该文章，比较详细的说明了，网络同步相关

<https://gameinstitute.qq.com/community/detail/121545>

服务器存储的属性发现了变化，才会将Bunch推送。

RPC，<https://docs.unrealengine.com/zh-CN/Gameplay/Networking/Actors/RPCs/index.html>

RPC （远程过程调用）是在本地调用但在其他机器（不同于执行调用的机器）上远程执行的函数。

RPC 函数非常有用，可允许客户端或服务器通过网络连接相互发送消息。

这些功能的主要作用是执行那些不可靠的暂时性/修饰性游戏事件。这其中包括播放声音、生成粒子或产生其他临时效果 之类的事件，它们对于 Actor 的正常运作并不重要。在此之前，这些类型的事件往往要通过 Actor 属性进行复制。

UPackageMap   
生成与维护Object与NGUID的映射，负责Object的序列化。每一个Connection对应一个UPackageMap   
（Packet与Bunch的区别：Packet里面可能不包含Bunch信息）

(PackgeMapClient.cpp )

bool UPackageMapClient::SerializeNewActor(FArchive& Ar, class UActorChannel \*Channel, class AActor\*& Actor)

会在这里面，进行Acotr的序列化！

/\*\*

\* This is the meat of the PackageMap class which serializes a reference to Object.

\*/

bool UPackageMapClient::SerializeObject( FArchive& Ar, UClass\* Class, UObject\*& Object, FNetworkGUID \*OutNetGUID)

UPackage的核心部分，序列化了对对象的引用。

ActorChannel:

在UActorChannel::ReceivedBunch 中接受块数据，在通过ProcessBunch 处理数据块，其中就会进行Actor的反序列化，并为已经生成的Actor赋值属性，如果该Actor没有创建，则会调用到UPackageMap进行创建。

UActorChannel::ReplicateActor() 同步对象信息就是这一个函数了,只是在录制的时候同步给服务器，在播放的时候不会调用这个的。

执行UActorChannel::ReplicateActor执行真正的Actor同步以及内部数据的同步，这里会将Actor（PackageMap->SerializeNewActor），Actor子对象以及其属性序列化（ReplicateProperties）封装到OutBunch并发送给客户端

下面描述客户端是如何接收到服务器同步过来的Actor的。首先客户端TickDispatch检测服务器的消息，收到消息后通过Connection以及Channel进行解析（第二部分已经讲解），最后一步解析出完整数据的操作在UActorChannel::ProcessBunch执行，在这个函数里面：

1. 如果发现当前的ActorChannel对应的Actor为NULL，就对当前的Bunch进行反序列化

Connection->PackageMap->SerializeNewActor(Bunch, this, NewChannelActor);解析出Actor的内容并执行PostInitializeComponents。如果Actor不为NULL，跳过这一步（参考下面图一堆栈）

2、随后根据Bunch信息找到同步过来的属性值并对当前Actor对应的属性进行赋值

3、最后执行PostNetInit调用Actor的BeginPlay。

所以其实，只是在每一次接收到Bunsh的时候，去同步属性而已？

属性复制：



DemoNetDriver.cpp

对Actor进行属性复制，是在在TickFlush的时候，层层调用到ReplicatePrioritizedActors ， ReplicatePrioritizedActor，DemoReplicateActor（在这之中会调用到UDemoNetDriver::UpdateExternalDataForActor，这其中就有使用到FRepChangedPropertyTracker类型的变量），之后会调用到UActorChannel的ReplicateActor

DataReplication.h

FObjectReplicator Obj的复制器, FObjectReplicator::ReplicateProperties 进行属性同步，也是在录制时向服务器同步属性时调用。

## 同步数据蓝图相关：

switch has authority可在蓝图中调用，

此节点检查当前正在运行蓝图脚本的是服务器还是客户端

**Authority**（服务器）还是 **Remote** 机器（客户端）