Packet System:

I use a byte serialisation system to transmit data between clients and the server. This system uses enums to define the type of information that any one packet contains, then though the use of a Packet class inheritance structure, provides processing for that type of packet, and in the case of the server, redistribution of that packet to the other clients on the network.

Packets start on the client as data attached to a game object. This data is translated into a packet that can contain that datatype (ie, TransformPacket for an object transform), then it is serialised using C# MemoryStream(1) and BinaryWriter(2) if it is generated in the server or on a unity client, and if the packet is generated on an unreal client, it uses FMemoyrWriter(5). This produces either a C# array, or an Unreal TArray (6) (depending on where the packet is generated), which is then sent to the server. The server then checks the packets origin (which is stored at position 0 in the packet), before sending the packet to all other clients currently connected.

1. <https://learn.microsoft.com/en-us/dotnet/api/system.io.memorystream?view=net-8.0>
2. <https://learn.microsoft.com/en-us/dotnet/api/system.io.binarywriter?view=net-8.0>
3. <https://learn.microsoft.com/en-us/dotnet/api/system.io.binaryreader?view=net-8.0>
4. <https://dev.epicgames.com/documentation/en-us/unreal-engine/API/Runtime/Core/Serialization/FMemoryReader/__ctor>
5. <https://dev.epicgames.com/documentation/en-us/unreal-engine/API/Runtime/Core/Serialization/FMemoryWriter>
6. <https://dev.epicgames.com/documentation/en-us/unreal-engine/array-containers-in-unreal-engine>