，将数据传输的范围限定在指定的小范围区域；
3. 需要对网络进行划分、再划分时采用大网网络 ID，这样数据传输的范围也就不会受到子网的限制；对网络的划分也就是设定设备的子网网络 ID 的过程。

对网络的远程划分，需要在不同的网络 ID 之间进行选择，需要发送命令修改指定的设备的子网网络 ID，整个过程十分繁琐，需要用户做大量的开发工作才能实现。为此，在 WaveMesh AMR 等协议 1.15 之后的固件版本对设备子网网络 ID 的修改过程做了封装，用户仅仅需要发送几条 AT 指令给网关设备即可实现，网络 ID 的切换、设备子网网络 ID 的修改等工作由模块自动完成。这样，在用户的层面看来，修改子网网络 ID 的 AT 指令和普通的数据报文的发送没任何区别。

实现跨网互通的另外一个难点的是需要在子网和大网之间迅速建立和切换路由，对网络的维护比如划分可能会好几年都不会进行一次。因此，在子网之间的路由需要在需要时临时建立。WaveMesh 协议“零组网时间”的特性在这里发挥的淋漓尽致，无论是在子网内传输还是在子网间传输，都可以立即进行不需要做任何等待。

相关参数

采用“节点被动方式”实现远程网络划分的相关模块配置参数有：“网络 ID 高字节”、“网络 ID 低字节”、“允许跨网互通”、“默认跨网互通”、“允许远程读取”和“允许远程写入”。这些参数在版本 1.15 的中文版的配置工具界面上的位置如下图所示（不同的配置工具版本参数名称和位置可能有所不同）：



网络 ID 高字节

WaveMesh 协议的网络 ID 长度为 3 个字节，由于前两个字节（高字节）同时作为协议物理帧的起始同步字，因此不能随意设置。目前配置工具提供了 256 组选择，通过查表得到实际采用的网络 ID 高字节（前两个字节）。因此该参数“网络 ID 高字节”的值为查表的索引号，并不是实际真实设置的网络 ID 的高字节。

对于远程网络划分来说，大网网络 ID 的高位 2 字节与小网网络 ID 保持一致，因此网络划分并不会修改网络 ID 的高位 2 字节（仅修改网络 ID 低字节），该参数会在正在网络使用中、远程网络划分的前后都会一直保持不变。

需要将组成大网的所有节点、网关模块设置为完全相同的“网络 ID 高字节”才能实现跨网互通、远程网络划分以及现成零配置。该参数一旦选定，就需要保持不变，否则会不能进行跨

网，也就没有办法实现远程网络划分，带来降低维护成本的目的。因此**建议对于一种类型的应用选择一个永久固定的“网络 ID 高字节”，这样会便于维护。**

网络 ID 低字节

WaveMesh 协议的网络 ID 长度为 3 个字节，第三个字节为网络 ID 低字节，该参数为网络 ID 第三个字节的真实值。对于远程网络划分来说，网络 ID 低字节为 0 时为默认的大网网络 ID，不为 0 时为子网网络 ID。这样一个较大的网络可以被划分为 254 个子网，即使子网在地理位置是完全重叠的，也不会相互干扰。如果采用网络 ID 低字节为 0 的网络 ID 的报文，就可以在子网之间进行传输。

举例来说，假设我们用“123.210.xxx”表示一个网络 ID 的三个字节从高到低分别为 123、210 和 xxx，我们可以将“123.210”看成一个网段地址：

大网网络 ID 默认为：“123.210.000”

254 个子网网络 ID 为：“123.210.1”~“123.210.255”

那么，被网络 ID 被设置为“123.210.1”的设备会同时监听、接收和发送网络 ID 为“123.210.000”以及网络 ID 为“123.210.1”的数据报文。而网络 ID 为“123.210.1”的数据报文不会被网络 ID 为“123.210.2”~“123.210.255”的设备转发。

说明：

1. 仅需要用配置工具或者 AT 指令设置设备的子网网络 ID，不需要设置大网网络 ID；大网网络 ID 的数据报文的收发通过“允许跨网互通”选项来使能。
2. 节点设备会自动选择到底是采用大网网络 ID 还是采用小网网络 ID 进行数据转发，连接无线模块的外设不需要进行干涉。具体的选择策略请参考后续章节。

3. 网络 ID 是与设备地址/模块 MAC 地址是完全不同的参数，设置子网网络 ID 对网络容纳的设备数量没有任何影响。一个网段/大网可以被分成 254 个子网，子网的个数有限（254 个），但每个子网可以容纳设备数量没有限制。

既然可以采用远程网络划分修改该参数“网络 ID 低字节”，因此模块/设备在出厂时不需要刻意配置改参数，建议将该参数设置为 0（默认仅接收大网网络 ID 的数据报文）。

允许跨网互通

该选项决定无线模块是否接收携带大网网络 ID 的数据报文。如果不使能该选项，仅仅接收和发送和所设定的网络 ID 完全匹配的数据报文；如果使能该选项，除此之外还可以接收和发送“网络 ID 高字节”相同但“网络 ID 低字节”为零的大网网络 ID 数据报文。

对于需要进行远程网络划分的应用，则需要使能该选项。

默认跨网互通

在使能“允许跨网互通”的功能后，该选项设置模块在对外发送时默认是采用“子网网络 ID”还是采用“大网网络 ID”。无线模块会根据接收到的数据报文自动在两种网络 ID 之间进行切换，这里设置的是初始值。在不使能“允许跨网互通”的情况下，该参数没有意义。

对于需要进行远程网络划分的应用，除非是特殊需要会希望数据报文仅仅限制在小网的范围内转发。因此该参数不应该被默认使能。

允许远程读取

进行远程网络划分往往需要获得当前的节点网络 ID 等参数的设置情况，以及距离网关的距离等信息。出于安全性的考虑，默认情况下是不允许远程访问节点设备的参数信息的，除非明确使能该选项。在使能“允许远程读取”选项后，可以通过远程的 AT 指令读取节点设备的参数配置、内存数据甚至是一些寄存器的即时值，方便远程调试和测试。

对于需要进行远程网络划分的应用，需要使能该参数。

允许远程写入

进行远程网络划分避免不了对节点网络 ID 等参数进行修改。由于模块的参数能够被远程的指令所修改是一件非常不安全的事情，可能会被用来对网络进行恶意破坏。因此该参数默认是不使能的。在使能该参数后，可以通过远程 AT 指令修改无线模块的控制块参数、内存以及部分寄存器的值。

对于需要进行远程网络划分的应用，必须使能该参数以便能够远程修改无线模块的相关参数。

相关 AT 指令

与远程网络划分相关的 AT 指令有 2 大类，一类是远程指令；另外一类是本地指令。涉及到的远程指令有 3 个：ATRN、ATRP 和 ATRI；本地指令也有 2 个：ATCB 和 ATCM。

AT 指令的格式

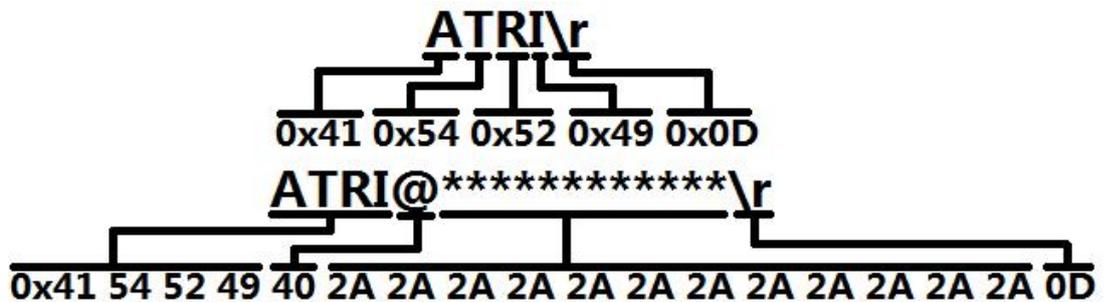
WaveMesh 模块的远程 AT 指令格式：

"AT" 前缀 + ASCII 命令 + ['@' + ASCII 地址 + '=' + ASCII 参数 +] 回车符<CR>

为了便于手工输入，WaveMesh 模块 AT 指令采用 ASCII 字符，以“AT”字符打头和回车符<CR>作为结束。对于地址和参数字段，需要将每字节对应的 HEX（16 进制）值转换为两个字节的 ASCII 字符（去掉 0x 头）。字符“[”和字符”]”之间为可选项，也就是说 AT 指令的执行的目的地址和命令的参数是可选的。如何将地址和参数转换为 ASCII 字符，举例如下：

1. 字节值为 0，则对应的 HEX 值为 0x00，由 ASCII 字符'0'表示 HEX 值中的一个 0，0x00 则需要用两个 ASCII 字符'0'来表示，字符'0'的对应的值为十六进制 0x30。因此，用两个字节“0x30 0x30”来表示一个值为 0 的字节。
2. 字节值为 250，则对应的 HEX 值为 0xFA，由 ASCII 字符'F'和'A'表示 HEX 值中的 F 和 A，字符'F'和'A'的对应的值为十六进制 0x46 和 0x41。因此，需要用两个字节“0x46 0x41”来表示一个值为 250 的字节。

AT 指令的字符对应的十六进制值举例如下图所示：



值得注意的是，最后的'\r'是回车字符值为 0x0D，而不是'\r'字符和'r'字符的组合。举例如下：

1. 发送 ATRI 命令给地址为 1.2.3.4.5.6 的节点，地址长度为 6 个字节，每个字节的 10 进制数由 '.' 进行隔开：
 - a) 字符串为"ATRI@010203040506\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x49 0x40 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x30 0x36 0x0D
2. 发送 ATRI 命令给地址为 0.0.0.0.123.223 的节点，地址长度为 6 个字节，每个字节的 10 进制数由 '.' 进行隔开：
 - a) 字符串为"ATRI@000000007BDF\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x49 0x40 0x30 0x30 0x30 0x30 0x30 0x30 0x30 0x37 0x42 0x44 0x46 0x0D

指定 AT 指令地址

在 AT 指令后可以用 '@' 字符指定执行该指令的目的节点地址。如果一条 AT 指令需要多个节点都需要执行，可以用多个 '@' 字符+ASCII 地址表示。举例如下：

1. 发送 ATRN 命令给地址为 1.2.3.4.5.6 的节点，地址长度为 6 个字节，每个字节的 10 进制数由 '.' 进行隔开：
 - a) 字符串为 "ATRN@010203040506\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x4E 0x40 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x30 0x36 0x0D
2. 发送 ATRN 命令给地址为 1.2.3.4.5.6 和地址为 7.8.9.10.11.12 的节点，地址长度为 6 个字节，每个字节的 10 进制数由 '.' 进行隔开：
 - a) 字符串为 "ATRN@010203040506@0708090A0B0C\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x4E 0x40 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x30 0x36 0x40 0x30 0x37 0x30 0x38 0x30 0x39 0x30 0x41 0x30 0x42 0x30 0x43 0x0D

说明：

1. 远程 AT 指令中携带的地址为原始地址转换为 ASCII 字符后的地址，地址长度会加倍；
2. 节点的地址跟模块所选择的地址模式有关，可以是外设地址也可以是节点的 MAC 地址；
3. 节点的地址长度为多设置的“地址长度”参数决定，不是固定的长度；
4. 如果可以不指定远程 AT 指令目的地址，则会被广播到所有的节点。

AT 指令地址通配符

字符 '*' (十六进制的值为 0x2A) 为 AT 指令的地址通配符，可以取代地址中的某个 ASCII 字节，以达到广播、多播的功能。举例如下：

1. 广播发送 ATRN 命令给所有的节点（假设地址长度为 6 字节，转换为 ASCII 字符之后的长度则为 12 字节）：
 - a) 不指定目的地址的广播指令"ATRN\r"
 - b) 目的地址为全'*'字符的广播命令"ATRN@*****\r"
2. 发送 ATRN 命令给地址前 5 个字节为 1.2.3.4.5 的所有节点（假设地址长度为 6 字节，转换为 ASCII 字符之后的长度则为 12 字节）：
 - a) 字符串为: "ATRN@0102030405**\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x4E 0x40 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x2A 0x2A 0x0D
3. 发送 ATRN 命令给地址前 5 个字节为 1.2.3.4.5 和前 4 个字节为 7.8.9.10 的所有节点（假设地址长度为 6 字节，转换为 ASCII 字符之后的长度则为 12 字节）：
 - a) 字符串为"ATRN@0102030405**@0708090A****\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x4E 0x40 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x2A 0x2A 0x40 0x30 0x37 0x30 0x38 0x30 0x39 0x30 0x41 0x2A 0x2A 0x2A 0x2A 0x0D
4. 发送 ATRN 命令给地址为 1.2.3.4.5.0 ~ 1.2.3.4.5.15 段的 16 个节点（假设地址长度为 6 字节，转换为 ASCII 字符之后的长度则为 12 字节）：
 - a) 字符串为: "ATRN@01020304050*\r"
 - b) 对应的十六进制为 0x41 0x54 0x52 0x4E 0x40 0x30 0x31 0x30 0x32 0x30 0x33 0x30 0x34 0x30 0x35 0x30 0x2A 0x0D

说明：

1. 利用 '@' 和 '*' 字符的组合，可以节省 AT 指令的长度，节省传输时间。
2. WaveMesh 协议默认数据报文的最大传输单元 MTU 为 250 字节，请注意 AT 指令的长度，如果总长度超出 250 字节的限制需要进行分包传输。

ATRN

获取指定远程无线模块/节点设备的当前网络 ID，以及到达网关的综合距离。该综合距离为路由跳数、信号强度、信号质量和电池剩余电量综合计算结果。指令格式：

```
ATRN[@address1@address2@...]<CR>
```

其中“[]”中的参数为可选项，该指令可以指定目的节点地址列表。如果指定的目的节点地址为空时则对全网所有节点有效。

举例（假设节点的地址长度为 6 字节）：

1. ATRN<CR>

获取大网中所有节点的网络 ID

2. ATRN@010203040506<CR>

获取大网中地址为 01.02.03.04.05.06 的节点的网络 ID

3. ATRN@010203040506@0708090A0B0C<CR>

获取大网中地址为 01.02.03.04.05.06 和 07.08.09.10.11.12 的节点的网络 ID

4. ATRN@*****<CR>

获取大网中所有节点的网络 ID

5. ATRN@0102030405**<CR>

获取大网中地址在 01.02.03.04.05.0 ~ 01.02.03.04.05.255 之间的节点的网络 ID

6. ATRN@01020304050* <CR>

获取大网中地址在 01.02.03.04.05.0 ~ 01.02.03.04.05.15 之间的节点的网络 ID

7. ATRN@0102030405**@0708090A0B**<CR>

获取大网中地址在[01.02.03.04.05.0 ~ 01.02.03.04.05.255]以及[07.08.09.10.11.0 ~ 07.08.09.10.11.255]之间的节点的网络 ID

远程节点的返回格式为：

```
ATRN@remote_add=netID distance root_add<CR>
```

其中：

1. “remote_add”为远程节点的 ASCII 字符地址；
2. “netID” 为当前所设置的三字节子网网络 ID 的 ASCII 字符，高字节在前低字节在后；
3. “distance”为远程节点到网关节点的综合距离（并非跳数），取值范围为 1~255，注意这里是 ASCII 字符；
4. “root_add”为网关的 ASCII 字符地址。

举例如下：

```
ATRN@303436464443=11BB02 02 303436464442<CR>
```

上述返回的 ATRN 指令的含义为：地址为 48.52.54.70.68.67 的远程节点，目前的网络 ID 为 17.187.2，到达网关 48.52.54.70.68.66 的综合距离为 2。

- a) 在 AT 指令中远程节点 ASCII 地址为字符 '@' 和字符 '=' 之间的字符串 "303436464443"，其对应 6 个字节的十六进制为 0x30 0x34 0x36 0x46 0x44 0x43，对应的 10 进制的为 48 52 54 70 68 67。
- b) 在 '=' 字符和第一个空格之间的字符串 "11BB00" 为用 ASCII 字符表示的远程节点当前使用的网络 ID，其对应 3 个字节的十六进制为 0x11 0xBB 0x02，对应的 10 进制为 17 187 2。其中 17.187 为网络 ID 的高字节，2 为网络 ID 的低位字节。
- c) 在第一个空格和第二个空格之间的字符串 "02" 为用 ASCII 字符表示的远程节点到达网关的综合距离，其对应 1 个字节的十六进制为 0x02，对应的 10 进制为 2。

- d) 在第二个空格和回车符<CR>之间的字符串 “303436464442 ”为网关的 ASCII 地址，其对应 6 个字节的十六进制为 0x30 0x34 0x36 0x46 0x44 0x42，对应的 10 进制的为 48 52 54 70 68 66。

说明：

1. **ATRN 指令会强制自动采用大网网络 ID 进行传输**，以确保不受子网的限制，可以获得整个大网中所有节点设备的网络 ID；
2. ATRN 指令不仅可以得到所有节点设备的当前使用的网络 ID，还可以得到所有节点到达网络的综合距离，这个信息对网络划分至关重要；
3. ATRN 指令返回的地址长度跟模块配置块中所设置的 “地址长度 “有关，不同的”地址长度”返回的 ATRN 报文长度也会不同；
4. 两个连续 ATRN 指令之间需要保证一定的时间间隔，以便无线模块能够有时间执行。在没有使能休眠的情况下建议最小的时间间隔为 1s 以上，入股使能了异步休眠建议时间间隔为 2-3 倍异步休眠的周期；
5. 值得注意的是，一般大网中会有多个网关设备，为了节省数据传输时间，远程节点返回的数据报文会自动选择不同的网关上报，也就是说 ATRN 的指令返回数据报文并不一定会由下发该指令的网关返回。因此，这就要求一张大网通过不同的网关上报的数据需要在服务器端被汇聚。

ATRP

修改指定远程无线模块/节点设备的网络 ID 低字节（第三个字节），实现远程网络划分的目的。指令格式：

```
ATRP@address[@address2@...]=new_netID_low_byte<CR>
```

该命令必需指定一个以上的目的节点地址，另外新的网络 ID 的低字节作为该命令的参数也不能被省略。

举例（假设节点的地址长度为 6 字节）：

1. `ATRP@*****=22<CR>`

修改大网中所有节点的网络 ID 低字节为 34 (0x22)。

2. `ATRP@010203040506=22<CR>`

修改大网中地址为 01.02.03.04.05.06 的节点的网络 ID 低字节为 34 (0x22)。

3. `ATRP@010203040506@0708090A0B0C=22<CR>`

修改大网中地址为 01.02.03.04.05.06 和 07.08.09.10.11.12 的节点的网络 ID 低字节为 34 (0x22)。

4. `ATRP@0102030405**=22<CR>`

修改大网中地址在 01.02.03.04.05.0 ~ 01.02.03.04.05.255 之间的节点的网络 ID 低字节为 34 (0x22)。

5. `ATRP@01020304050*=22<CR>`

修改大网中地址在 01.02.03.04.05.0 ~ 01.02.03.04.05.15 之间的节点的网络 ID 低字节为 34 (0x22)。

6. `ATRP@0102030405**@0708090A0B**=22<CR>`

获取大网中地址在[01.02.03.04.05.0 ~ 01.02.03.04.05.255]以及[07.08.09.10.11.0 ~ 07.08.09.10.11.255]之间的节点的网络 ID 低字节为 34 (0x22)。

远程节点的返回格式为：

```
ATRP@remote_add=netID distance root_add<CR>
```

该指令返回的报文格式和 ATRN 指令完全一致（除了 AT 指令的名称不同），请参考对 ATRN 返回指令的描述。

说明：

1. **ATRP 指令会强制自动采用大网网络 ID 进行传输**，以确保不受子网的限制，可以修改整个大网中任意节点设备的网络 ID；
2. 网关（ROOT 模块）的网络 ID 不需要和 ATRP 命令中携带的网络 ID 低字节参数相同，比如网关的网络 ID 为：123.231.1 可以修改指定的节点的网络 ID 为 123.231.2；
3. 远程的节点模块必须使能“允许远程写入”选项，否则不能被修改成功；
4. ATRP 指令返回的地址长度跟模块配置块中所设置的“地址长度”有关，不同的“地址长度”返回的 ATRP 报文长度也会不同；
5. 两个连续 ATRP 指令之间需要保证一定的时间间隔，以便无线模块能够有时间执行。在没有使能休眠的情况下建议最小的时间间隔为 1s 以上，入股使能了异步休眠建议时间间隔为 2-3 倍异步休眠的周期；
6. 值得注意的是，一般大网中会有多个网关设备，为了节省数据传输时间，远程节点返回的数据报文会自动选择不同的网关上报，也就是说 ATRP 的指令返回数据报文并不一定会由下发该指令的网关返回。因此，这就要求一张大网通过不同的网关上报的数据需要在服务器端被汇聚。

ATRI

ATRI 命令可以获得当前网络所有节点或者指定节点的运行状态，包括如节点路由、电压、邻居数、射频底噪、信号强度、综合距离、网络 ID、唯一号 UUID 等信息。该命令其实跟网络远程划分没有关系，在这里要提到这个指令有个重要的原因：与 ATRN 和 ATRP 指令不同，ATRI 指令不会强制采用大网网络 ID 进行发送数据报文。因此如果在没有打开“默认跨网互通”或者用 ATCZ 指令指定采用大网网络 ID 下，数据报文会采用小网网络 ID 进行发送。我们可以利用小网网络 ID 传输数据报文特点，对划分后的网络进行测试：

1. 采用小网网络 ID 时远程节点响应的数据报文仅会返回给具有相同小网网络 ID 的网关，而不会发送给别的网关。如果相同小网网络 ID 的网关只有一个，这样上行数据报文的发送就会有确定性。
2. 采用小网网络 ID 时，数据报文仅会在具有相同小网网络 ID 的节点之间中继、路由，而大网中的其他节点对网络 ID 不同的数据报文会完全不做理会。这样会导致节点的密度、路径有可能远远少于大网，可能在大网中能够可靠传输的节点在小网中变得不稳定甚至数据报文传输失败。相对大网，小网中节点到网关的综合距离、链路质量等信息都会有变化，在修改网络 ID 之后可以利用 ATRI 命令来查看新的小网网络的运行状态，是否所有的节点都能够在网络的覆盖范围之内、节点的网络 ID 是否真的设置成功。

指令格式：

```
ATRI[@address1@address2@...]<CR>
```

ATRI 指令的参数和用法与 ATRN 命令相同，请参考 ATRN 命令。

ATRI 指令的返回：

```
ATRI 303436464443 303436464442 7D 7B 73 F7 02 02 00 DC6F04000000 FF 00 010F  
Ult 54 00 11BB00<CR>
```

ATRI 返回的报文如上所示（1.15 软件版本），在 ATRI 之后的每项信息都由空格隔开，依次是：源节点 ASCII 地址、路由节点 ASCII 地址、上行信号强度、下行信号强度、电压、射频底噪、综合距离、邻居数、错误次数、唯一号、签名、重复次数、软件版本号、软件版本类型、温度、频偏和网络 ID。

ATRI 指令不是本文的重点，这里就不再赘述，请参见相关文档。

ATCB

该命令用于通过模块的串口在正常工作模式下读取模块的配置块。采用远程 AT 指令为了安全性考虑对本地模块无效，因此为了获得网关（ROOT 模块）的当前的网络 ID，必须采用本地 AT 指令 ATCB/ATCb，指令格式如下：

```
ATCB offset length<CR>
```

```
ATCb offset length<CR>
```

该命令有 2 个参数：第一个参数为读取控制块的偏移量（16bit），第二个参数是读取控制块的长度（8bit），注意 AT 指令的参数为 ASCII 字符表示的十六进制数，举例说明如下：

1. ATCB000010<CR> 读取偏移量从 0 开始 16 字节长度控制块
2. ATCB002803<CR> 读取偏移量从 40 开始 3 字节长度控制块
3. ATCB010010<CR> 读取偏移量从 256 开始 16 字节长度控制块

ATRB 指令的返回格式为：

```
ATCB data<CR>
```

```
ATCb data<CR>
```

举例说明如下（对应之前的 3 个命令）：

1. ATCB 57 61 76 65 4D 65 73 68 20 41 4D 52 20 55 6C 74<CR>
2. ATCB 00 BB 11<CR>
3. ATCB 52 44 44 0D 00 00 00 00 00 00 00 00 00 00 00 00<CR>

关于 ATCB 的具体的使用以及模块控制块的结构体，请参见相关文档，这里就不再赘述。

对于远程划分网络来说，需要利用该指令获得本地/网关模块当前的网络 ID，网络 ID 参数为 3 个字节，在配置块中的偏移量 10 进制为 40，**低字节在前高字节在后**。因此，获取本地模块网络 ID 的命令为：

```
ATCB002803<CR>
ATCb002803<CR>
```

该命令的返回：

```
ATCB 00 BB 11<CR>
ATCb00BB11<CR>
```

“ATCB”返回的参数每个字节（2 个 ASCII 字符）之间有空格进行隔离，而“ATCb”则没有空格。如果仅想得到网络 ID 的最低位，则可以发送如下的命令：

```
ATCB002801<CR>
ATCb002801<CR>
```

该命令的返回：

```
ATCB 00<CR>
ATCb00<CR>
```

ATCM

该命令用于通过模块的串口在正常工作模式下修改模块的配置块。采用远程 AT 指令为了安全性考虑对本地模块无效，因此为了修改网关（ROOT 模块）的网络 ID，必须采用本地 AT 指令 ATCM，指令格式如下：

```
ATCM offset data<CR>
```

关于 ATCM 的具体使用以及模块控制块的结构体，请参见相关文档就不再赘述，这里仅介绍如何修改网络 ID 的最低字节。对于远程划分网络来说，需要利用该指令修改本地/网关模块的网络 ID，网络 ID 参数为 3 个字节，在配置块中的偏移量 10 进制为 40，**低字节在前高字节在后**。因此，修改本地模块网络 ID 的命令为：

ATCM0028XX<CR>

其中 XX 为新的网络 ID 的低字节对应的十六进制值对应的 ASCII 字符，举例如下：

1. ATCM002800<CR> 修改网络 ID 低字节为 0
2. ATCM002801<CR> 修改网络 ID 低字节为 1
3. ATCM0028FF<CR> 修改网络 ID 低字节为 255

说明：ATCM 命令在修改完模块的控制块后会导致模块复位。

建议设置过程

实例说明

采用“节点被动方式”实现远程网络划分的相关模块配置

常见问题解答

节点设备自动进行大网网络 ID 和子网网络 ID 切换的策略？

节点如果使能“允许跨网互通”选项之后，会同时接收携带大网网络 ID 和子网网络 ID 的两种报文。在节点需要发送时就需要进行选择，到底是采用大网网络 ID 还是子网网络 ID。

WaveMesh 协议的网络 ID 切换策略如下：

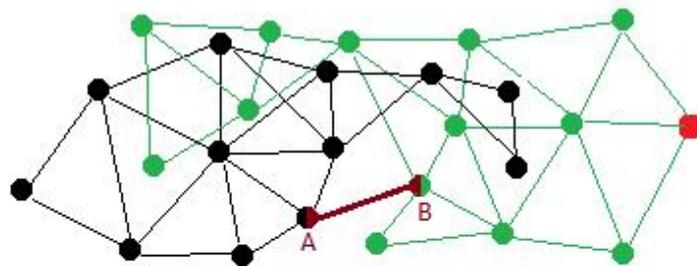
1. 在接收到下行报文通知时，会记录下行报文的网络 ID（大网还是下网）；
2. 采用 1 中记录下的网络 ID 请求接收下行的数据报文；

3. 采用和接收到的请求报文相同的网络 ID 转发下行和接收上行数据报文（有些节点可能会强制进行子网间通信）；
4. 采用 1 中记录下的网络 ID 请求发送上行的数据报文。

由以上的网络 ID 切换策略可以看出：节点模块在和上一级（比自己距离网关更近）节点通讯时采用哪种网络 ID 取决于接收到的最近一次的下行数据报文中携带的网络 ID；节点在和下一级（比自己离网关更远）节点通讯时采用对方使用的网络 ID。

跨网互通还能实现什么别的功能？

跨网互通除了能够实现远程网络划分之外，还能够实现一些特殊的功能，比如网络的级联、指定路由等。如下图所示的网络：



- 节点，子网12.34.56，未使能划网互通
- 节点，子网12.34.78，未使能划网互通
- 节点，子网12.34.56，使能划网互通
- 节点，子网12.34.78，使能划网互通
- 网关，子网12.34.78，未使能划网互通

子网 12.34.56 与子网 12.34.78 在位置上有部分重叠，但是由于属于不同的子网，在没有使能“跨网互通”的节点之间不能进行数据传输。我们在子网 12.34.56 中的 A 点和子网 12.34.78 中的 B 点开启“划网互通”的功能，则可以实现子网 12.34.56 中的节点通过 A-B 的连接桥梁实现与子网 12.34.78 的网关进行通信。这相当于子网 A 级联子网 B，另外 A 中所有的节点都需要经过 A 和 B 进行路由。

大网边界是否也可以限定？

小网的边界可以通过子网的 ID 来划分，而划分网络需要在整个大网中传输 AT 指令。如果大网的规模非常大，大网中的数据传输就会涉及到非常多的点，转发次数越多所需的时间也就越长。而一般小网边界划分的方式基本都是就近原则，比较远的节点即使能够接收和转发 AT 指令也似乎没有太大意义，因为根本就不太可能划分到这样的网络中。是否可以限定大网中数据传输的边界，使得在一定范围之内节点才响应网络划分指令，这样达到节省节点功耗和减小数据传输时间的目的。

WaveMesh 协议中一直都有一个简单的办法去限制网络的边界：最大距离（为综合距离并非路由级数）。WaveMesh 协议默认网络的最大距离为 255，这个参数在很多场合下显得过大。该参数可以通过 ATCH 指令对网关（ROOT 模块）进行设置，该指令的参数即是网络的最大距离。如“ATCH09<CR>”即可以限制网络中最远的节点到达网关的最大距离为 9。该参数可采用何种网络 ID 无关，也就是说即使采用大网网络 ID 也会对网络的边界进行限制。

大网网络 ID 冲突的概率？

大网网络 ID 的三个字节中的最低字节为 0，只有高位 2 个字节可以设置。而高位两个字节作为 WaveMesh 协议物理帧的起始同步字，这就要求这两个字节不能随意设置。目前配置工具给出的选择范围只有 256 个，那么对于频点相同、无线波特率、调制方式都相同的网络，大网网络 ID 的可能发生冲突的概率为 $1/256$ ，这样的冲突概率的确有点大。如何减小大网网络 ID 的冲突：

1. 网络 ID 高两字节采用非配置工具中给出的范围。适合做物理帧的起始同步字的网络 ID 高两字节大概有几千个，采用不同于配置工具给出的值（256 个）还仍然有很多选择可能，使得网络 ID 冲突的概率下降到几千分之一甚至更小。
2. 设置不同物理频点，即使网络 ID 相同但无线报文在物理上并没有冲突。这样做可以进一步降低网络 ID 冲突的概率。

3. 对数据帧进行加密，WaveMesh 无线模块支持 AES-128bit 对称加密传输，即使网络 ID 相同也不会担心应用层数据报文被破解、指令被错误执行。

为什么不直接使用远程参数修改指令 ATRM ？

远程参数修改指令 ATRM 虽然也能够实现远程修改节点配置块中的参数的目的，但是不推荐这么做，原因有以下几点：

1. ATRM 指令不会自动强制使用大网网络 ID 进行发送，也就是说在向网关模块发送该 AT 指令前必须发送别的 AT 指令切换到大网网络 ID 模式，并且在发送 ATRM 指令后必须再次切换回小网网络 ID 模式。而 ATRP 指令会自动完成这个过程。
2. ATRM 指令会导致修改参数的节点模块自动复位，而 ATRP 则不会复位。某些情况下模块复位后会重新获得外设的地址和寻找路由等操作，会导致上行 AT 指令返回报文被外设的响应报文覆盖，或者上、下行发送失败。
3. ATRM 指令可以修改节点模块控制块中的任意参数，如果出现参数错误则可能会造成比较严重的后果，如只能现场修复参数；而 ATRP 则只修改网络 ID 的低字节，即使参数出现错误有可以修正，所以 ATRP 会是个比较安全的指令。