游戏网络技术分析

鄢磊 2020214426

Game	Genre	Network Protocol	Sync Method	Network Topology
StartCraft I/II	RTS	UDP	Lock Step	P2P
Warcraft I/II/III	RTS	UDP	Lock Step	P2P
Dota	RTS	UDP	Lock Step	P2P
World Of Warcraft	MMORPG	TCP	State Sync	Client-Server
League Of Legends	MOBA	UDP	State Sync	Client-Server
王者荣耀	MOBA	UDP	Lock Step	Client-Server
全民超神	MOBA	UDP	State Sync	Client-Server
Doom I/II	FPS	UDP	Lock Step	P2P
Quake I/II/II	FPS	UDP	State Sync	P2P
Counter Strike	FPS	UDP	State Sync	P2P
和平精英	FPS	UDP	State Sync	Client-Server
守望先锋	FPS	UDP	State Sync	Client-Server

问题：为什么他们会采取这样的方案，找找看有什么规律?

1. 网络协议。

从表中可以看出，绝大多数游戏采用了 UDP 协议，只有 WOW 采用了 TCP 协议。

1.1 为什么都采用 UDP：

TCP:

- Connection based
- Guaranteed reliable and ordered
- Automatically breaks up your data into packets for you
- Makes sure it doesn't send data too fast for the internet connection to handle (flow control)
- Easy to use, you just read and write data like its a file

UDP:

- No concept of connection, you have to code this yourself
- No guarantee of reliability or ordering of packets, they may arrive out of order, be duplicated, or not arrive at all!
- You have to manually break your data up into packets and send them
- You have to make sure you don't send data too fast for your internet connection to handle
- If a packet is lost, you need to devise some way to detect this, and resend that data if necessary
- You can't even rely on the UDP checksum so you must add your own

图 1 两类协议的区别[2]

TCP 的优势（保证可靠性、按序等），是建立在一系列开销的基础上。

首先，它是流协议， TCP 构建在 IP 之上，而 IP 构建在数据包之上，因此 TCP 必须将数据流分解成数据包。而一些内部 TCP 代码会对您发送的数据进行排队，当队列中有足够的数据挂起时，它会向另一台机器发送一个包。这对于多人游戏来说可能是个问题，如果你发送非常小的数据包。这里可能会发生的情况是，TCP 可能会决定不发送数据，直到您缓冲了足够的数据，使其生成一个适当大小的包来通过网络发送。而这对于绝大部分的需要实时同步的游戏是一个问题，因为我们希望客户端玩家输入能尽快到达服务器，如果它被延迟，那么客户端的多人游戏的用户体验将非常糟糕。TCP 中有一个 NO_DELAY，可以禁用 Nagle 算法从而立即刷新写入的数据。但此外，

TCP 仍然有许多问题，因为本质上，它建立可靠、有序的连接是基于发送数据包,等待一段时间,直到检测到包丢了,然后重新发送数据包输给了其他机器。重复的数据包在接收端被丢弃，无序的数据包被重新排序。问题是，如果我们要通过 TCP 发送时间关键的游戏数据，每当数据包被丢弃时，它就必须停止并等待数据被重新发送。是的，即使最近的数据到达，新的数据也会被放入一个队列，直到丢失的数据包被重新传输，你才能访问它。在网络堵塞的情况下，重新发送数据包需要大量时间。因此,许多需要玩家实时交互的游戏都基于 UDP 构建网络。课堂上，老师介绍了 RUDP 的方式，来自己进行业务的控制与适应，也说明我们可以基于 UDP，进行针对业务需求的具体适配，从而达到想要的效果。

1.2 WOW 为什么采用 TCP?

《魔兽世界》和许多其他 MMO 游戏可以考虑采用 TCP 用作传输协议。因为 TCP 是可以保证传输的可靠性，从而防止长时间内错误积累后的不一致行为，例如浮点数误差等等。并且其玩法特性让其相对于其他联网游戏而言，对网速有更高的容忍性。并且可以采取前文提到的采用 NO_DELAY 的 TCP 传输等方式，以提高一部分速度。

2. 同步方式

查阅资料后发现，帧同步（Lockstep）与状态同步(State Sync)，有时是可以互换的类型，但在特定的游戏中，使用某种同步方式可以对游戏的通信进行特定的优化。状态同步适用型更广，特别适合复杂度高，延迟要求高，玩家多的游戏，例如图中的 FPS，MMO 等等。帧同步相对适合小兵很多，玩家少且固定，单局时间短，对打击感公平性要求高，追求一致性的游戏，例如图中的 RTS，MOBA 等。

从技术角度来说，帧同步有一些技术限制，比如状态同步的断线重连是把整个场景和人物全部重新生成一遍，各种数值根据服务端提供加到人物身上，而帧同步的断线重连就比较麻烦，例如客户端在战场开始的第 10 秒短线了，第 15 秒连回来了，就需要服务端把第 10 秒到第 15 秒之间 5 秒内的所有消息一次性发给客户端，然后客户端加速整个游戏的核心逻辑运行速度（例如加速成 10 倍），直到追上现有进度。而状态同步有更多的优化手段可以更好的降低延迟感。某种意义上，状态同步的适用范围相对帧同步更广。

	帧同步	状态同步
一致性	**逻辑上绝对一致，天然要求一致性**	**也可以做到绝对一致。**但是一般来说增量式预表现的状态同步会有一些拉扯不同步现象
响应性	操作需要发送到服务器再等服务器返回按帧执行，也可以做预表现但很受限，**响应性较差**	**状态同步可以更好的进行预表现，响应性较好。**因此大多数FPS都采用状态同步
带宽	**人数较少的情况下带宽极低，但随着人数增长带宽几何程度的增加**，例如帧同步的王者荣耀带宽比状态同步的全民超神低很多，但是帧同步无法做MMO类的游戏	可以通过各种方法优化带宽，可以开发千人万人战斗的游戏
延迟	对延迟要求较高，高延迟体验很差	**可以通过延迟补偿，智能预测等优化方式在高延迟下降低延迟感受**
开发效率	**开发非常简单（逻辑只需要客户端写一遍，几乎不需要联调），但是排查BUG非常难。**任何一个错误都有可能导致不同步的严重后果（硬件，随机数，未初始化变量，使用浮点数，渲染帧计算逻辑等等），并且真正问题发生时往往发现不了，要经过一段时间的累积之后才能发现（但反过来说，也是因为这样的要求帧同步的游戏往往bug会更少，有更少的技术债务）	也可以采用使用一套代码的方式，但是因为服务器和客户端机制的不用，以及要加入预测，预表现等开发还是会复杂很多，特别是需要大量两条
玩家数量	**少量玩家。**过多的玩家带宽（但是比较适合同屏大量小兵的情况不需要同步小兵的状态）和客户端性能都无法满足	少量和海量都可以，可以通过分区域，分设备等优化方式优化（特别事服务器硬件可以比玩家设备高很多）
跨平台	**较难**（因为不同的硬件不同编译器等原因导致一些计算不一致，设备不一致特别是浮点数，可以通过使用顶点数来解决，但是要求非常严格测试难度很高）	较容易
外挂	**在服务器也跑战斗逻辑的情况下天然就容易反作弊。**如果服务器不跑逻辑，多人竞技也可以通过投票机制等很容易反作弊。**但是都反不了全图挂，可以参考**：https://zhuanlan.zhihu.com/p/34014063	防外挂难度依赖服务器跑逻辑的比例和开发时是否考虑到反作弊，可以一定程度上避免全图挂。状态同步可以再反作弊上做的更极致因此有些moba游戏平时使用帧同步，线上比赛的时候使用状态同步
中途退出和加入	**难以中途退出和加入。**因为帧同步的机制一般会要求所有玩家从第一帧开始一起计算，包括**断线重连**也非常困难（从第一帧开始追帧）。当然也可以通过定期保存所有玩家快照的方式，但是这个复杂度和风险就更高了**得不偿失。**	支持
优化	**服务器可以做到极低的开销。客户端负载较高需要计算整个战斗的所有运行**（天然要求逻辑和渲染分离，比较容易通过限逻辑帧，负载均衡等方式优化）。	支持更多的优化手段

帧同步与状态同步的优劣[3]

可以看出，状态同步相对于帧同步在许多方面都能做到更好，例如支持的玩家数量，响应性，检测外挂等等，所以有很多类型的游戏愿意使用状态同步，但其中有一个细节值得关注。首先，王者荣耀与全民超神类型几乎一致，却选择了帧同步而非状态同步。我认为从流量的角度，这是非常正确的决定，因为移动网络下，一局流量消耗的多少，是用户非常关心的内容。正是像这样的普遍用户关心的细节做得到位，才能让前者在市场中胜出。可能在现在，已经可以通过增量状态同步、事件同步等方式实现消耗流量更少的状态同步了，但在当时，王者荣耀采用的帧同步方案是一次大胆的尝试，

3. 拓扑结构

可以看到，图中绝大部分的现代游戏，都采用的 C/S 的网络结构。网络结构是与同步方式相关的，例如采用状态同步，一定需要一个中心化的服务器端，来对所有时间进行统一处理，哪怕是用联机的某一个用户的机器，即 State Sync 一定是对应 C/S 的模式。

DOOM,星际，魔兽争霸等游戏采用基于 P2P 的模型，且是较为简单的模式（QUAKE 的连接方式为 C/S，下文讨论。半条命基于 QUAKE 引擎开发，而 CS 作为半条命的 MOD，也沿用了此网络连接方式），其优点在于延时较小。但 P2P 架构则是让信息直接在两个客户端间传递，不需要服务器端。整体的拓扑结构相对于多人联机的拓扑结构显得简单。

而 C/S 的方式，相对 P2P 有诸多优点，简单列举如下：

1. 优化拓扑结构。例如有 100 个客户端，采用 P2P 会有大量的连接，会使得网络非常拥堵，但采用 C/S 的方式，每个客户端只需要负责与服务器端通信。
2. 更容易做状态同步，让服务器承担中心逻辑的决策，维护游戏中各个玩家的状态。QUAKE 的方式即是如此，让服务器负责所有的逻辑判断，客户端只负责渲染。服务器把压缩后的快照发给客户端，而客户端使用这些快照来插值或推导出平滑连贯的体验。甚至在 Quake3 中，客户端不会傻等，而会预测可能的游戏状态，其实预测状态所用的代码跟服务器端的代码是一样的，所以服务器端的状态和客户端的状态往往是一致的。如果确实不一致，则“服务器为准原则”将生效。

参考资料：

[1] Traffic Analysis and Modeling for World of Warcraft

[2] https://gafferongames.com/post/udp_vs_tcp/

[3] https://zhuanlan.zhihu.com/p/104932624

[4] WIKI "Lockstep (computing)". Available: https://en.wikipedia.org/wiki/Lockstep_(computing)[Accessed: 2020-03-24]

[5] MP van Waveren, "The DOOM III Network Architecture" ,2006. http://fabiensanglard.net/doom3_documentation/The-DOOM-III-Network-Architecture.pdf[Accessed: 2020-03-24]

[6] eff S. Steinrnan,“BREATHING TIME WARP” May 1993. Available:https://dl.acm.org/doi/pdf/10.1145/174134.158473[Accessed:2020-03-24]

[7] hristophe DIOT, Laurent GAUTIER, "A Distributed Architecture for Multiplayer Interactive Applications on the Internet", IEEE, 1999. Available: https://www.cs.ubc.ca/~krasic/cpsc538a-2005/papers/diot99distributed.pdf[Accessed: 2020-03-24]

[8] https://zhuanlan.zhihu.com/p/130702310