

ETHERNET ■■■■■■ **POWERLINK**

基于 POWERLINK 的冗余方案

王谨秋

POWERLINK 开发板，POWERLINK 协议软件和技术支持，IO 模块，运动控制器，工业机器人控制器。

QQ 群： [151181908](#)

联系人： 王先生

手机： 13917489045（微信同号）

邮箱： openpowerlink@163.com

网址： <http://www.openat.cn>

目 录

冗余设计.....	3
1 双网冗余.....	3
1.1 双网冗余的机制:	3
1.2 线路状态.....	5
1.3 选择器的设计.....	6
1.4 双网冗余的 FPGA 解决方案.....	8
2 环形冗余.....	9
2.1 环形冗余的拓扑.....	9
2.2 环形冗余的机制.....	11
2.3 网络状态检测.....	15
2.4 冗余模块状态检测.....	16
2.5 链路状态检测.....	16
2.6 环形冗余的无缝切换.....	17
2.7 环网冗余的 FPGA 解决方案.....	20
3 多主冗余.....	20
3.1 节点号 NODE ID 的分配.....	21
3.2 配置管理.....	22
3.3 接管仲裁.....	24
3.4 状态切换.....	28
4 冗余系统的典型拓扑结构.....	33
4.1 多主环网.....	33
4.2 多主双网.....	33
4.3 多主双环网.....	34

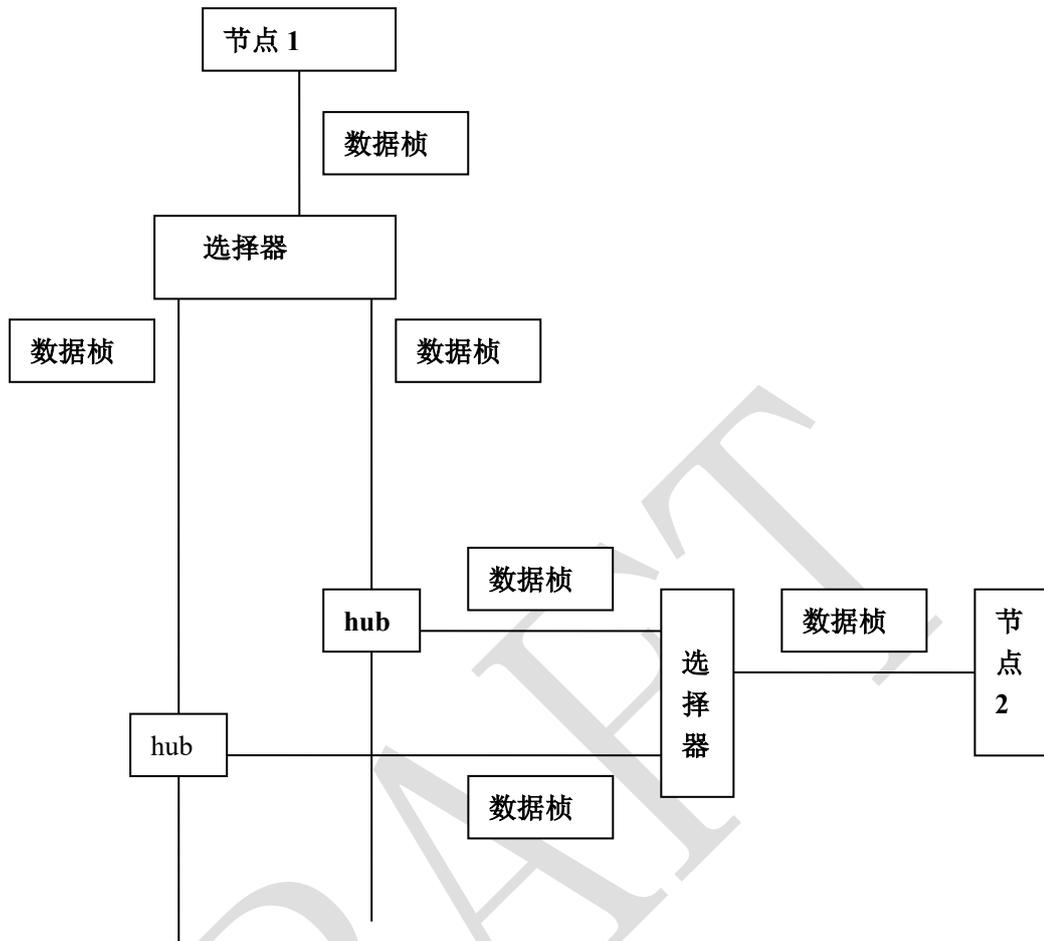
冗余设计

POWERLINK 的冗余包括三种冗余：线缆冗余，环网冗余和多主冗余。
我们提供这三种冗余的 IP。

1 双网冗余

双网冗余顾名思义，就是系统中有两个独立的网络，当一个网络出现故障，另一个网络依然可以工作。双网冗余是一种物理介质的冗余,又称“线缆冗余”，对于每个节点，都有两个网络接口，或多个网络接口。

1.1 双网冗余的机制：



上图所示，节点 1 和节点 2 既可以是主站节点也可以从站节点。节点 1 发出的数据帧在经过选择器的时候，一个数据帧被复制成两个数据帧，同时在两个网络中传输。在接收方节点 2 的选择器处，会同时有两个数据帧到来，选择器选择其中一个，发给节点 2。同理，当节点 2 发送数据时，在经过节点 2 的选择器时，也被复制成两个数据帧，同时在两个网络中传输。在接收方节点 1 的选择器处，会同时有两个数据帧到来，选择器选择其中一个，发给节点 1。

这样，两个网络同时工作，当一个网络出现故障，网络节点可以从另外一个网络收发数据。这种机制的好处是无需网络切换，所以当在一个网络出现故障时，整个系统的通信没有延迟，不会丢失数据帧。

1.2 线路状态

当某个网段出现线路故障时，需要将此信息通知应用程序，以帮助用户修理网络。

选择器检查网络状态的方法如下：

1. 选择器检查每个网络数据接收情况，假设某个选择器有两个网口，一个称为 A 网，另一个称为 B 网。如果 B 网已经接收到数据，而 A 网从 B 网接收到数据时开始，在一段时间 T 内没有收到任何数据（或者在一定时间内 B 网已经收到了多于 2 次数据帧，而 A 网在这段时间内没有收到数据），就说明 A 网出现了故障。也可以简单的检查节点的网口状态，即网口的 Link 状态和 Active 状态的信号。
2. 当选择器检测到某个网络出现故障时，将该信息上报给与其连的 POWERLINK 节点，该 POWERLINK 节点的协议栈需要将这个信息上报给网络中的其它节点，对于从站节点，需要将该信息包含在 PollResponse, StatusResponse, IdentResponse 这三种上报的数据帧里；对于主站节点，需要将该信息包含在 PollReques 数据帧里发给相应的从站。
3. 网路信息在 PollResponse, StatusResponse, IdentResponse, PollReques 数据帧中的位置如下：

Octet Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0	Res	Message Type						
1	Destination							
2	source							
3	xxx							
4	xx	xx	xx	xx	xx	xx	xx	xx
5	FLS	SLS	PR			RS		
6-n	xxx							

在 POWERLINK 数据帧的第 5 个字节的第 6 和第 7 比特位分别表示第一个网络和第二个网络的网络情况：

Bit 7 = First Link Status (FLS) (0 = Link OK, 1 = Link not OK).

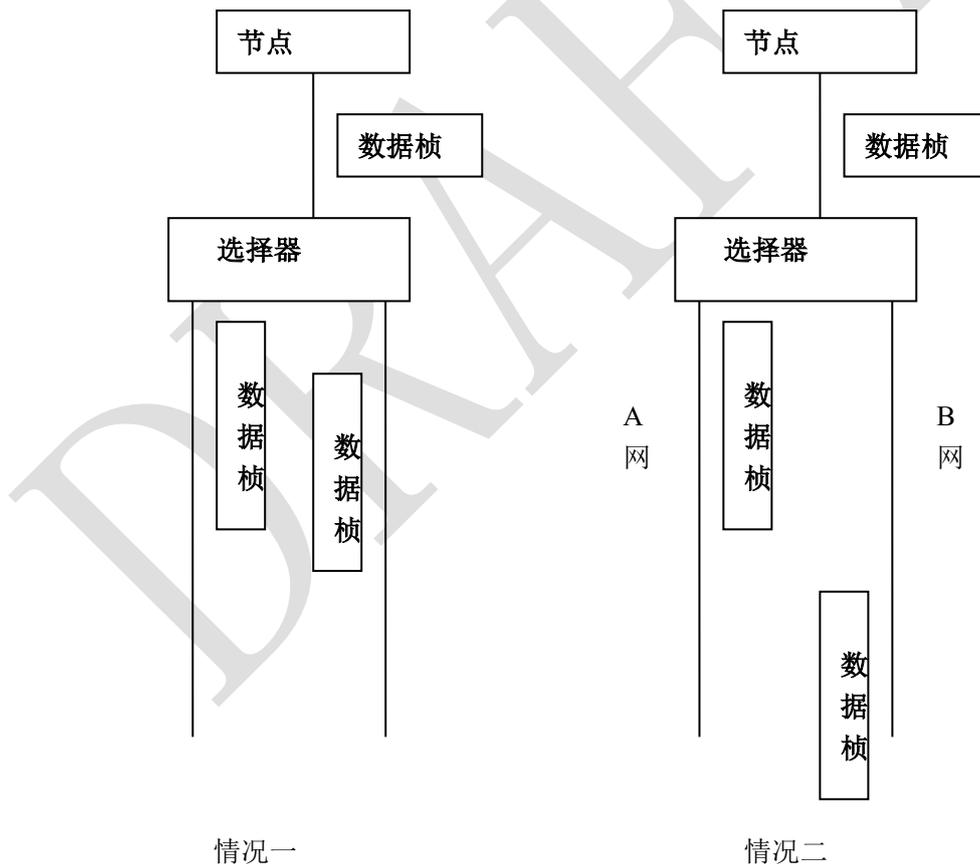
Bit 6 = Second Link Status (SLS) (0 = Link OK, 1 = Link not OK)

注意: POWERLINK 数据帧的第 5 个 BYTE,是以太网数据帧的第 19 个 BYTE。

1.3 选择器的设计

在双网冗余的核心就是选择器，他的功能如下：

1. 发送数据时，将数据帧复制，在两个网络中同时发送。
2. 接收数据时，如果两个网络都正常工作，接收节点将从两个网络收到两包相同的数据，选择器需要从两个网络中选择一个数据帧，发给节点的应用。选择的机制可以很多，最简单的选择机制是选择最先到达的那个数据帧。如下所示数据帧到达某个POWERLINK 节点的时序：



对于“情况一”，从两个网络上传来的数据帧在时间上交叠在一起，这对于选择器来说，很容易确定这两个数据帧是同样的数据帧，选择最早到来的那个数据帧，丢

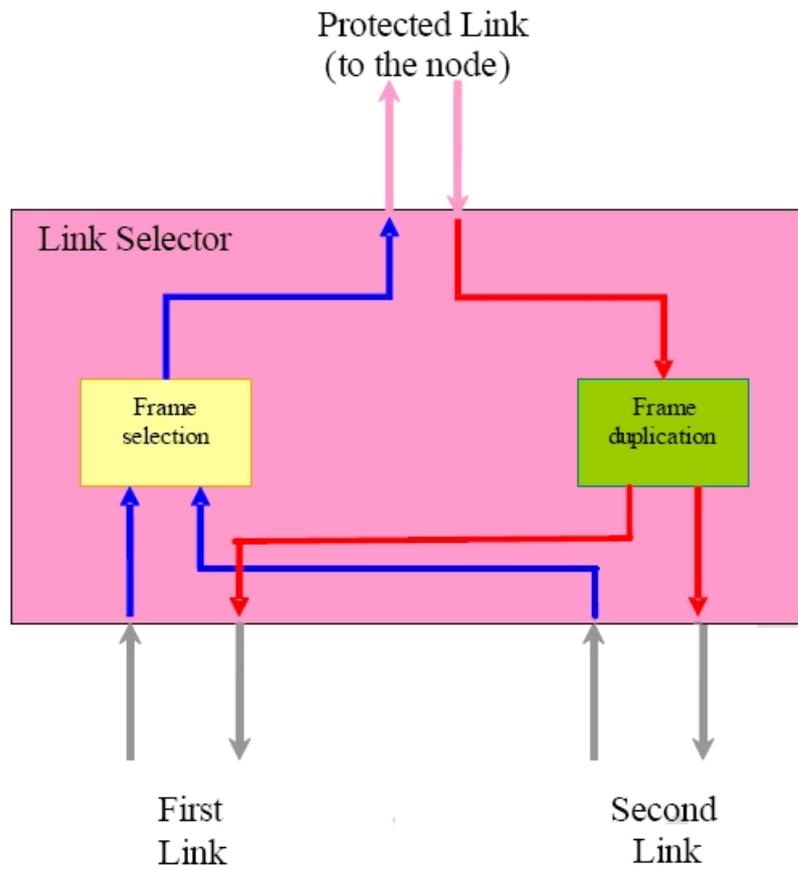
掉另一个。对于这种网络，要求两个网络对同一个数据帧的延迟时间不能大于一个最小以太网帧传输的延迟（对于 100Mbps 网络来说，这个时间是 5.2 微秒）。如果大于了这个时间，就有可能成为“情况二”。

对于“情况二”，选择器需要首先确定这两帧数据是否为相同的数据帧。因为有可能 A 网过来的数据帧是上一次的老数据，而 B 网过来的数据帧是后一次的新数据，此时 B 网过来数据帧需要传给 POWERLINK 节点。也有可能这两个数据帧是同一数据帧，此时 B 网过来数据帧不需要传给 POWERLINK 节点。对于这种情况如何区分？需要在选择器中缓存先到达的数据帧，将后到达的数据帧和前面缓存的先到达的数据进行比较，如果是相同的，就丢掉后到达的数据帧，否则就需要将后到达的数据帧传给 POWERLINK 节点，同时缓存此帧数据，因为该帧数据为一帧新的数据。

3. 在数据帧中填写线路状态信息。

通过一个选择器（Selector）与两个不同的网络相连。双网冗余支持任意的拓扑结构，树形、星形等。在连接的时候，两个网最好选择同样的 HUB、同样长度的线缆，以保证两个网络的数据包到达某一节点是的时间差不超过 5.2us。

在双网冗余中，每个节点需要一个 Selector，该 Selector 可以集成在节点内部，也可以作为一个单独的设备外接在节点外面。每个节点需要把自己要发送的数据同时发送到两个网络上，这样，接收节点会在两个网络上都能收到该数据包，Selector 的作用就是从这两个数据包中选择一个作为接收的数据。

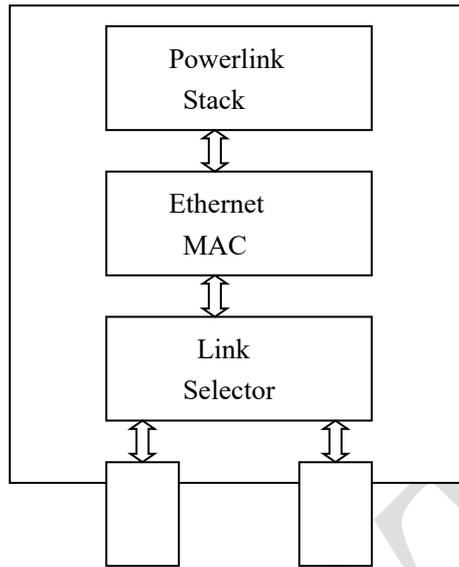


1.4 双网冗余的 FPGA 解决方案

双网冗余的 FPGA 方案如下：

双网冗余是在单网的 FPGA 方案上，将“openHUB”模块替换为“Link Selector”。“Link Selector”的功能如上所述。如需要双网冗余的方案，可以购买基于双网冗余的开发板，我们将提供双网冗余的全套源代码。

更多信息，请登陆 <http://www.powerlink.taobao.com>

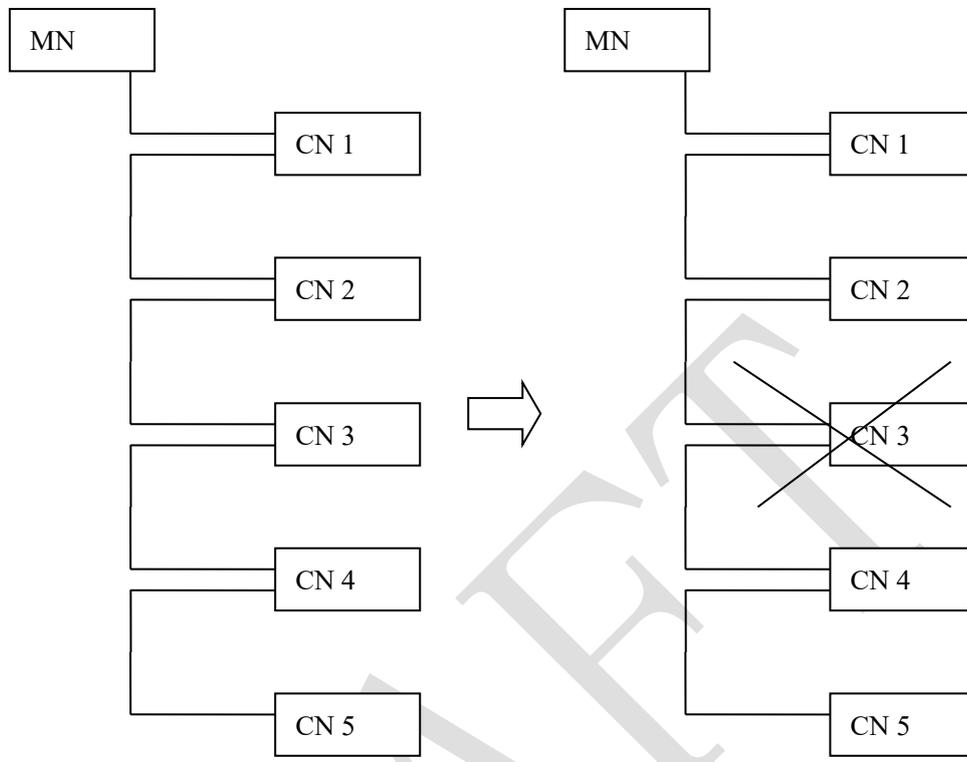


2 环形冗余

环形冗余是一种常用的冗余，也是一种线路的冗余，当菊花链的拓扑结构的最后一个节点再与主站相连接时，就构成了一个环。当某一根线缆出问题，这个系统依然可以继续工作，但是如果两根线缆出现问题，就会导致某个或某些节点从网络中分离开。

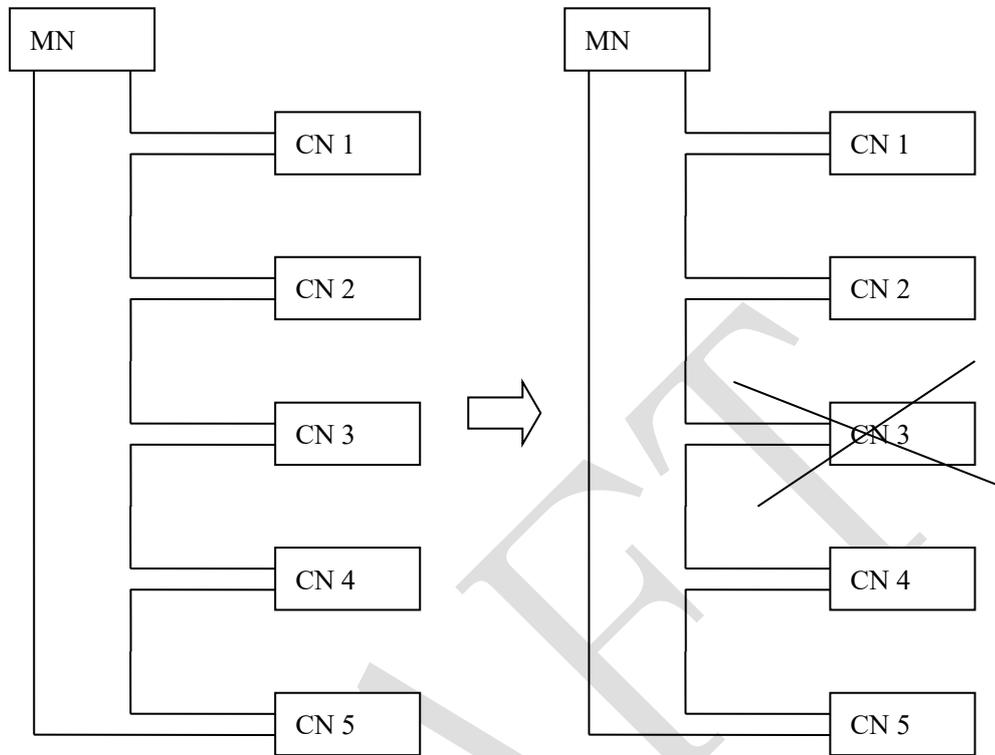
2.1 环形冗余的拓扑

如下图所示两种拓扑结构，对于第一种拓扑结构，如果某个节点或某段线路出现故障，就会导致部分节点从网络上脱离，例如当3号节点出现故障，或者和3号节点相连的线缆发生脱落，就导致4号和5号节点从网络中脱离，而失去控制。



拓扑一

为此，我们引入了拓扑二，我们将最后一个节点和第一个节点用线缆相连，这样整个网络上的节点就构成了一个环，因此称之为环形冗余。对于拓扑二，当3号从节点出现故障，或者和3号从节点相连的线缆发生脱落，4号和5号节点依然和网络相连。因此环形冗余允许一个节点或者一处网段出现故障。



拓扑 二

2.2 环形冗余的机制

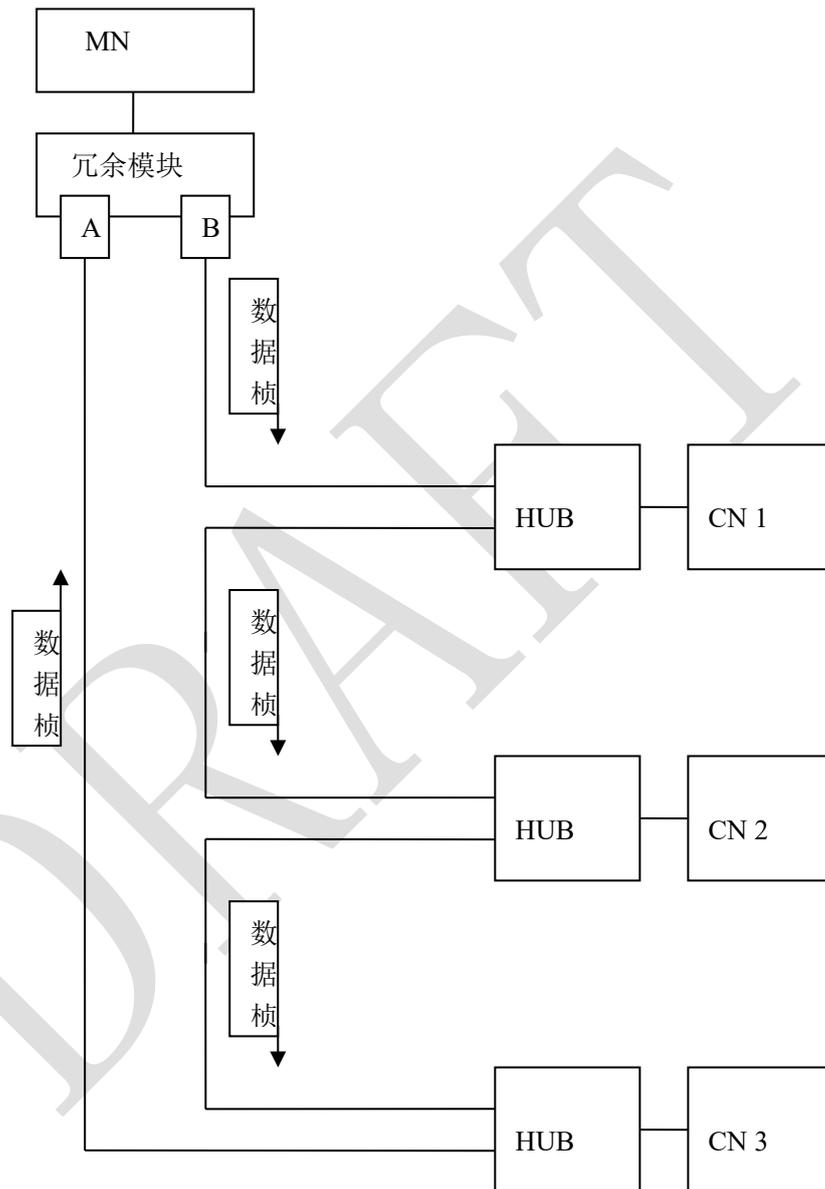
网络闭合时的通信机制：

网络的“环”是闭合的，也就是中间没有断开的点，整个网络形成一个完整的环。

对于环形冗余，有一个节点的模块特殊一些，我们可以称之为“冗余模块”，通常将该模块放置在 POWERLINK 主站节点处，其他节点都是用普通的 HUB。

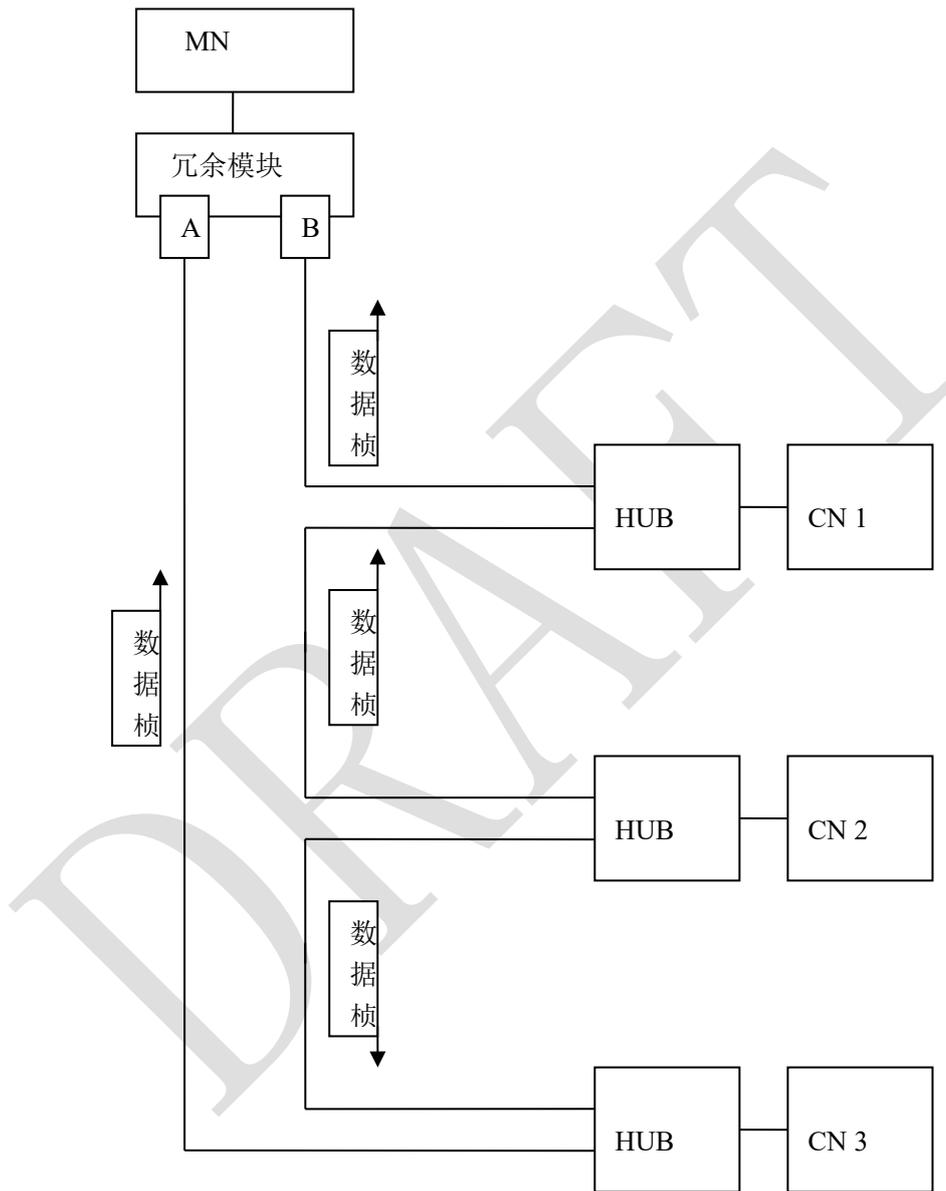
当主站有数据需要发送时，它从冗余模块的两个网口中选择一个，假定我们选择 B 口，将数据发送出去，如果当前网络的“环”是闭合的，也就是中间没有断开的点。这个数据帧从 B 口发出，经过网络上所有的节点，最后到达冗余模

块 A 口。至此，该数据帧已经遍历了网络的所有节点，它的作用已经完成，冗余模块从 A 口接收到该数据帧之后，需要把这个数据帧处理掉，无需再放到网络中传输。



当某个从站节点发送数据时，例如 2 号从节点，如果当前网络的“环”是闭合的，也就是中间没有断开的点，该节点发送的数据会从 HUB 的两个网口同时发送出去，由于网络没有断点，所以该数据帧在遍历完网络所有节点以后，也会

到达冗余模块的 A 口和 B 口，此时它的作用已经完成，冗余模块也需要把这个数据帧处理掉，无需再放到网络中传输。

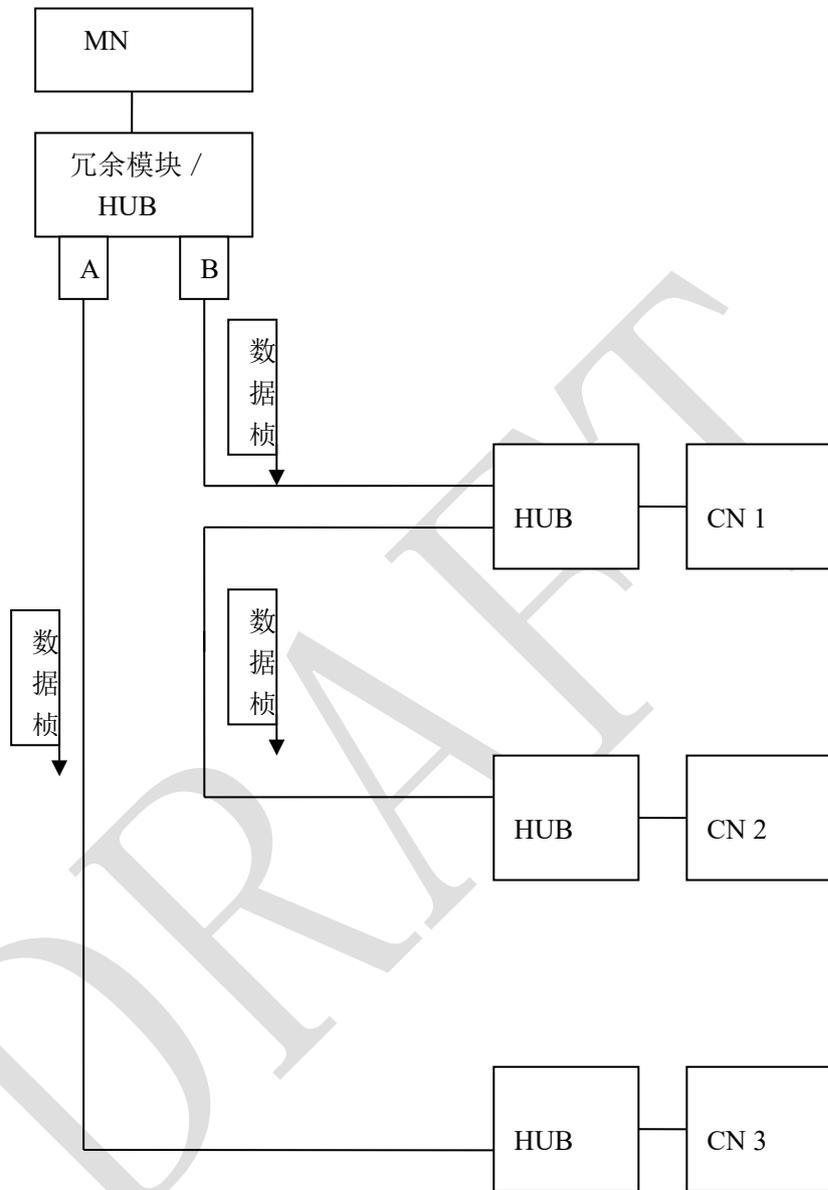


综上，当网络的“环”是闭合的，冗余模块只从一个网口收 / 发数据帧，而另一个网口在收到数据后，将该数据帧吸收掉。

网络开环时的通信机制：

网络开环，也就是网络中有一个断点，使得网络的“环”断开，成为

菊花链结构。如下图所示，假如 2 号从节点和 3 号节点之间的网络脱落。此时冗余模块的作用和 HUB 的功能相同。因为只有这样，才能使网络中的每一个数据帧都遍历整个网络。



因此对于环形冗余，在网络“闭环”时，冗余模块的一个网口用来收 / 发数据，而另一个口用来“吸收”数据；在网络“开环”时，冗余模块就变成了一个 HUB。

2.3 网络状态检测

由于冗余模块的功能在不同的网络状态（开环和闭环）下，功能是不同的，

因此冗余模块需要实时监控网络的状态，根据网络的状态来切换自己的功能。

网络闭环状态的检测：

当网络的状态为闭环时，主站从冗余模块网口 B 发出的数据，能达到冗余模块的网口 A，因此检测网络状态是否为闭环最简单的方法就是监控冗余模块的网口 A 是否收到了来之主站的数据帧，即数据帧中的源 node id 的值大于 239。这些数据帧包括：SoA, SoC, PollPreq, PollPresMN, ASND 等，通常采用 SoA 数据帧。

网络开环状态的检测：

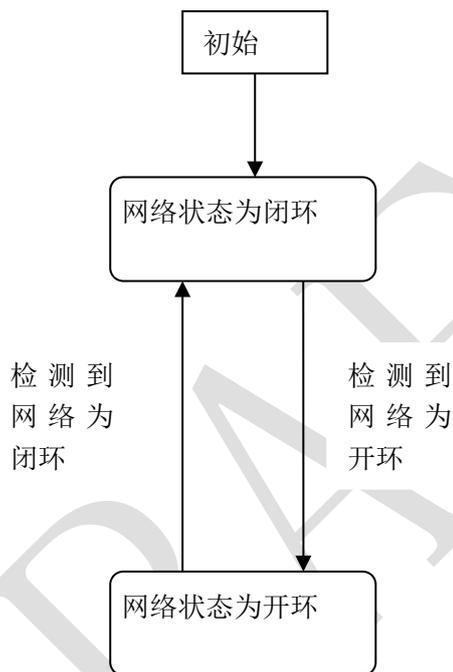
当网络的状态为开环时，主站从冗余模块网口 B 发出的数据，不能达到冗余模块的网口 A，因此检测网络状态是否为开环最简单的方法就是监控冗余模块的网口 A，如果冗余模块的网口 A 在一段时间 T 内没有收到来自主站的数据，说明网络状态为开环。这里最关键的是的时间参数 T 的选择，如果 T 选的比较小，就会出现错误的判断，如果 T 选的比较大，就会导致丢失太多的数据帧。时间参数 T 的选取主要取决于网络的传输延迟，即数据帧从冗余模块网口 B 发出，至到达冗余模块网口 A，所经历的时间 T_{delay} 。当主站从冗余模块网口 B 发出一个数据帧，如果网络是闭合的，那么冗余模块网口 A 应该在时间 T_{delay} 后收到该数据，如果没有收到，就说明网络是开环的。时间 T_{delay} 可以通过在冗余模块中实际测量得到。

一种比较简单的算法：冗余模块网口 B 已经连续发送了 2 次的 SoA,但是冗余模块网口 A 还没收到来自主站的数据，这时说明网络状态是开环的。因为冗余模块网口 B 已经连续发送了 2 次 SoA 的时间间隔为一个 POWERLINK 循环周期，数据帧在网络中的传输时间 T_{delay} 一定小于 POWERLINK 循环周期。否则，意味着主站发送数据至最后一个节点的时间大于一个 POWERLINK 循环周期，这是不可能的。

2.4 冗余模块状态检测

如前所述，冗余模块的功能和当前的网络状态有关，当网络状态为闭环

时，冗余模块的网口 A 需要吸收掉接收到的数据帧，同时不断检测当前的网络状态，当发现网络状态变为开环，就切换到开环状态；当网络状态为开环时，冗余模块成为一个 HUB，冗余模块的网口 A 就成了 HUB 的一个接口，同时不断检测当前的网络状态，当发现网络状态变为闭环，就切换到闭环状态。下图为冗余模块工作状态图：



注意：当网络状态从闭环变为开环时，由于检测算法的原因，会导致丢失一个循环周期内的部分数据；当网络状态从开环变为闭环时，由于检测算法的原因，会导致丢失一个循环周期内的部分数据在网络中一直被传递，直到冗余模块的工作状态切换到闭环状态，冗余模块的网口 A 将其吸收为止。

2.5 链路状态检测

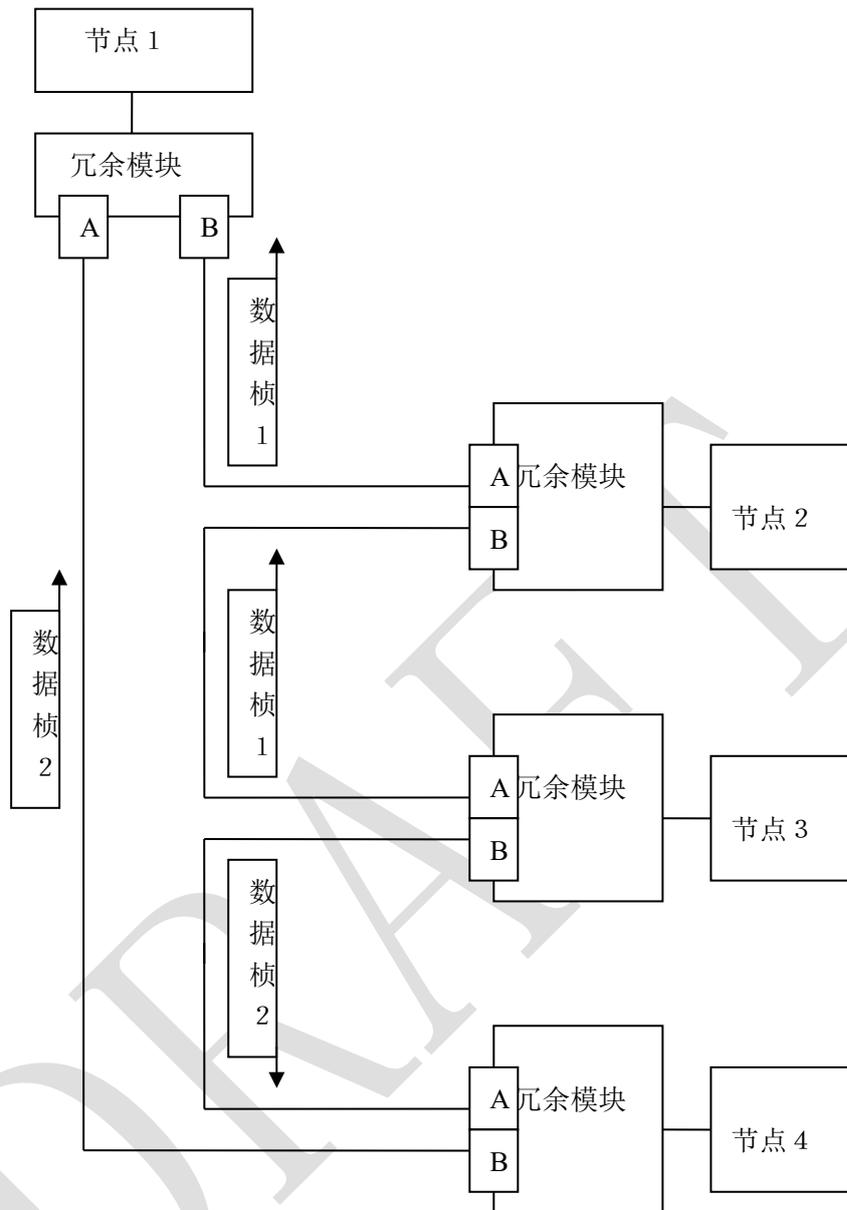
当网络的某个点出现故障时，网络中的节点需要将此信息传递到应用层，方便用户检查并修理故障点。HUB 和冗余模块的每个端口都需要检测对数据帧的接收情况。某个端口如果在一段较长的时间内没有收到数据，说明有可能与该端口连接的网段出现了故障，HUB 或冗余模块需要将该端口的状态信息上报到 POWERLINK 协议栈。POWERLINK 协议栈中可以定义一个 object 来保存该信息。其他的节点可用通过 PDO 或者 SDO 的方式得到该信息，上报给其他节点的应用程序。

2.6 环形冗余的无缝切换

此前所讲的环形冗余，网络从闭环状态转变为开环时，会丢失一个循环周期的部分数据；网络从开环状态转变为闭环时，会导致一个循环周期的部分数据在网络上传输多次，造成网路风暴。也就是说前面的设计方案，在网络状态切换时会造成一个循环周期内的部分数据丢失或多次传输。大部分应用场合，可以接受这样的扰动，而有些应用场合则要求无缝的切换，也就是网络状态切换时，网络通信不受影响。

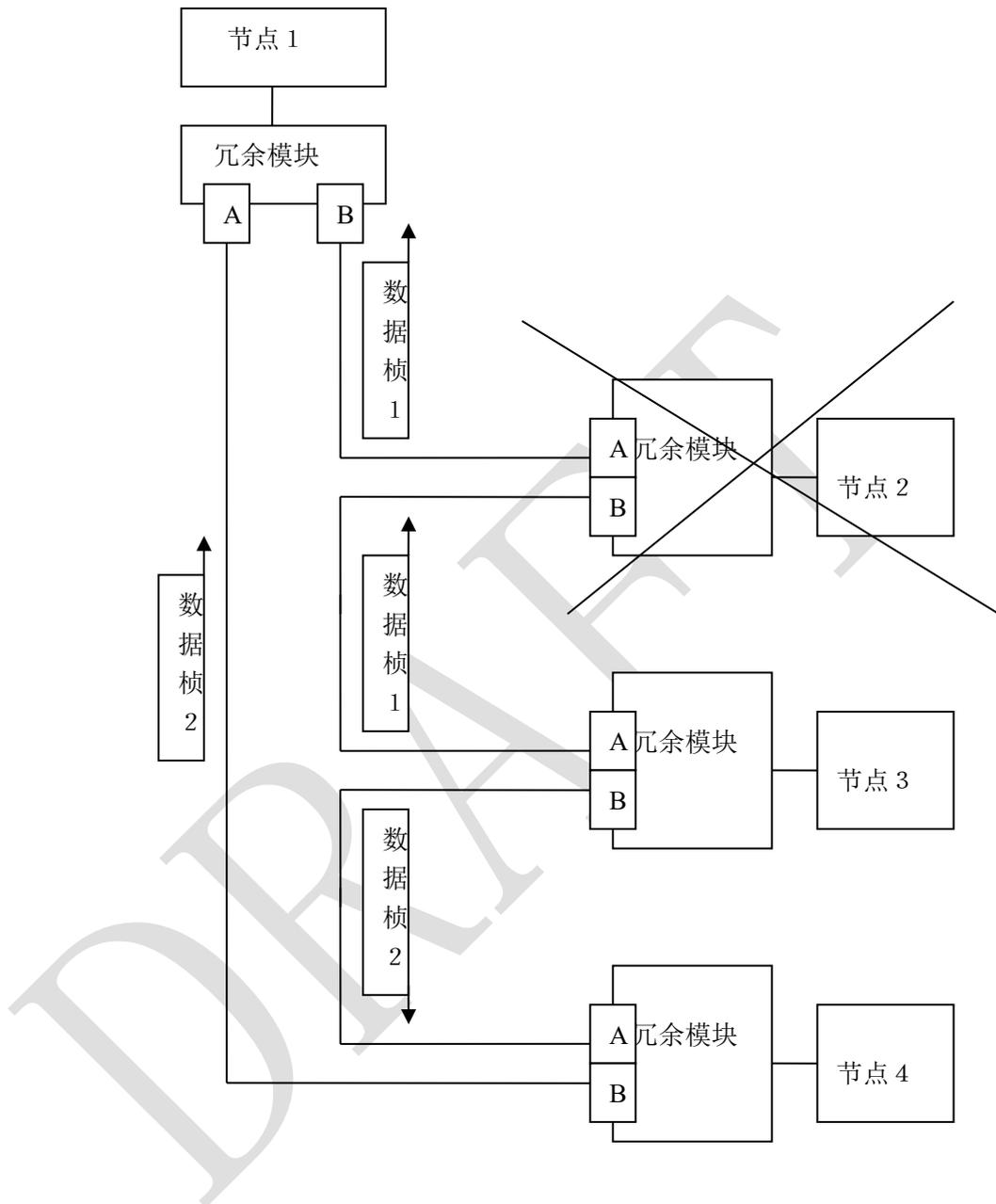
为了达到无缝切换的效果，我们对 POWERLINK 稍加改动，就可以实现。环形冗余网络无缝切换的架构如下：

DRAFT



上图描述了节点 3 将要发送的数据帧，从冗余模块的两个端口同时发出，假如这两帧数据同时到达节点 1 的冗余模块，这里约定从端口 A 发出的数据帧为数据帧 1，从端口 B 发出的数据帧为数据帧 2。这两帧数据，就像两列列车，分别沿着顺时针方向和逆时针方向行驶，列车经过的地方，就是数据到达的地方。两列列车，相向而行，必然会在某一个点相遇，假设数据帧 1 和数据帧 2 在节点 1 处相遇，这时冗余模块从两个数据帧中选择一个上报给与该冗余模块相连的节点，同时数据帧 1 和数据帧 2 中止在网络上传输。由于 POWERLINK 是工作在以太网的半双工模式下，所以一个端口在接收数据的同时，不能发送数据，所以必然会导致数据帧 1 和数据帧 2 在相遇节点处的冲突。

当网络出现某个断点时，例如节点 2 从网络上断开，数据帧 1 和数据帧 2 依然可以遍历网络上所有的节点，只是这两帧数据不会发生碰撞。



因此，此方案中，每个节点的冗余模块不单单是一个 HUB,而是具有如下功能：

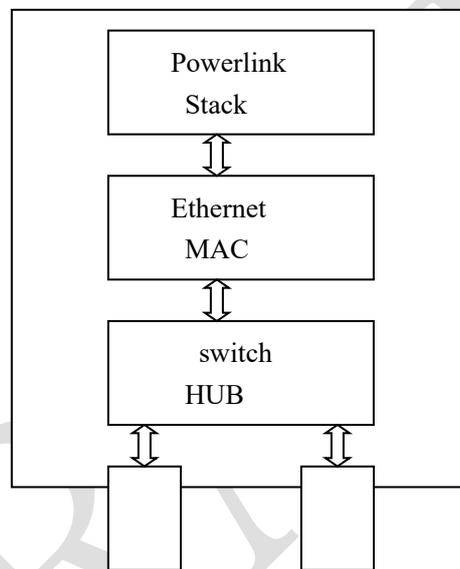
1. HUB 功能
2. 对于同时收到的两个相同数据帧， 选择一个上报给与该冗余模块相连的节点， 同时中止这两个数据帧在网络上传输。比较简单做法是， 当在某个冗余模块中检测到冲突， 就选择一个上报给与该冗余模块相连的节点， 同时中止这两个数据帧在网络

上传输。因为 POWERLINK 的数据帧具有严格的时序，如果不出故障，就不会出现数据帧传输的冲突。

2.7 环网冗余的 FPGA 解决方案

环网冗余的 FPGA 方案如下：

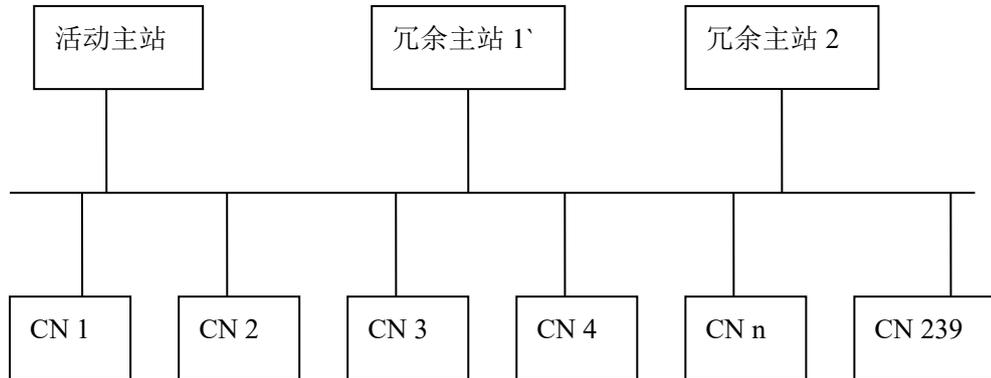
环双网冗余是在单网的 FPGA 方案上，将“openHUB”模块替换为“Switch HUB”。“Switch HUB”的功能如上所述。如需环网冗余的方案，可以购买基于环网冗余的开发板，我们将提供环网冗余的全套源代码。更多信息，请登陆 <http://www.powerlink.taobao.com>



3 多主冗余

由于一个 POWERLINK 网络有且只有一个 MN，当正在工作的主站出现故障时，网络就会瘫痪，因此对于要求较高的一些场合，需要多主冗余。在一个系统中，存在多个主站，其中一个处于活动状态，其他的主站处于备用状态，当正在工作的主站，出现故障，备用主站接替其工作，继续维持网络的稳定运行。

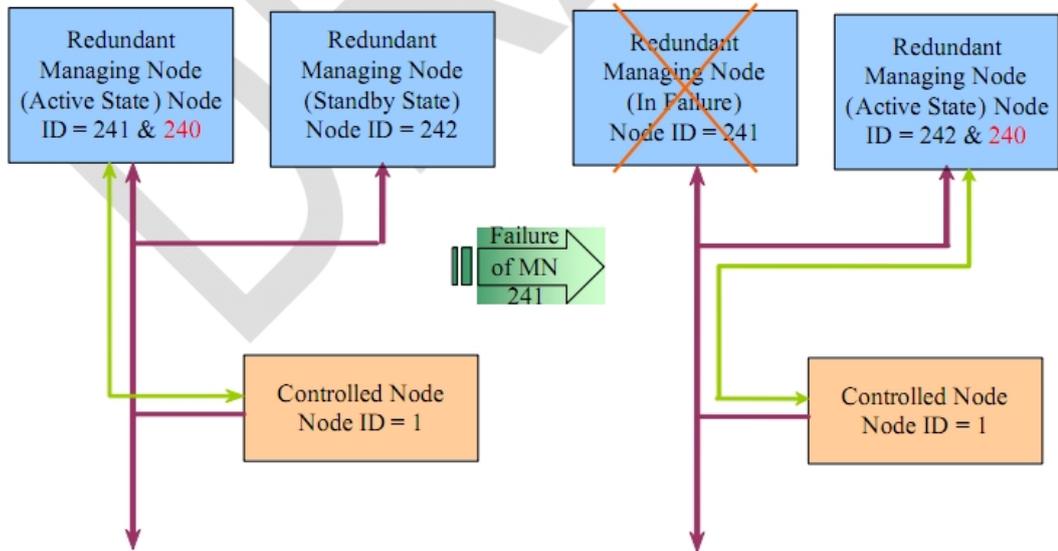
如下图所示：



3.1 节点号 NODE ID 的分配

对于 POWERLINK 的从站节点号的范围是从 1 至 239；而对于标准的主站，节点号为 240；对于主站冗余系统，由于有多个主站，这些主站的节点号范围为 241 至 250。

在多主冗余中，无论是活动主站还是备用主站，他们的节点号范围是从 241 至 250。对于活动的主站，在向外发送 SoC, SoA, PollRequest, ASND 数据帧时，将其源 NODE ID 的值设置为 240；在向外发送 PollResponse 数据帧时，数据帧中源 NODE ID 的值设为活动主站自身的 NODE ID。对于备用主站，使用其自身的节点号来收发数据。如下图所示，当节点号为 241 的主站处于活动状态时，他使用 241 和 240 作为自己的节点号，来收发数据，当节点号为 241 的主站出现故障，节点号为 242 的主站接替其工作时，节点号为 242 的主站使用 241 和 240 作为自己的节点号，来收发数据。



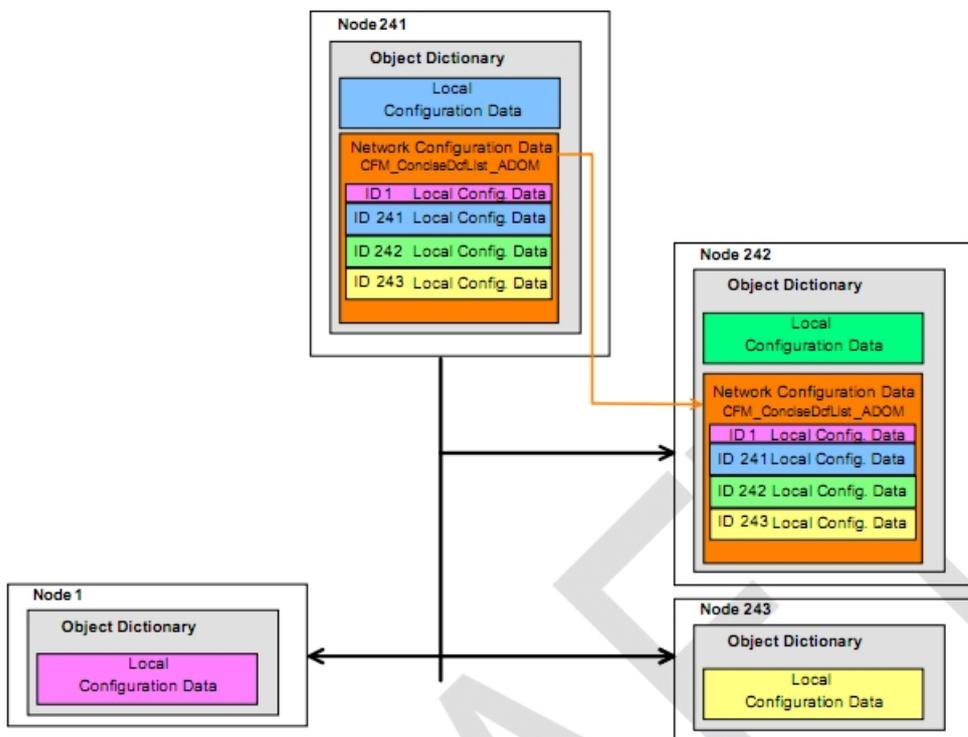
如下所示,当活动主站从 MN1 切换到 MN2 时,MN1 和 MN2 发送的数据帧的目标 MAC 地址和 POWERLINK 源地址, 目标地址值的变化。

MAC Source	MAC Destination	Message Type	POWERLINK src	POWERLINK dest
MAC-MN1	01:11:1E:00:00:01	SoC	240	255
MAC-MN1	MAC-CN1	PollRequest	240	1
MAC-CN1	01:11:1E:00:00:02	PollResponse	1	255
MAC-MN1	MAC-MN2	PollRequest	240	242
MAC-MN2	01:11:1E:00:00:02	PollResponse	242 ³	255
MAC-MN1	01:11:1E:00:00:02	PollResponse	241 ³	255
MAC-MN1	01:11:1E:00:00:03	SoA	240	255
Switch-over, MN2 becomes active				
MAC-MN2	01:11:1E:00:00:01	SoC	240 ⁴	255
MAC-MN2	MAC-CN1	PollRequest	240	1
MAC-CN1	01:11:1E:00:00:02	PollResponse	1	255
MAC-MN2	MAC-MN1	PollRequest	240	241
MAC-MN1	01:11:1E:00:00:02	PollResponse	241 ³	255
MAC-MN2	01:11:1E:00:00:02	PollResponse	242 ³	255
MAC-MN2	01:11:1E:00:00:03	SoA	240	255

3.2 配置管理

主站的对象字典中保存了对网络上所有从节点和冗余主站的配置信息。当网络上电以后,主站首先检查是否需要配置网络上的其他节点,如果需要,就根据保存在主站内的配置信息通过 SDO 的方式发送给相应的节点。当某个节

点重新回到网络中时，主站也会去检查是否需要重新配置该节点。因此活动主站从 MN1 切换到 MN2 时，MN2 的对象字典里也需要保存有与 MN1 相同的对网络上节点的配置信息。



3.3 接管仲裁

由于在一个系统中有多个备用主站，当活动主站出现故障时，需要有

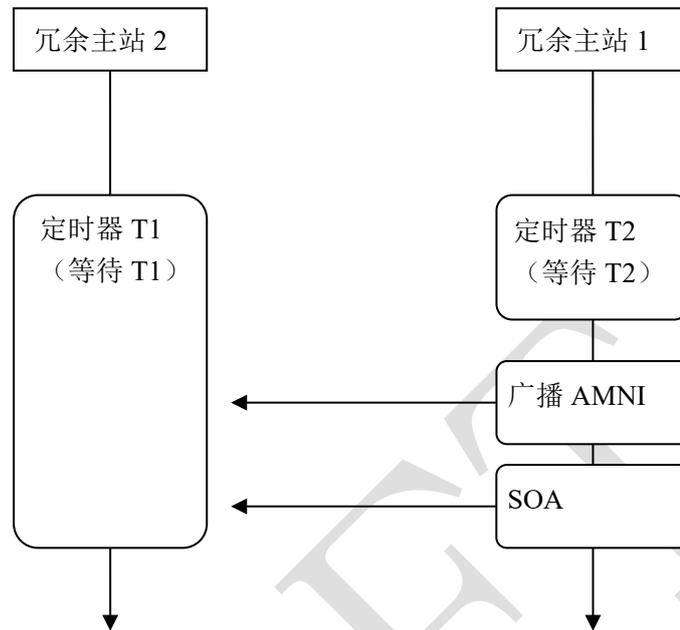
一种机制,从多个备用主站中选择一个接替活动主站的工作。这里采用竞争机制,每个备用主站有唯一的优先级,这个优先级有两种方式确定:

1. 在备用主站中定一个 **object**, 用来保存优先级参数, 这个 **object** 的索引为 0x1f89,子索引为 0x0a。在组建 POWERLINK 网络时, 事先配置好每个备用主站的优先级。优先级是一个数值, 取值范围为 1, 2.....等自然数。数值越小, 优先级越高。

2. 由节点号来确定优先级。由于每个备用节点有唯一的节点号, 可以将节点号的数值看作是优先级的值。

系统上电时,多个冗余主站同时启动,此时哪个冗余主站应该率先接管网络,成为活动主站? 这由前面所说的优先级决定。所有冗余主站根据优先级的不同,分别等待不同的时间。如果在等待的时间内收到了 SOA 数据帧,说明已经有其他的主站开始工作,该冗余主站进入 **standby** 状态;如果在等待的时间内没有收到 SOA 数据帧,该冗余主站进入活动主站状态,进入 **Reduced Cycle**。由于 POWERLINK 主站在启动的时候,首先进入到 **Reduced Cycle**,在这个状态下,主站只是产生 SOA 和 ASND 来配置网络。如果活动主站在这个正在工作在该状态下,由于某种故障停机,备用主站需要有能力检测,并根据优先级的机制选出一个备用主站,接替活动主站的工作。因此当活动主站工作在 **Reduced Cycle** 状态时,备用主站需要周期性检测 SOA,如果很长时间没有收到 SOA,就认为活动主站出现故障。具体的方法是在备用主站中实现一个定时器。定时器的定时时间为 **T_reduced_switch_over_MN** 是指如果在该时间间隔内没有收到 SOA,就认为网络中没有活动的主站来产生 SOA。备用主站每次收到 SOA 数据帧,就重新设置自己的定时器;如果定时器的定时时间到,但是还没收到 SOA 数据帧,就启动接管程序。启动接管程序时,首先发送一个广播消息 **ActiveManagingNodeIndication (AMNI)**,来通知网络上其他的备用主站,本主站将要成为活动主站。其它的备用主站,在收到该消息时,需要重新设置自己的定时器。

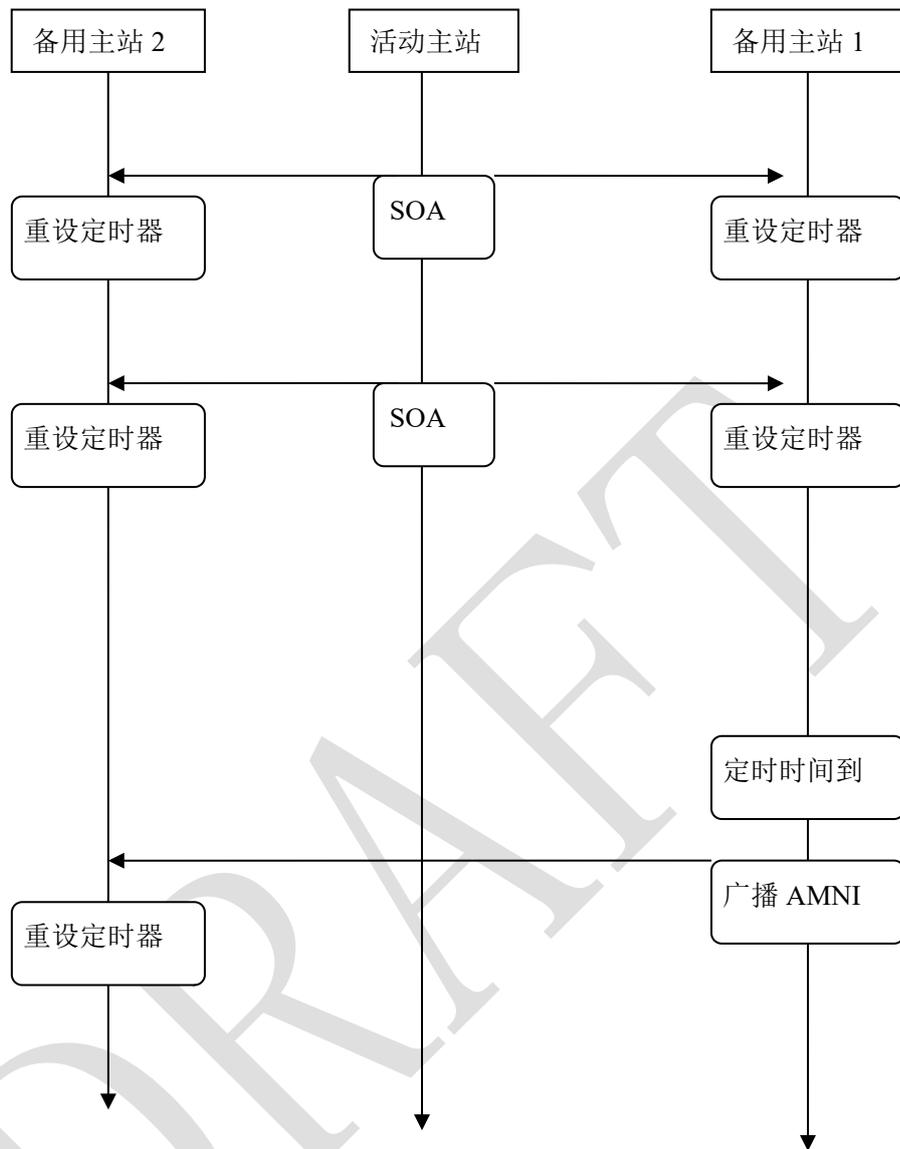
优先级高的备用主站, **T_reduced_switch_over_MN** 的时间小,这样的备用主站等待时间短,优先级低的备用主站, **T_reduced_switch_over_MN** 的时间大,定时器等待的时间长。



启动时的时序

冗余主站 1 的优先级高, 等待的时间短, 当定时结束后, 还没有收到 SOA/SOC 数据帧, 会发送广播 AMNI 数据帧, 通知其他冗余主站, 将由本站做为活动主站。冗余主站 2 的优先级高, 等待的时间短, 当定时结束后, 冗余主站 2 收到了 AMNI 数据帧或者 SOA 数据帧, 这时冗余主站 2 将进入到” standby”状态。

注意在启动时, 这里的 T1, T2 要设置的足够长, 以便有明显的优先级, 如果|T1-T2| 的值太小 (例如几个微秒), 由于冗余主站 1 和冗余主站 2 在上电顺序, 初始化的速度等差异, 导致两个设备不能同时启动定时器, 从而使优先级失效, 因此最好把这两个参数设置的大一些, 而且之间的差值也足够大。



当活动主站进入到 POWERLINK 正常的循环周期，会周期性产生 SOC 数据帧。在该状态下，备用主站需要检测 SOC 数据帧，如果很长时间没有收到 SOC 数据帧，就说明当前的活动主站出现故障，优先级最高的备用主站将接替活动主站的工作。备用主站检测 SOC 数据帧的方法依然是通过设置定时器。定时器的值如下：

$$T_{switch_over_MN} = T_{cycle_MN} + \frac{T_{cycle_MN} * (MNSwitchOverPriority_U32 + MNSwitchOverDelay_U32)}{MNSwitchOverCycleDivider_U32}$$

Tcycle_MN 是 POWERLINK 循环周期长度

MNSwitchOverPriority_U32 是节点的优先级，优先级越高，该数值越小。

MNSwitchOverDelay_U32 是一个可选的延时参数，该参数不是必须的。

MNSwitchOverCycleDivider_U32 是一个除数因子，他将循环周期等分，得到较小的时间片。

在主站进入到 operational 工作模式时，之所以采用监控 SOC 数据帧而不是 SOA 数据帧，原因是 SOC 数据帧的抖动更小，更准确。

对于备用主站，每次收到 SOC 数据帧时，定时器清零重新启动。当某个备用主站的定时器定时时间到，说明该备用主站超过一个循环周期没有收到 SOC 数据帧，意味着活动主站可能已经出现故障（但是也不排除如下两种可能：SOC 抖动过大和 SOC 数据帧丢失）。每个备用主站，由于优先级不同，所以每个备用主站的定时器设置的时间不同。当某个备用主站的定时时间到，它就发送一个 AMNI 消息来通知网络上的其他备用主站，本备用主站将要接替活动主站的工作，其他备用主站收到 AMNI 消息，需要将自身的定时器清零重新设置。下面举例说明：

假设在我们的系统里有 3 个冗余主站 RMN (POWERLINK node-ID 241, 242, 243) ， 241 是活动主站，假设他现在出现了故障，仲裁机制启动：

RMN241

- o MNSwitchOverPriority_U32 = 1

- o MNSwitchOverCycleDivider_U32 = 3

RMN242

- o MNSwitchOverPriority_U32 = 2

- o MNSwitchOverCycleDivider_U32 = 3

RMN243

- o MNSwitchOverPriority_U32 = 3

- o MNSwitchOverCycleDivider_U32 = 3

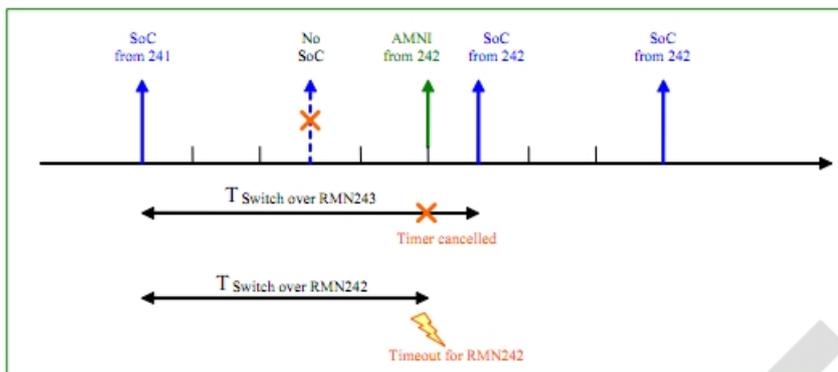
下面是各个备用主站的定时器定时时间。

$$T_{switch_over_MN241} = T_{cycle_MN} + \frac{T_{cycle_MN} * (1+0)}{3} = T_{cycle_MN} + \frac{T_{cycle_MN} * 1}{3}$$

$$T_{switch_over_MN242} = T_{cycle_MN} + \frac{T_{cycle_MN} * (2+0)}{3} = T_{cycle_MN} + \frac{T_{cycle_MN} * 2}{3}$$

$$T_{switch_over_MN243} = T_{cycle_MN} + \frac{T_{cycle_MN} * (3+0)}{3} = T_{cycle_MN} + \frac{T_{cycle_MN} * 3}{3}$$

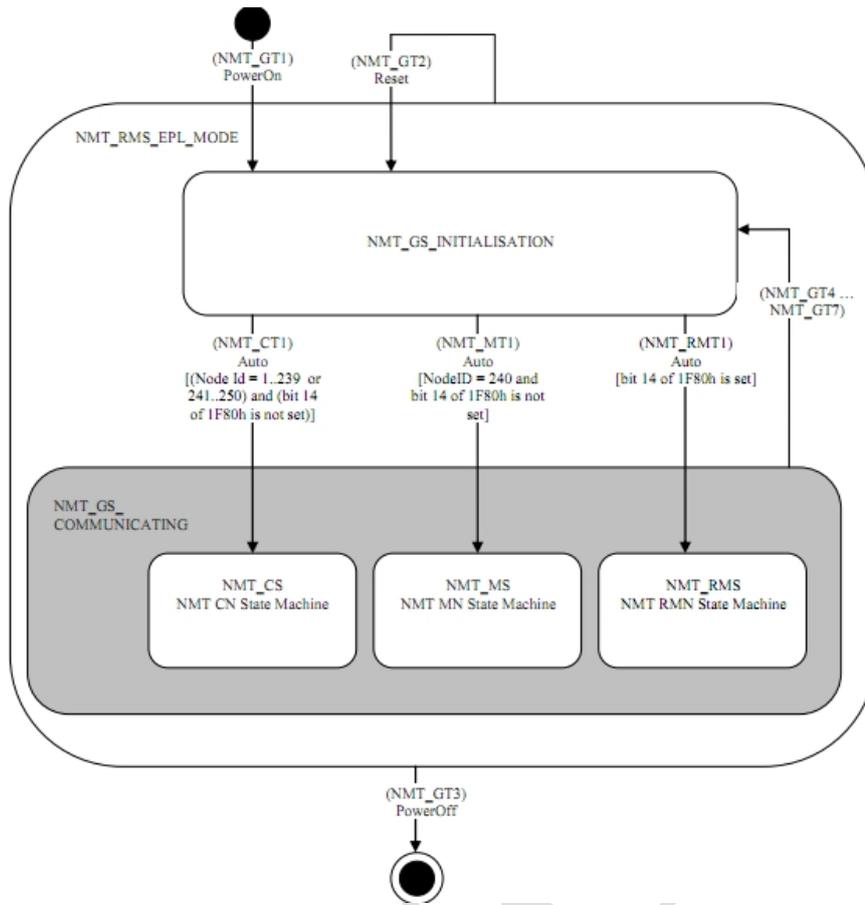
下图为接管的时序图。当 242 号备用主站在规定的时间内没有收到 SOC 数据帧，就发送一个 AMNI 消息，然后接管活动主站的工作。243 号备用主站收到 AMNI 消息后，将定时器清零重新设置。



除此之外，应用程序可以通过发送命令让本地或者远程的活动主站进入到 standby 状态。

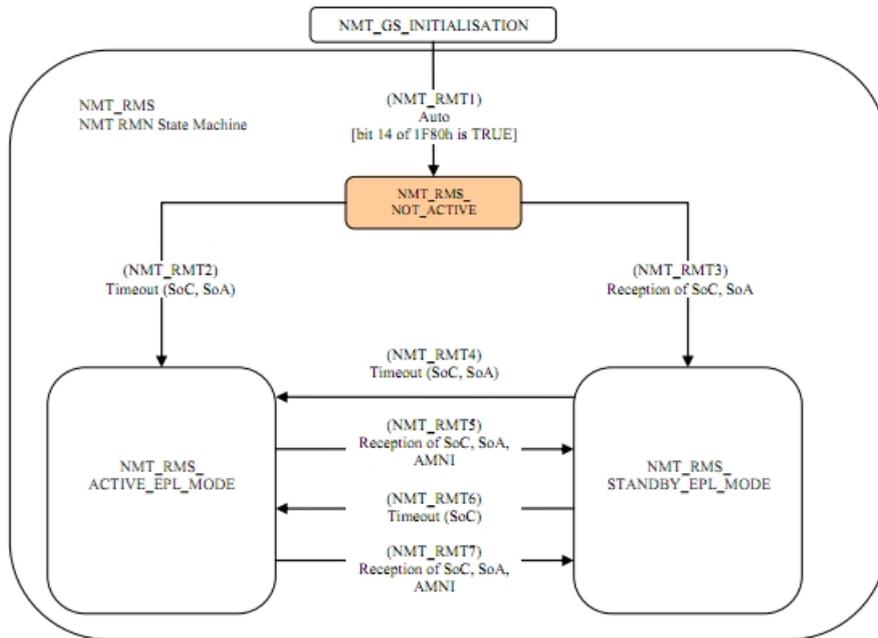
3.4 状态切换

1. 启动 POWERLINK 设备在启动时，根据节点号和 NMT_StartUp_U32 object (index 1F80h)的设置来决定该节点接下来的行为。如果该节点的节点号为 1 至 239，那说明该节点是从站，接下来应该进入到从站的状态机。如果节点号是 240 且 NMT_StartUp_U32 object (index 1F80h)的第 14 bit 为 0，则说明该节点是标准的主站。如果 NMT_StartUp_U32 object (index 1F80h)的第 14 bit 为 1，则说明该节点是冗余主站，进入到冗余主站的状态机。



2. 冗余主站的状态机

冗余主站的状态机主要有两个大的状态，一个是 **active** 状态，一个是 **standby** 状态。当处于活动状态时，该节点作为活动的主站在工作。当处于 **standby** 状态时，该节点处于备用主站。当节点启动完成，进入到冗余主站的状态机时，首先检测是否能在规定的时间内受到 **SOC** 和 **SOA** 数据帧，如果能收到，说明在网络上其他的活动主站，该节点应该进入到 **standby** 状态；否则进入到 **active** 状态。



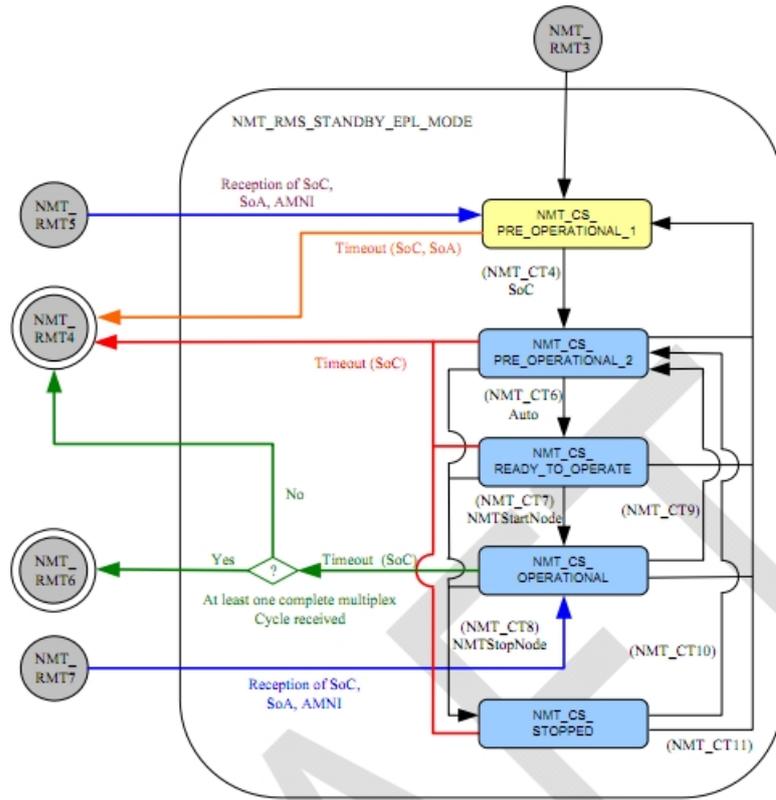
Active 状态:

当节点工作在冗余主站的 Active 状态时，他的作用和标准的主站一样。此外增加了切换到 standby 状态的功能。当该节点收到了 SOC,SOA,AMNI 等消息时，就从 Active 状态切换到 standby 状态。

standby 状态:

当节点工作在冗余主站的 standby 状态时，从活动主站的角度看，备用主站的就像标准的从站那样工作，从活动主站接收 PollReq 数据帧，回复 PollRes 数据帧。此外他还具有如下功能：

1. 定时检测 SOA 和 SOC 数据帧。如果在规定的时间内没有收到 SOA 和 SOC，需要切换状态，接管活动主站的工作。
2. 监控网络上其他节点的状态，备用主站通过监听节点的 PollResponse，StatusResponse，IdentResponse 这些数据帧的信息来得知其状态信息，并将这些信息更新到自己的列表中。在接管活动主站的工作时，需要这些信息，来达到无缝切换。



3. 状态切换

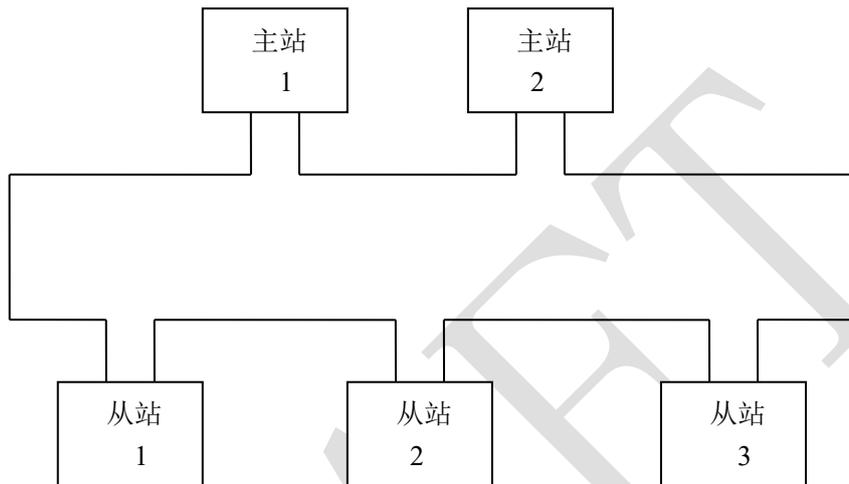
如下所示为状态切换的条件和事件：

NMT_RMT1	<p>Auto [configuration setup completed, Configuration flag is TRUE / start observing network traffic to decide whether to boot as AMN or SMN</p> <p>If NMT_GS_INITIALISATION sub-state NMT_GS_RESET_CONFIGURATION is completed and configuration flag is TRUE, state NMT_RMS_NOT_ACTIVE shall be entered autonomously, e.g. the NMT RMN state machine shall be entered.</p>
NMT_RMT2	<p>Timeout SoC, SoA</p> <p>No SoC or SoA message is received within the time defined by NMT_BootTime_REC.MnWaitNoAct_U32 (see 3.5.1). No AMN is present in the network. Boot as an AMN i.e. NMT_RMS_NOT_ACTIVE to NMT_RMS_ACTIVE_EPL_MODE</p>
NMT_RMT3	<p>Reception of SoC, SoA</p> <p>Reception of SoC or SoA within the time defined by NMT_BootTime_REC.MnWaitNoAct_U32 (see 3.5.1). Start as a SMN i.e. NMT_RMS_NOT_ACTIVE to NMT_STANDBY_EPL_MODE.</p>
NMT_RMT4	<p>Timeout (SoC, SoA⁽¹⁾)</p> <p>No SoC or SoA message is received within the time defined by $T_{reduced_switch_over_MN}$ resp. $T_{switch_over_MN}$. (see 3.5.2) No AMN is present in the network. The RMN NMT state machine is not in NMT_CS_OPERATIONAL state or the multiplex cycle has not been fully received yet, so go to NMT_MS_PRE_OPERATIONAL_1.</p> <p>(1) SoA are only monitored in NMT_CS_PRE_OPERATIONAL_1</p>
NMT_RMT4	<p>Timeout (SoC, SoA⁽¹⁾)</p> <p>No SoC or SoA message is received within the time defined by $T_{reduced_switch_over_MN}$ resp. $T_{switch_over_MN}$. (see 3.5.2) No AMN is present in the network. The RMN NMT state machine is not in NMT_CS_OPERATIONAL state or the multiplex cycle has not been fully received yet, so go to NMT_MS_PRE_OPERATIONAL_1.</p> <p>(1) SoA are only monitored in NMT_CS_PRE_OPERATIONAL_1</p>
NMT_RMT5	<p>Reception of SoC, SoA or AMNI.</p> <p>Reception of SoC, SoA or AMNI within the time defined by $T_{reduced_switch_over_MN}$ resp. $T_{switch_over_MN}$ (see 3.5.2), go to standby mode and stop sending messages immediately. The MN state machine is not in NMT_MS_OPERATIONAL so start from NMT_CS_PRE_OPERATIONAL_1.</p> <p>Application can also trigger this transition on the local node.</p>
NMT_RMT6	<p>Timeout (SoC)</p> <p>No SoC message is received within the time defined by $T_{switch_over_MN}$. No AMN is present in the network. The RMS_STANDBY state machine is in NMT_CS_OPERATIONAL state, so go to NMT_MS_OPERATIONAL</p>
NMT_RMT7	<p>Reception of SoC, SoA or AMNI</p> <p>Reception of SoC, SoA or AMNI within the time defined by $T_{switch_over_MN}$, go to standby mode and stop sending messages immediately. The RMS_ACTIVE state machine is in NMT_MS_OPERATIONAL so go to NMT_CS_OPERATIONAL.</p> <p>Application can also trigger this transition on the local node.</p>

4 冗余系统的典型拓扑结构

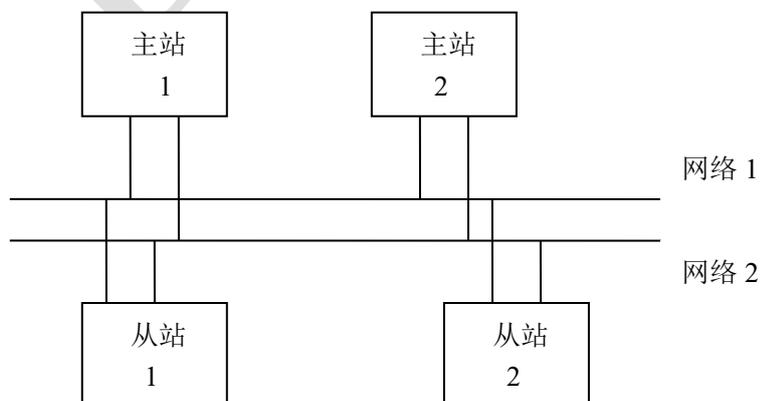
4.1 多主环网

多主环网，即在一个环形网络中，存在多个主站，拓扑如下：



4.2 多主双网

多主双网，即在双网的架构中，存在两个或者多个冗余主站，拓扑如下：



4.3 多主双环网

多主双环网，即在一个双环网架构中，存在两个或多个冗余主站，这种架构的安全性更高，拓扑如下：

