Multiplayer Networking for Large-Scale Fast-Paced Games

By Blake Bellamy

Computer Science Senior Seminar, April 18, 2020

Outline

- Introduction and background to this genre
- Three of the most common problems on networking for this genre
- Solutions to these problems
- Conclusion

Introduction

- Large scale
- Fast paced
- Popular releases in this genre include PlayerUnknown's™ Battlegrounds, Fortnite, and Apex Legends

Why this matters

- Enormous success of this genre
- Not all of the popular games in the genre were received well for networking related issues
- Network latency (lag) is a common factor that drastically affects the enjoyment of games in this genre
- Howard, Cooper, Wittie, Swinford, and Yang (Fig. 3)

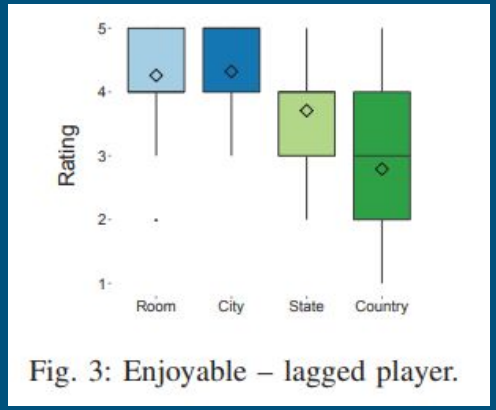


Fig. 3: Enjoyable – lagged player.

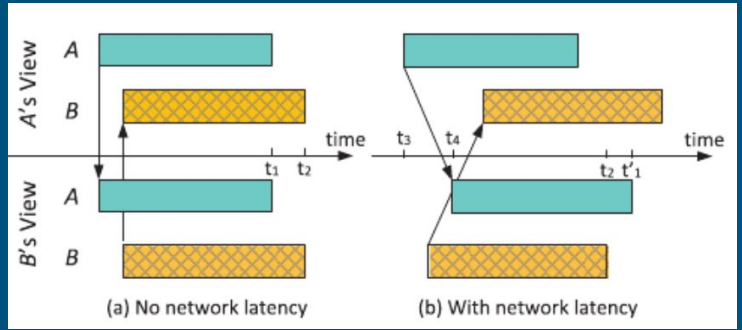
Terms in this presentation

- Action
  - An action is any option a player has in a genre, be it moving, shooting, jumping, etc.
- Latency
  - Physical clock-time an action takes to reach another player or server. More on this next slide.
- Network Lag
  - Often shortened to Lag, and used interchangeably with ‘Latency’.
- Server
  - A computer running the game for other players to connect to and communicate with one another.
- Client
  - A player connected to a server

Problems

Problems: Latency

- Most common of the three, and noticeable
- The actual clock time required for communication to occur from client to server
- “Shot behind cover” situations
- Order of event synchronization

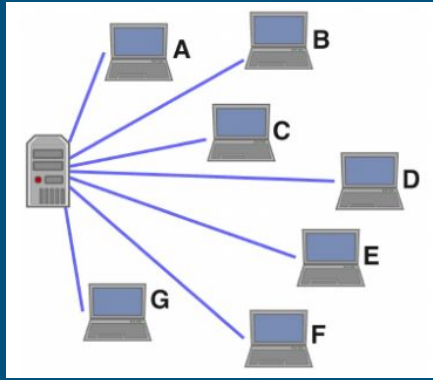


Problems: Workload

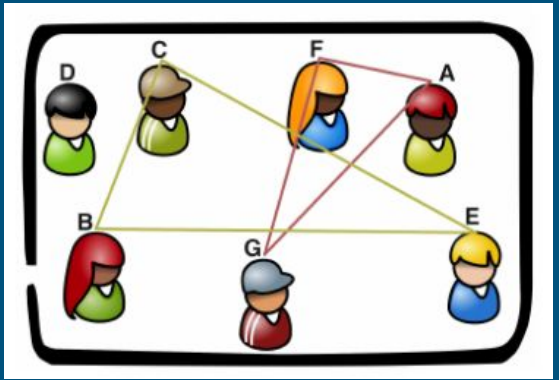
- The amount of processing information done by a server or client.
- Who takes the burden? Client, or server?
- Effects of overburdening a client or server.
- Cheating?

Problems: Infrastructure

- The physical structure of the network
- Actions take time to travel through the physical cables in the ground
- Farther away connections will naturally take longer
- Peer to Peer
- Dedicated Server



Dedicated Server Network



Peer to Peer Network

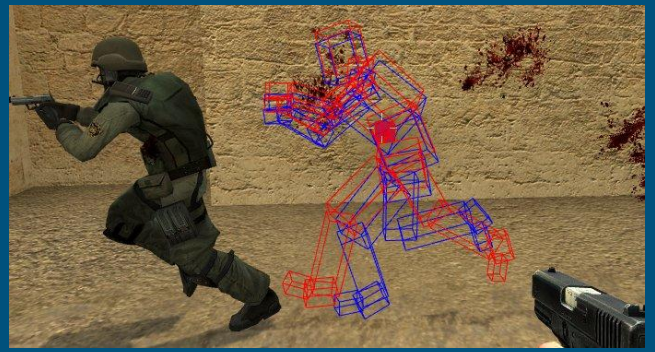
How can we solve these problems?

Solution: Algorithms

- Some problems are out of reach for the developers of these games.
- Developers can mitigate the effects of network-borne issues using algorithms.

Tackling Latency: Lag Compensation

- Traditional Lag Compensation
- Lag compensation minimizes the effect of lag on players with high latency
- Player actions takes time to get to the server, delaying when other players see the action
- By ‘rewinding’ time, we can get a snapshot of what the game looked like at the time the action was sent, not received



A moving player’s actual position (left) versus where the server and other clients see the player (right)

Tackling Latency: Advanced Lag Compensation

- “Shot behind cover” incidents.
- Clients can dispute an action.
- Server can accept a dispute, or deny it.
- Significant drop in “shot behind cover” incidents.

A moving player behind a wall whose position is not behind a wall for other players yet.

Tackling Latency: Advanced Lag Compensation

	TLC			ALC		
Player #	Hits	SBC	%	Hits	SBC	%
1	95	11	11.6%	113	0	0%
2	79	9	11.4%	66	1	1.52%
3	95	4	4.21%	126	2	1.59%
4	100	13	13%	96	0	0%
5	101	16	15.8%	104	1	0.96%
6	82	4	4.88%	92	0	0%
7	93	7	7.53%	97	0	0%
8	89	8	8.99%	88	0	0%
9	101	6	5.94%	92	0	0%
10	76	2	2.63%	77	0	0%
11	125	2	1.6%	103	0	0%
12	85	3	3.53%	85	1	1.18%
Total	1121	85	7.58%	1139	5	0.44%

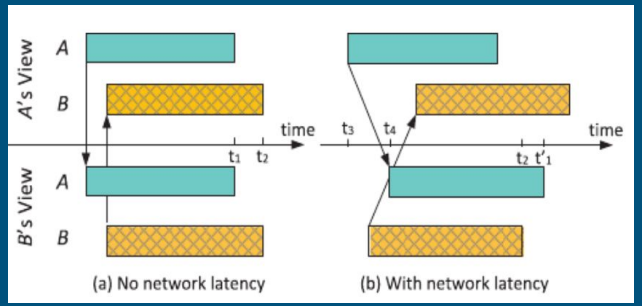
Study conducted by Lee and Chang, in which measured algorithm effectiveness for handling shot behind cover (SBC) situations for both Traditional Lag Compensation (TLC) and Advanced Lag Compensation (ALC).

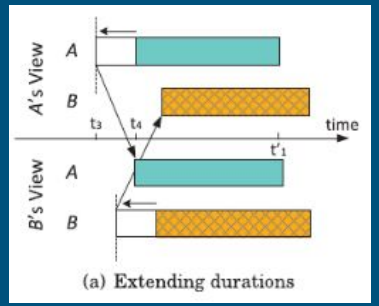
Tackling Latency: Action Synchronization

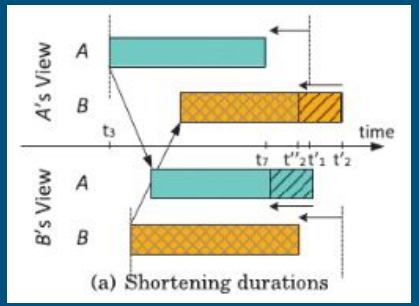
- Latency causes actions to take moving between clients and servers
- Action start time and end time are both influenced by latency
- Actions may arrive out of order

Tackling Latency: Action Synchronization Cont.

- Local Lag - Introduce client-side lag to allow for action to reach other players.
- Perception Filtering - Changing the length of an action
  - Extend a local action's length causing it to start sooner on a remote player's screen while ending at the same expected time.
  - Short a remote action's length causing it to end sooner on a remote player's screen while starting at the same expected time.

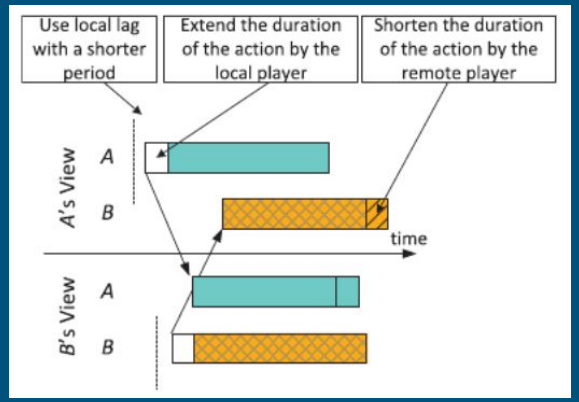


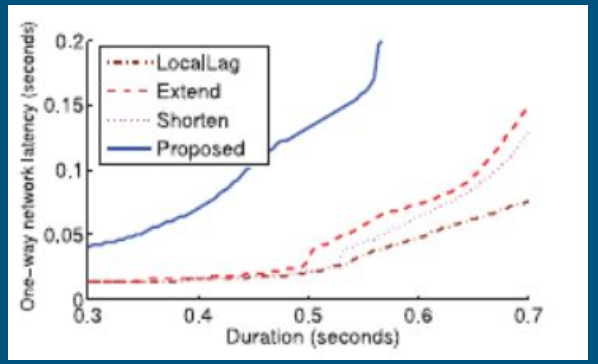




Tackling Latency: Action Synchronization Cont.

- Each on their own are effective
- Combined even more effective
- Trades small amount of extra local-latency for high stability and synchronization in real time.



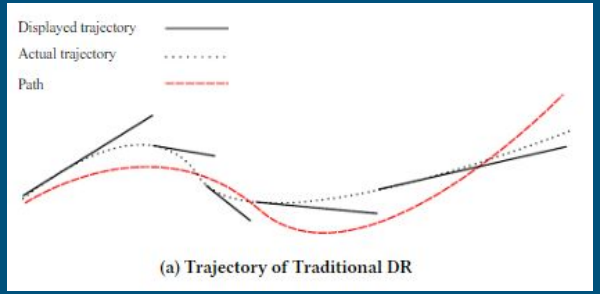


Tackling Workload: Zone splitting

- Hundreds of players connected communicating all at once can bog down a client or server.
- Overburdening may result in slowdowns or crashes.
- Zone adding / Shedding (ZAS) migrates players to less crowded servers.
- Common in Massively Multiplayer Online (MMO) games such as World of Warcraft.
- More than one server can be response for the same world.
- Move freely between zones.

Tackling Workload: Dead Reckoning

- Sending everything about your avatar is inefficient.
  - XYZ Position, Animation, Frame, Velocity, Facing Direction, etc.
- Only send what is needed in ‘ticks’
  - XYZ Position, Velocity
- Allow clients to do rendering using predefined algorithms.
- Allow for more server resources and less network traffic, with only slight more burden on clients.



(a) Trajectory of Traditional DR

Tackling Infrastructure: Hybrid Clustering

- Traditionally, clients connect to one another via one of two network structures
  - Peer-to-Peer networks, in which every player is connected to one another, and sends actions and other messages directly to each other player.
    - Does not scale well into large groups
    - Latency highly based on player's internet, which is usually residential
    - Cheating
  - Dedicated Server networks, in which every player connects to a server, which then is responsible for relaying the actions to the other connected players.
    - Server location matters.
    - Harder to cheat
    - Scales better, but still prone to mass connections leading to slowdowns.

Tackling Infrastructure: Hybrid Clustering

- Hybrid Clustering uses a mix of both.
- Hybrid Clustering works by using topology to group players
- Group players into peer-to-peer ‘pools’ based on in-game participation.
- These ‘pools’ connect to a larger server and communicate only the relevant information

- Pools
  - Player Location, Velocity, Animation, etc..
  - Other Players
  - Local Activities

- Larger Server
  - Information Validation
  - Information Saving
  - World Activities

Tackling Infrastructure: Hybrid Clustering

In larger groups, Hybrid Clustering works well at reducing latency. However, for individuals on their own, it works against them and increases latency.

Table 1: Latency Results [5]

Connections	Latency	New Latency
Common Gameplay	15.2 ±5.1 ms	8.4 ±4.0 ms
Server to Client	7.6 ±3.7 ms	6.3 ±3.5 ms
Area-of-Interest	15.2 ±5.1 ms	18.5 ±7.6 ms

Conclusion

- The key to solving these network-borne issues is to mitigate existing problems.
- Every game desires its own solution, but some are more general than others
  - Advanced Lag Compensation has a spot in all games of this genre due to the effectiveness of the solution with no real drawbacks
  - Action Synchronization may be troublesome to integrate in games where even the slightest of input delay is unacceptable.
- Case by case basis.

Questions?

Thank you for listening