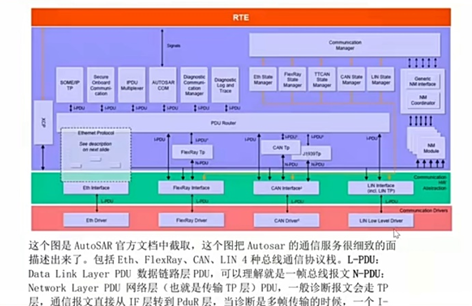
# PDU和收发数据流

当前主流通信总线是以太网、CAN、LIN通信总线，flexray使用较少。最下面是Mcal驱动层，向上是各个总线的if层，再往上，数据汇总到pdur模块。Pdur这一层将pdu进行路由，会根据下层发上来的报文，路由到com或者是dcm模块。通信报文路由到com模块，诊断报文路由到dcm模块。

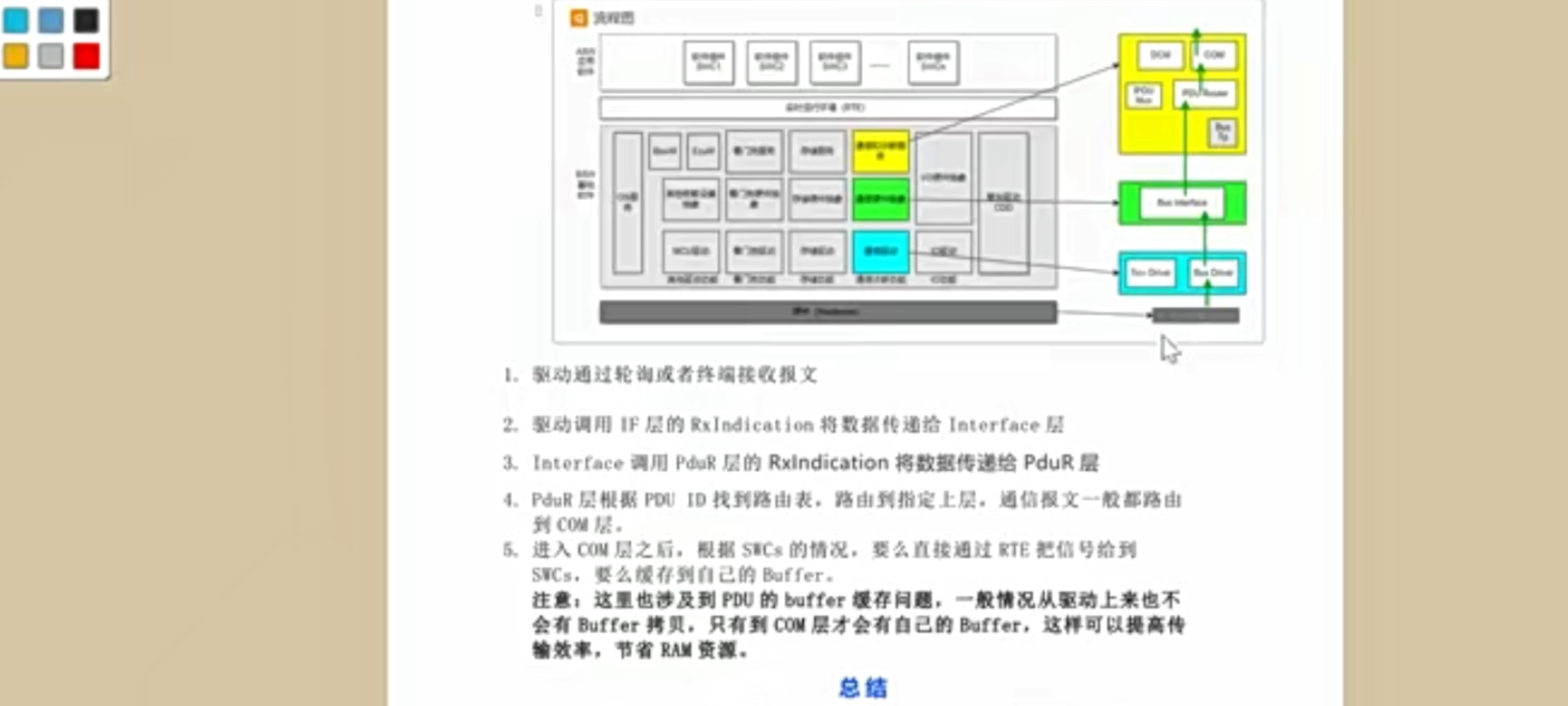


上图是Autosar官文文档中截取的，这个图把Autosar的通信服务很细致的面描述出来了。包括ETH、Flexray、CAN、LIN4种总线通信协议栈。

L-PDU:Data Link Layer PDU数据链路层PDU，可以理解就是一帧总线报文。L-PDU，数据链路层的协议数据单元，加上数据链路层的一些帧头，但是CAN报文不需要加，完整的报文会给到上面。

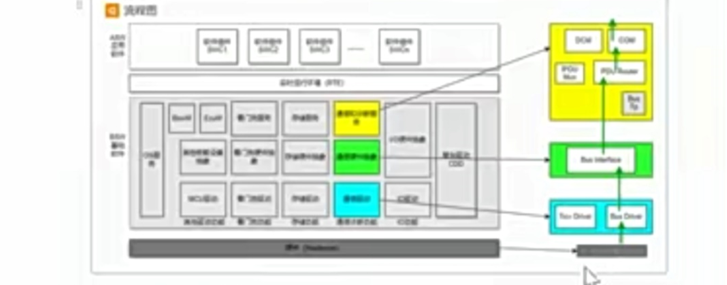
N-PDU：Network Layer PDU 网络层（也就是传输TP层）PDU，一般诊断报文会走TP层，通信报文直接从IF层转到PDUR层，当诊断是多帧传输的时候，一个I-PDU就会被分段成多个N-PDU。N-PDU是被TP层使用的，诊断报文到网络传输层的存在的形式，有可能会加上一些帧头的信息。诊断报文中会有PCI、SID等服务信息。通信报文直接从IF层转到PDUR层，已经处于服务层了。如果是业务报文的话，L-PDU和IPDU其实就是一帧报文，没有什么特别的含义。如是诊断报文，L-PDU和N-PDU有一些不一样，因为诊断报文存在多帧传输，在N-PDU可能需要进行多帧的重组，一次完整的诊断请求，可能有多帧的报文。

I-PDU：Interaction Layer PDU交互层PDU，PDUR路由转发I-PDU。三种PDU代表在通信协议栈不同分层的协议数据单元，I-PDU就包含了数据buffer指针、数据长度、和I-PDU-ID，本质就是一个结构体。



1. 应用层模块通过RTE调用COM模块Com\_SendSignal函数标准接口请求发送信号
2. COM写信号进PDU buffer
3. PDU被事先定义好的PDU路由表。发送到指定目的层(根据PDU ID来查找PDU的路由表)。比如CAN总线的PDU就会被路由到CANif层，LIN的PDU就会路由到LINif层。
4. Interface层根据不同的通道，把报文写入到不同队列中。然后通过驱动发出去。
5. Driver根据报文优先级发送报文。

注意：这里涉及到PDU到buffer缓存问题，一般情况下上层Com到if层都不会有PDU拷贝过程，直接把Buffer缓存指针进行传递。到了驱动层可能会拷贝进入驱动buffer进行发送。这样可以提高传输效率，节省RAM资源。



报文的接收过程：从下至上，从CAN总线向上层的程序去转发。

1. 驱动通过中断的方式来接收报文。
2. 驱动调用if层的RxIndication将数据传递给Interface层。（数据传递过程中并不会拷贝，直接传指针）
3. Interface通过PDUR层的RxIndication将数据传递给PDUR层
4. PDUR层根据PDU ID找到路由表，路由到指定上层，通信报文一般都路由到COM层
5. 进入COM层之后，根据swcs的情况，要么直接通过RTE把信号给到swcs，要么缓存到自己的buffer

注：这里也涉及到PDU的buffer缓存问题，一般情况从驱动上来也不会有buffer拷贝，只有到com层才会有自己的buffer，这样可以提高传输效率，节省RAM资源。

