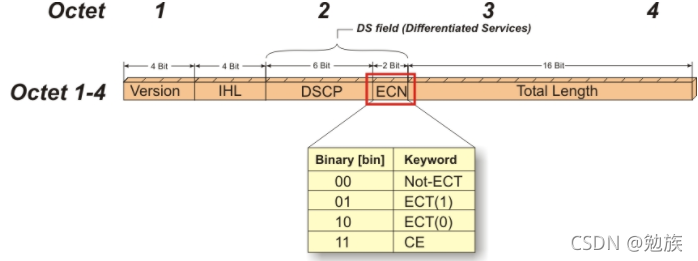
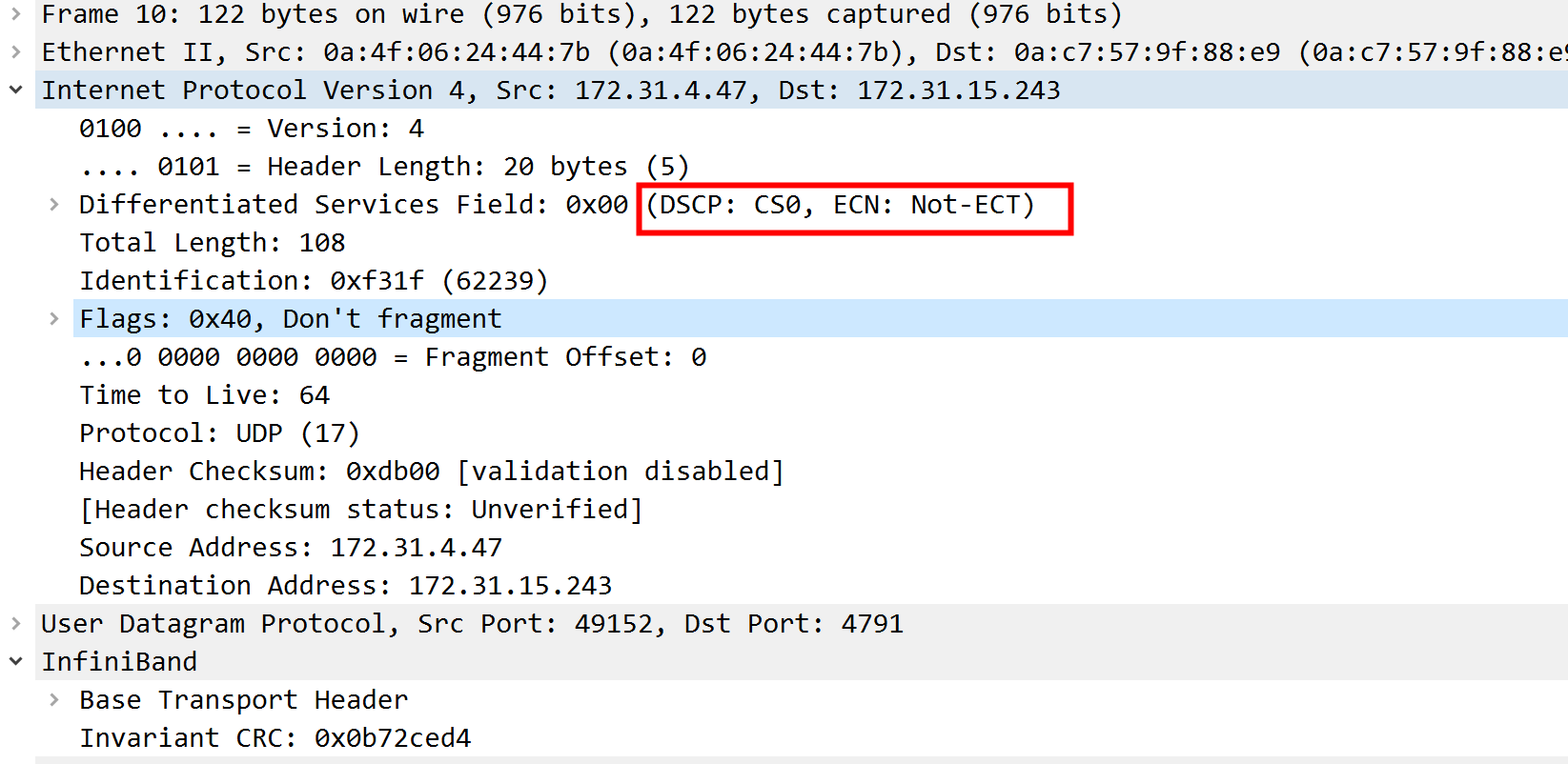
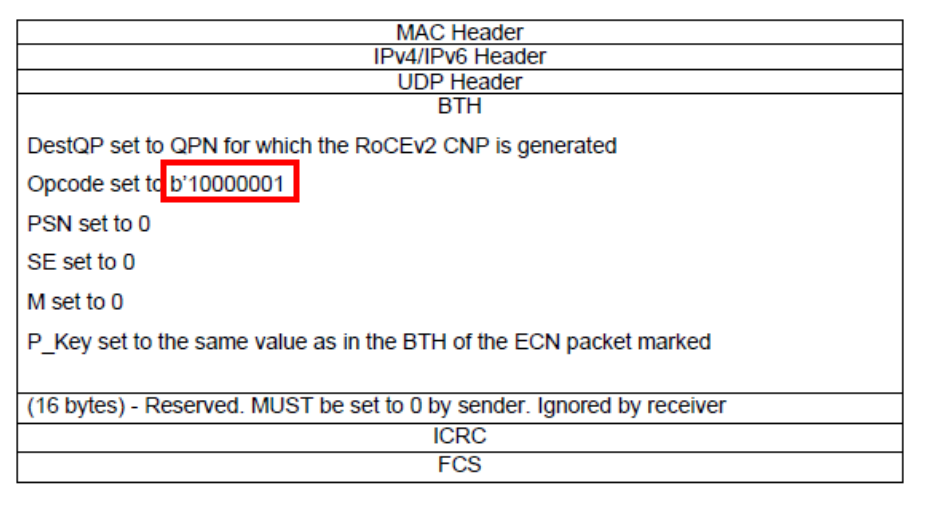
# ROCEV2 CNP

在DSCP的后两位做ECN，Not-ECT “00”表示不支持ECT（ECN-capable Transport）；CE“11”表示发生拥塞；ECT(1)“01”一般是发送端设置，ECT(0)“10”一般是接收端设置（两者也可以互换，也就是“10”在发送端设置，“01”在接收端设置）。其实ECT(0)和ECT(1)两个编码就是一个互相验证的机制。

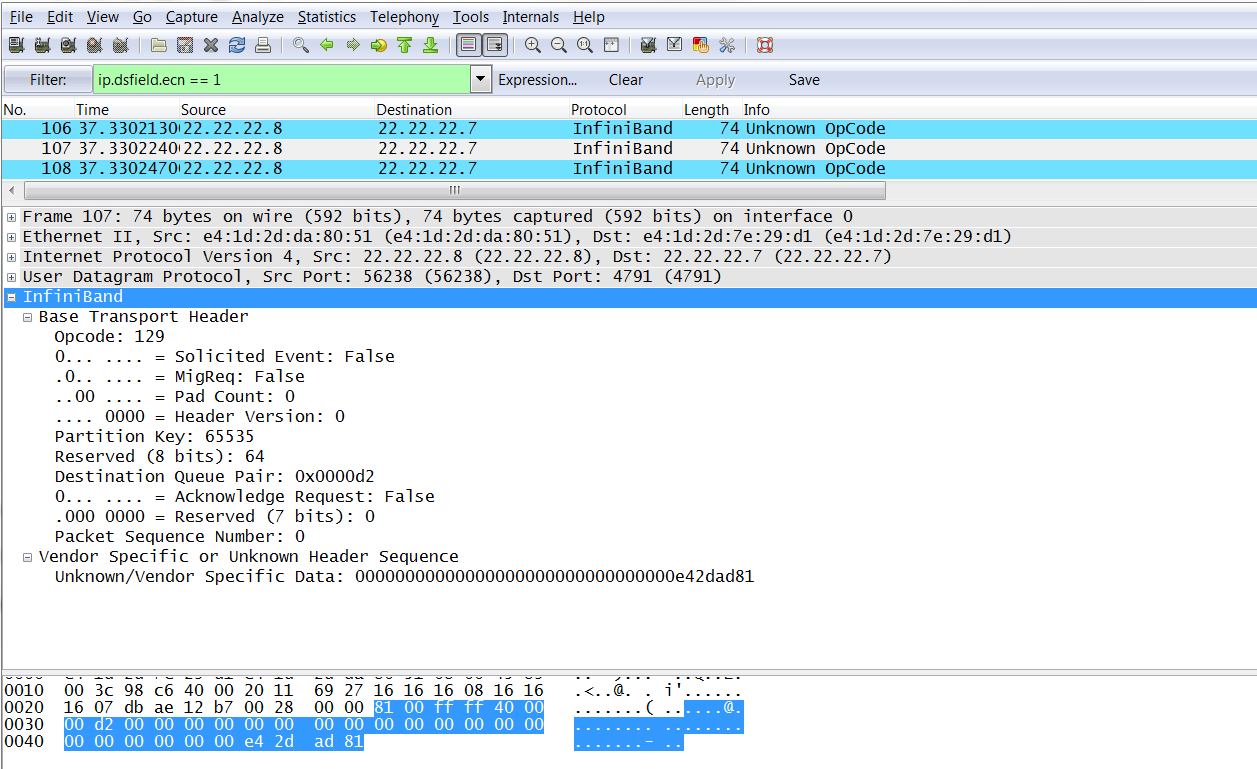




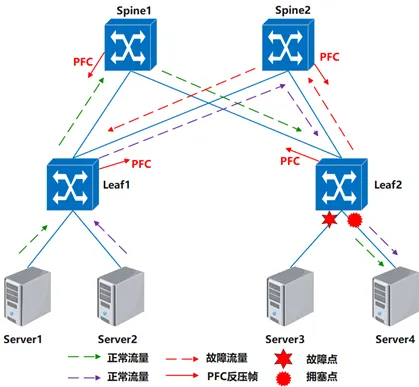
Here is the RoCEv2 CNP Packet format as described in the standard [Annex 17.9.3 RoCEv2](https://cw.infinibandta.org/document/dl/7781" \t "https://enterprise-support.nvidia.com/s/article/_blank):



**Here is a wireshark example of this packet:**



## 拥塞控制



* 稳定性与可靠性：RoCEv2依赖PFC实现无损，会出现PFC风暴与死锁问题，严重损害延迟和吞吐量性能
* 组网规模：DCQCN与PFC支持组网规模节点<=1000

## 拥塞控制产生原因

数据中心场景中，产生拥塞比较常见的两种情况是：

1. 收敛比(总输入带宽/总的输出带宽)

进行数据中心网络架构设计时，多数会采取非对称带宽设计，即上下行链路带宽不一致，当下联的服务器上行发包总速率超过上行链路总带宽时，就会在上行口出现拥塞；

下行可供服务器输入的带宽是48\*10G=480G，上行输出的带宽是6\*40G=240G，整机收敛比为2:1。下行可供服务器输入的带宽是48\*25G=1200G，上行输出的带宽是8\*100G=800G，整机收敛比是1.5:1。也就是说，当下联的服务器上行发包总速率超过上行链路总带宽时，就会在上行口出现拥塞。

1. Incast（多对一）

incast对应多点对一点的流量，会在接收端造成流量突发，瞬间超过接收端接口能力，造成拥塞丢包；

减少incast造成的流量突发，可以从两个方面解决：

发送端减缓incast：发送端通过优良的拥塞算法控制发送窗口与发送速率，从而减缓incast造成的拥塞；

接收端解决incast：将发送侧需要发送的流量转换成从接收侧主动读取流量的方式，这样接收侧能够根据网络状态，高效的实现将发送流量传输到接收侧，解决incast造成的拥塞。

1. ECMP(ECMP构建多条等价负载链路,HASH选择到已拥塞链路发送加剧拥塞）

当前数据中心网络多采用Fabric架构，并采用ECMP来构建多条等价负载均衡的链路，通过设置扰动因子并HASH选择一条链路来转发是简单的，但这个过程中却没有考虑到所选链路本身是否有拥塞。ECMP并没有拥塞感知的机制，只是将流分散到不同的链路上转发，对于已经产生拥塞的链路来说，很可能加剧链路的拥塞。