1. **试比较 Spanning tree 与 Steiner tree ?**

**Spanning tree:**生成树，防止交换机冗余链路产生的环路，用于确保以太网中无环路的逻辑拓扑结构，从而避免了广播风暴,大量占用交换机的资源。

**Steiner tree:** 斯坦纳树,连接所有路由器和附加组成员的**最低成本树（**最低成本**生成树）**，计算复杂，新增节点均重新计算，不在实践中用。（NP完全问题）

在多播技术中，每个路由器都可以利用Diskstra算法算出一个组内的最小生成树，即spanning tree，然而不同路由器计算得到的spanning tree并不相同，在这些spanning tree中，链路耗费之和最小的那颗生成树就是Steiner tree。固Steiner tree是多播组中最优的生成树。然而寻找Steiner tree是一个NP问题。因此，现实网络中一般采用基于中心源的算法，尽量寻找一棵接近Steiner tree的最短通路数。

1. **解释DVMRP与PIM ?**

**DVMRP**：距离矢量组播路由选择协议 （DVMRP：Distance Vector Multicast Routing Protocol）, 为互联网络的主机组提供了一种面向无连接信息组播的有效机制。DVMRP 是一个“内部网关路由协议”；适合在自治系统内的使用，不适合在不同的自治系统之间使用。

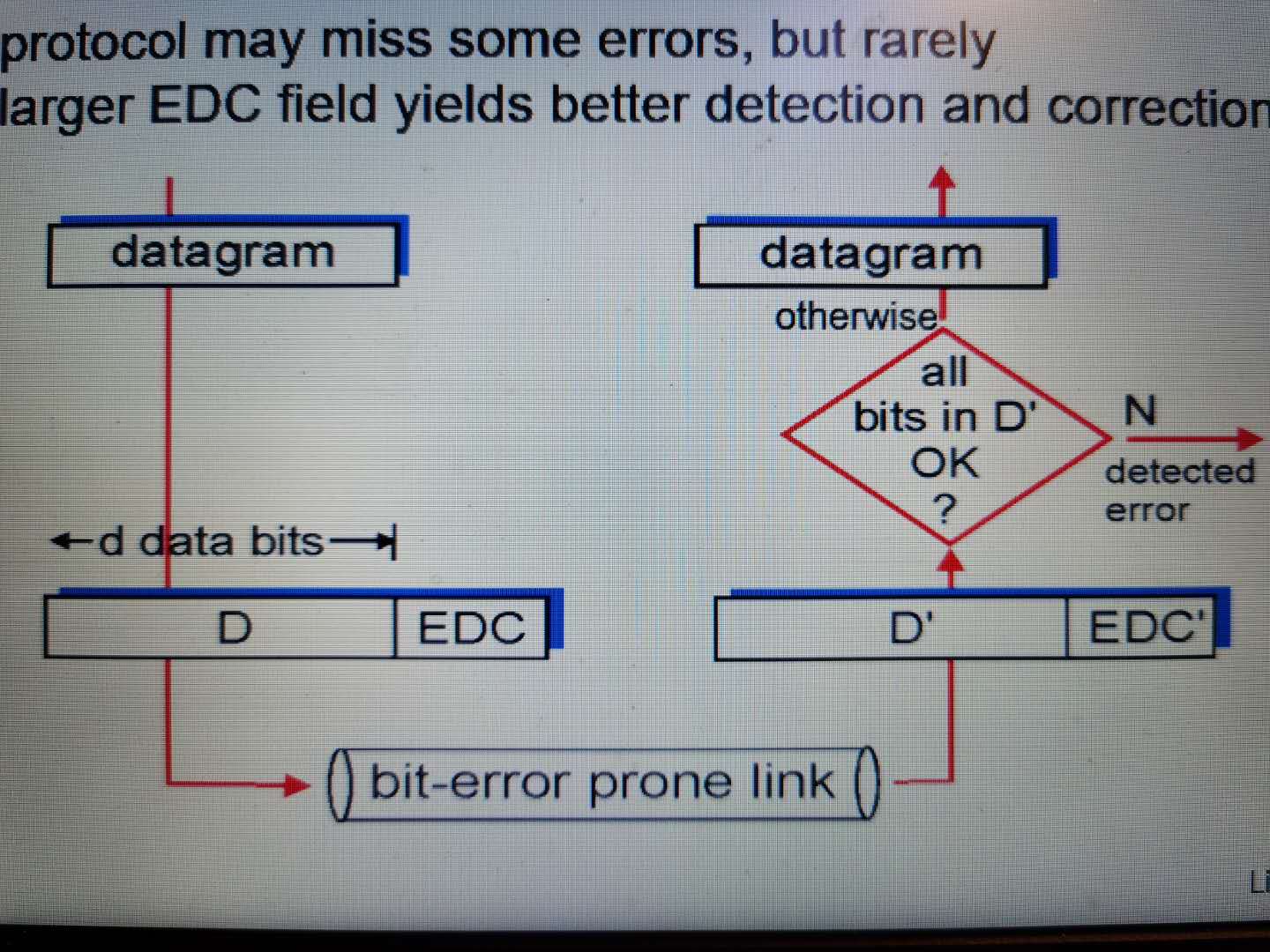
**PIM**：协议无关组播（PIM：Protocol Independent Multicast）。这里的协议无关指的是与单播路由协议无关，即PIM不需要维护专门的单播路由信息。作为组播路由解决方案，它直接利用单播路由表的路由信息，对组播报文执行RPF（Reverse Path Forwarding，逆向路径转发）检查，检查通过后创建组播路由表项，从而转发组播报文。

1. **Data plane 与 control plane 主要完成的功能是什么 ？**

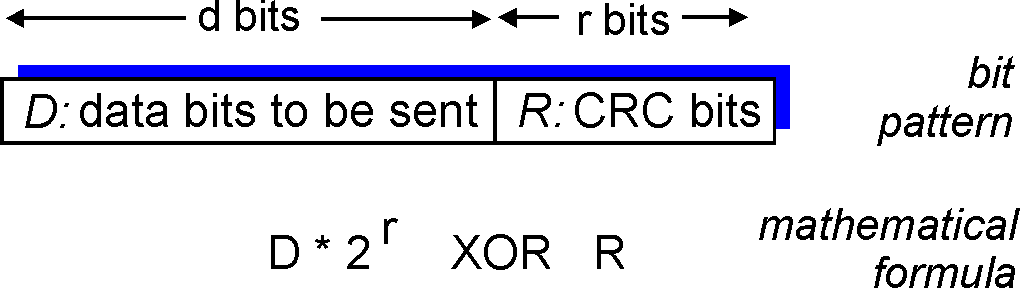
**Data plane**：转发数据是它唯一实现的功能，它决定到达路由器的一条输入链路的分组是如何转发到该路由器的一条输出链路的。数据平面的基本任务是处理和转发交换机各不同端口上各种类型的数据， L2/L3/ACL/QOS/组播/安全防护等各种具体的数据处理转发过程，都属于交换机数据平面的任务范畴。

**control plane**：控制平面负责执行路由选择协议，控制平面用于控制和管理所有网络协议（如生成树协议、VLAN 协议、ARP协议、各种路由协议和组播协议等等）的运行。控制平面提供了数据平面数据处理转发前所必须的各种网络信息和转发查询表项FIB表（Forward Information Base，转发信息库）。

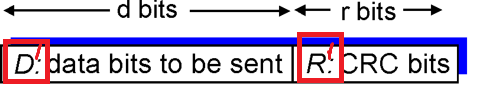
**4. 如下图，解释CRC 工作原理**



循环冗余校验，通过在要传输的k比特数据D后添加(n-k)比特冗余位(又称帧检验序列，Frame Check Sequence，FCS)F形成n比特的传输帧T，再将其发送出去。计算公式：



发送方发送的数据



接收方收到的数据

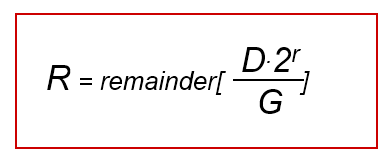
接收方收到数据后，计算*remainder[*D\*2r XOR R / G *]*是否为0，

其中R的计算如下：

如果：D\*2r XOR R = nG

有：D\*2r = nG XOR R

那么得到R的计算公式如下：



用此计算公式计算出R并加入EDC中形成冗余检查位，然后接收方计算*remainder[*D\*2r XOR R / G *]*是否位0。不为0则为“传输出错”。

1. **解释链路层完成的主要功能 ？**

数据链路层最基本的服务是将源计算机网络层来的数据可靠的传输到相邻节点的目标计算机的网络层。

(1)**成帧 （framing）**。在每个网络层数据报经链路传送之前，几乎所有的链路层协议都 要将其用链路层帧封装起来。一个帧由一个数据字段和若干首部字段组成，其中 网络层数据报就插在数据字段中。

(2)**链路接入**。媒体访问控制（**MAC）** 协议规定了帧在链路上 传输的规则。对于在链路的一端仅有一个发送方、链路的另一端仅有一个接收方 的点对点链路，**MAC**协议比较简单（或者不存在），即无论何时链路空闲，发送 方都能够发送帧。

(3)**可靠交付**。当链路层协议提供可靠交付服务时，它保证无差错地经链路层移动每个网络层数据报。

(4)**差错检测和纠正**。当帧中的一个比特作为**1** 传输时，接收方结点中的链路层硬件可能不正确地将其判断为**0 ,** 反之亦然。这种比特差错是由信号衰减和电磁噪声导致的。

**链路建立：**链路层负责在通信的实体间建立数据链路连接，供数据在其上的传输。

**成帧：**链路层利用成帧技术将数据以帧的形式封装起来并进行传输，成帧就是在一段数据的前后分别添加首部和尾部，这样就构成了一个帧。接收端在收到物理层上交的比特流后，就能根据首部和尾部的标记，从收到的比特流中识别帧的开始和结束。

**透明传输：**透明传输其实就是指无论是什么报文都可以传输，非透明传输就是指某些特殊字符不能传输，在计算机网络中，数据链路层将网络层协议封装成帧时，会在首部和尾部分别添加SOH以及EOT这两个特殊字符，接收方是根据这两个字符来确定帧首和帧尾的，如果上层协议发送过来的数据包含EOT，那么接收方在解析这个帧的时候就会误以为数据已经结束，所以，如果链路层对这种情况没有特殊处理，那么就可以理解链路层为非透明传输，但是数据链路层通过对这个字符添加转移符(如果网络层数据中还存在转移符，就再添加一个转移符)的办法来使数据部分可以传输EOT字符，就实现了透明传输。

**帧同步：**为了使传输中发生差错后只将有错的有限数据进行重发，数据链路层将比特流组合成以帧为单位传送。每个帧除了要传送的数据外，还包括校验码，以使接收方能发现传输中的差错。帧的组织结构必须设计成使接收方能够明确地从物理层收到的比特流中对其进行识别，也即能从比特流中区分出帧的起始与终止，这就是帧同步要解决的问题。

**差错控制：**一个实用的通信系统必须具备发现（即检测）这种差错的能力，并采取某种措施纠正之，使差错被控制在所能允许的尽可能小的范围内，这就是差错控制过程，也是数据链路层的主要功能之一。对差错编码（如奇偶校验码，检查和或CRC）的检查，可以判定一帧在传输过程中是否发生了错误。一旦发现错误，一般可以采用反馈重发的方法来纠正。

**流量控制：**流量控制并不是数据链路层所特有的功能，许多高层协议中也提供流时控功能，只不过流量控制的对象不同而已。比如，对于数据链路层来说，控制的是相邻两节点之间数据链路上的流量，而对于运输层来说，控制的则是从源到最终目的之间端的流量

**链路管理：**链路管理功能主要用于面向连接的服务。当链路两端的节点要进行通信前，必须首 先确认对方已处于就绪状态，并交换一些必要的信息以对帧序号初始化，然后才能建立连接，在传输过程中则要能维持该连接。如果出现差错，需要重新初始化，重新自动建立连接。传输完毕后则要释放连接。数据连路层连接的建立维持和释放就称作链路管理。在多个站点共享同一物理信道的情况下（例如在LAN中）如何在要求通信的站点间分配和管理信道也属于数据链路层管理的范畴。