

Methods of dealing with delay on communication of multiplayer mobile game

Group Number: **31**

Shuai Wang 830166

Fengkai Wu 936424

Peng Cheng 880420

Luyang Yang 880199

Abstract:

Mobile game plays an important role in daily life. Real-time multi-player mobile game is a very popular type of mobile game, and this type of game always needs to rely on network communication. The network situation will have a significant influence on the game because of the network communication. Therefore, network delay would be a big problem for this type of game. Network delay is inevitable, but the programmer could do some work to reduce the effect of network delay to games. There are some different ways to reduce network delay effects, such as separate communication thread, selecting different type of communication protocol, using different model to develop the game and more. Each method has strength and weakness. This essay will discuss these methods in a very specific way and make some comparison work on them.

Introduction

Mobile game is a very popular entertainment way for people now. People always spend a lot of time on their mobile device every day, and playing mobile game is a very common way for people to use their mobile device. There are different types of mobile game such as a real-time game, adventure game, combat game and more. Real-time multi-player mobile game becomes more and more popular recently, because of the better performance of the mobile phone and easy access to the internet. Real-time multi-player mobile game means multi-players could take part in the game at the same time, and each player could have their own status. Each player needs to communicate with the game server very frequently to let the server and other players know his own state, such as Player Unknown's Battlegrounds, Arena of Valor and more.

Real-time multi-player game is always time sensitive because most of the operations in the game need to be confirmed by the game server. The confirmed process has achieved the communication between server and client through the internet. If the network situation is bad, the delay will affect the game makes the game becomes lag. It is very bad for the user's experience. The reasons for network delay are diverse. Firstly, this type of game is run on a mobile device. One way for mobile devices to access the internet by WIFI and the WIFI is not stable. If the signal is weak, the communication between server and client would take a long time. Secondly, another way to access the internet is to grab the signal from the base station. This way is more stable than WIFI access, but it is always in low bandwidth. The communication would still take a long time in this way. Thirdly, this type of game always needs to communicate with a server, but the server owned by the company may be attacked by hackers or other competitors. If the server is attacked by others, all players would have a very bad experience of the game. Therefore, the network delay is inevitable.

As the previous paragraph indicates, the network delay is always existing, the game development team could just reduce the effects caused by network delay to improve user's experience on their game. There are multiple ways to reduce the effects caused by network delay. Generally, there are three different ways to solve this problem, which are select different programming methods to develop the game, selecting the different communication protocol for the communication between server and client, and selecting the different network model based on the situation. To be more specific, the programming methods means to use programming skills to develop the game in three different ways. The first one is to build separate communication thread. It means use multi-thread programming method to deal with the different type of events which may happen in the game. This way could reduce the network delay effects on local events like computation and more. The second one is pre-playing the animation. This is a way to make the player knows the operation result immediately with huge network delay. The third one is to reduce the synchronization frequency. This way means to reduce the times for client and server to send a message for synchronization in a short time. When the network situation is bad, the network delay effects would be reduced very

effectively in this way. The communication protocol is the rule for communication between client and server. When the server and client are communicating, the message type, sending way, and the reply way should obey the rule which is pre-set by the development team. There are two popular communication protocols, which are Transmission control protocol (TCP), and User datagram protocol (UDP). These two protocols have their own strength and weakness. TCP can ensure the data security and order but will cost more time. UDP could improve the data transmission efficiency, but the security of message cannot be maintained. Therefore, the development team should select a suitable protocol for the different type of communication. For example, when the player does some buying operation on their account the security should be in high priority, but when the player just does some moving operations the efficiency would be more important. Suitable communication protocol could improve the communication efficiency between client and server to reduce the network delay effects between communication. The third method is to use a different game development model to make the game. There are two different network models for the real-time multi-player mobile game. One is Client-Server (CS), and another is Peer to Peer (P2P). The difference between these two models is to select the different server for players. For a real-time game, the server is very important. The CS model means using a centralized server for all players to synchronize with. The game would be easy for management in this way, but the performance requirement would be very high for a large number of players. The P2P model means select one player's device as the server, and all other players could just communicate with this device for synchronization. This method could balance the latency between each client, but very difficult to manage the system. The game development team should select a suitable one for different situation.

As the last paragraph shows, there are different ways to reduce the effective cost of network delay in real-time multi-player mobile games, and each method has their own strength and weakness. This essay will discuss the specific details of each method and making some comparison between them to give some suggestions on mobile game development.

Methods

1. Programming way to reduce delay

1.1 Separate communication thread

Separate communication thread from other threads is a way to deal with the network work delay in mobile games. Basically, it uses multi-threading programming technology to deal with the network delay in mobile game. Below diagram shows the specific way of how this method works.

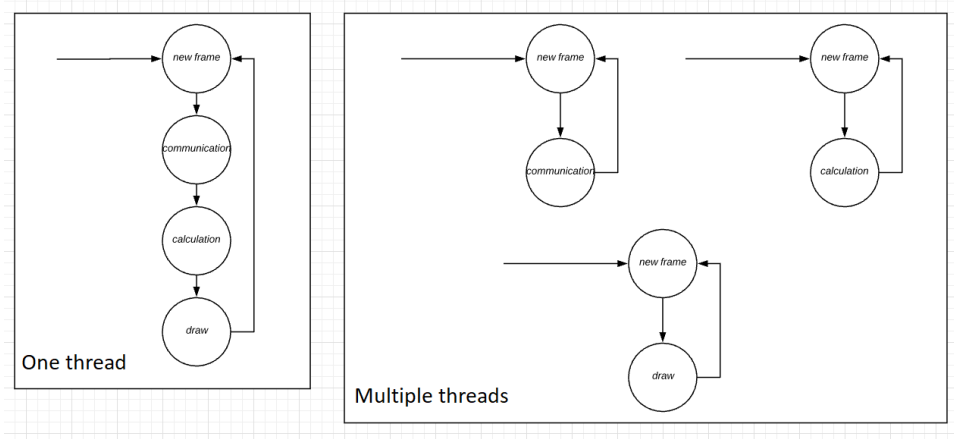


Figure 1 Introduction to Threads

In a real-time multi-player mobile game, the user needs to connect to server to play with others. Some operations need to send some information to server to get reply message, but some are not. If we use one thread to deal with all events in a mobile game, the process would be executed one by one as the left flow chart shows. If the network delay exists, the thread would stop working until it gets the reply message from game server. The calculation and draw process would also be delayed because of the network delay. The user experience would be bad in this way with network delay. If we use multi-threading way to develop the mobile game, the network delay will not affect the calculation and draw process. Because the draw and calculation could be done locally without server. As the right flow chart shows, different thread is used to deal with different type of events. Each event could be done by these threads separately. The communication, calculation and draw could be executed concurrently. If the network delay affects the communication process to do the next event, other process could finish their jobs locally. The network delay will not affect other events any more. The smoothness of game could be improved a lot in this way.

1.2 Pre-play animation

Pre-play animation is a very popular and useful way to reduce the effect of network delay to real-time mobile games. In real-time mobile game, server always need to synchronize each player's status. For example, when the player moves his character in

game, the client need to inform server that he moved from one position to another by sending a moving message. After server receives this moving message, it would update this player's status on server, and send a reply message to client to allow the moving process. When the client gets the reply message, it would do the calculation and draw process locally. Finally, the play could see the moving result on his device. If the network situation is bad, there will be huge communication delay between server and client. When the player does the moving operation, he cannot see any result for a long time because the communication between server and client would take a long time. Pre-play animation is a good way to solve this problem. Pre-play animation means when the player does operations, the client would send moving message to server and does the calculation and draw process at the same time. In this way, the calculation and draw process could be done locally without server's permission. If the network situation is bad, it can only affect the communication process between client and server, but the player won't be affected by this delay. The client does not need to wait for the server's reply message to do the next step, and calculation and drawing processes could be done locally. The result of user's operations could be shown immediately, because the of the concurrently execution of communication calculation and drawing. Users could get better game experience in this way.

1.3 Reduce synchronization frequency

As last paragraph mentioned, for real-time mobile game, the client needs to synchronize its status with server to make the server have each client's status. The synchronization process is achieved by network communication. For example, the player does attack operation, the client would send a message to server to update his status on server. After the updating, the server would send a message back to client to allow his operation. If the network situation is bad, each time when the server communication with client to get his status, there will be huge delay between them. If the client needs to synchronize with server a lot of times in a short time, the network delay will have serious influence on the game. To reduce the effect caused by synchronization, reduce the synchronization frequency is an effective way. For example, when the client needs to synchronize with server ten times in one second, the network delay would affect the game ten times in one second. However, when the client reduces to synchronization times to five in one second, the game smoothness could get lots of improvement. For a real-time game, some operations need to send information to server for synchronization, but some operations like moving, jumping, sitting (has no influence on other players' status) does not. Therefore, reducing the synchronization times is achievable for real-time games. When the game developing team design the synchronization system for game, they could choose different synchronization times for different situations. For example, when the player is just moving and has no effect to other players, they synchronization times could be less. When the player is fighting with players, the synchronization times could be more. Except the network delay, data flow cost is another problem for real-time mobile game. When the client reduces the synchronization times, it would also be data flow saving for the game.

2. Different communication protocol

2.1 TCP & UDP

Another major problem of real-time multiplayer mobile games is caused by the network transmission delay. It means that the information transmission between players and servers takes time. In some cases, the transmission time can be very long, which leads to bad user experiences. For instances, two players A and B are in a racing game. Both of them believe that they first across the finish line. But player A losses because it takes a while until receiving the position of player B. In fact, player B is the one who first across the finish line. Thus, it is necessary to consider what the influences such transmission delay can have on the performance of real-time multiplayer mobile game, what the potential reasons for such delay are, and what actions we can do to reduce the delay. This section will focus on the network transmission delay in real-time multiplayer mobile games.

There are two main internet transport protocols used to transmit data between players and servers: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Both of them are based on Internet Protocol. Thus, whether TCP or UDP, the packet is sent to an IP address. Choose which protocol to send data in a real-time game. We need to have a deeply understanding on the differences between TCP and UDP.

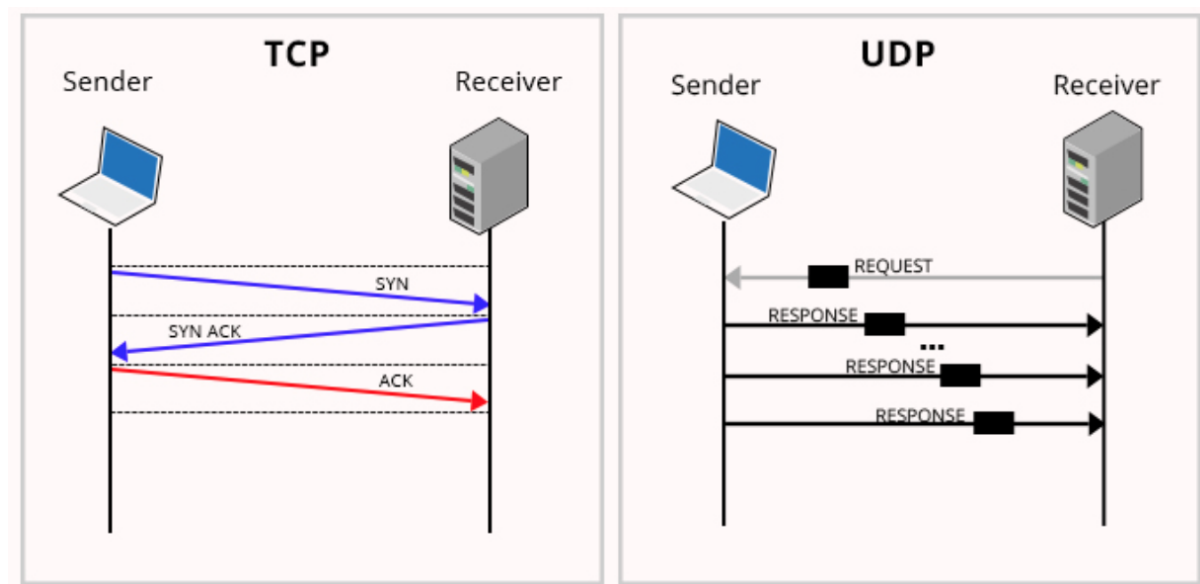


Figure 2 Communication between sender and receiver in TCP and UDP

TCP provides end-to-end, reliable, connection-oriented, ordered packets transmitting services. TCP is able to manage connection over unreliable networks. The connection

between sender and receiver is built before sending packets. Hence, TCP can avoid duplicated packets. To build a reliable connection, three-way handshakes are used. In this setup, the sender sends a connection request to the server, the server accepts its request and then returns the acknowledgement messages, the sender then tells the server it's going to sending data. The three-way handshake must be done to build a connection before sending messages [1]. TCP uses an end-to-end flow control protocol to avoid having the sender sends data too fast for the receiver and processes it reliably. It determines when data needs to be re-sent and stops the flow of data until previous packets are successfully transferred. When a packet got lost, the receiver will check which packet is lost, and let sender resend this packet again. But the packets that arrived early will not be read by receiver until that packet arrives. By this scheme, the arriving order of packets can be guaranteed.

UDP is a connectionless, unreliable, unordered protocol. It provides a way for applications to send encapsulated IP datagrams without building a connection in advance. The only thing that UDP does is sending packet. Applications are required to implement their own protocols on top. Moreover, there are no flow control and error control in UDP. When a packet got lost, UDP has no such flow control and error control to check and find out which packet is lost. Thus, it will do nothing. That's the reason some important data in games will not be sent using UDP. On the other hand, it can be a strength. UDP was born for real-time transmission, which provides fast data transmission. The reason UDP is faster than TCP is because there is no form of flow control or error correction or building connection. The data sent over the Internet is affected by collisions, and errors will be present. Remember that UDP is only concerned with speed.

Transmission delay can be caused by TCP. First of all, as long as a player or sever needs to send data, the connection must be built using three-way handshakes before starting transmission. In fact, this process takes a while before data exchanging which is we care about. It causes some delay. Second of all, if a packet is lost, TCP must stop receiving data to find out which that packet is and ask sender resends such packet. TCP will come back to receive packets until such packet successfully arrives. In this case, TCP takes some time to resending that packet, which causes delay. Real-time multiplayer mobile game requires the communication between every player and server should be fast. Therefore, the recommendation is using UDP for real-time game protocol to send and receive data.

3. Different Network model

3.1 Client – Server Network model

Client – Server network model has been implemented for many multiplier online games. In this model, the whole network system is controlled by several authoritative servers.

If a client wants to perform an action, for example, the client would like to move the character's position in game, this information or request will send to the authorized servers. The server will check whether the request is available, based on the relevant principles. If the request is correct, then the server will commit the request and update the global game state and send this state to all the other clients to guarantee synchronization [2].

By using of client – server model, the network delay is mainly depended by the network bandwidth and the server's capability. A high-quality network bandwidth can ensure the speed of information transferring between client and server. Generally speaking, the client will get a low ping value during playing game. On the other hand, the authoritative servers should have large capabilities in order to provide service for as many clients as possible at the same time. In multiplier online games, it is significant to held well-performed servers to provide stable and fluently gaming experience.

The advantages by using of this model are stable connection, low bandwidth requirement and more secure. Firstly, most of the game resources and core files are stored in the central servers. If one of the server crashes, it is easy to manage and restore data from the backup database server. In this case, client-server model could provide a stable gaming service for client. Secondly, client does not have to use a large bandwidth network, which is very expensive. Because the main services are provided by the servers, and these servers are distributed in different places. If a client could communicate with its closest server fluently, the client could get a satisfied gaming experience with low latency, even if the game is published in different country or platform. For instance, World of Warcraft is a global popular game for years, it has hundreds of servers locate at many countries. Therefore, every player could have a good experience with low latency. Finally, hackers are existing in game for years. They could let the opposite player experience high latency gaming experience in order to get the victory, especially in multiplayer online battle arena (MOBA) game. For example, in Dota2 hackers could force the opposite player to lost connection or having a high ping during playing. One of the effective methods is to defend is to use client – server model for the game. In this model, all of the core files and data are stored at server side, clients cannot change the data locally if they want to play. Servers could verify the data send from client, if a client is changing some of the data file for cheating, the server will perform deny of service (DOS). Therefore, hackers could not play the game anymore and get punishment. All in all, client – server model could provide stable gaming services, low bandwidth requirement and secure gaming environment to protect the internet communication as well as reduce the latency.

On the other hand, the drawbacks of client – server model are bottleneck of service, traffic congestion and servers may break down [3]. First of all, in order to provide a satisfied service for the client in online game, the quality of servers is getting significant. However, the costs of well – performed servers are extremely expensive. There is a trade – off between the costs and performance. Most of online game producer do not

want to spend lots of money for renting or holding up high quality servers. Therefore, there could be a bottleneck for the server's capability. If the client number hit the bottleneck, the latency problem will raise up. Furthermore, because the client – server model is using authoritative servers to provide main service, if a large number of player login the game at the same period, or performing guard fighting in a huge scale, the traffic congestion problem will come up. Traffic congestion is a common problem for massively multiplayer online role-playing game (MMORPG), when lots of players get tother and perform actions, the player will feel lagging and highly latency. That is because at this period plenty of data are send to the server at the same time, because of the bandwidth and the capability [3]. Servers cannot deal with information quickly, thus the latency occurs. Finally, if the authoritative servers are broken down at the same time unfortunately, there is no services provided anymore.

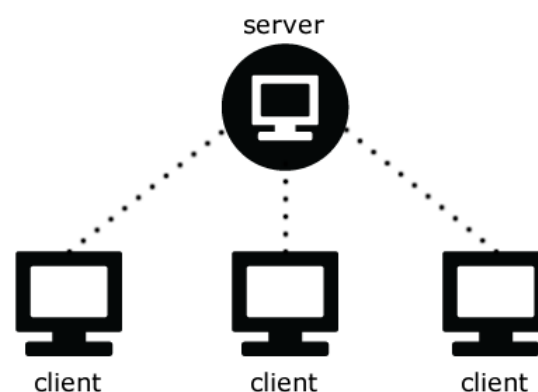


Figure 3 Client - Server model

3.2 Peer-to-Peer Network model

Unlike client-server model, peer-to-peer (P2P) network is more convenient to set up and use. In P2P network model, one computer could perform as a server and let others to join. These computers could communicate with each other through application directly and store their files locally.

In local area network, there is almost no latency in P2P game. The client – server does not have such a good performance. Because one computer is performed like server, thus in online multiplayer game this computer will not have any latency either.

The advantages of peer to peer network are balancing latency and no latency in some circumstances. In P2P network, the host will have no latency as playing the game, while the other players who joint the host will have different degrees of latency, based on the internet bandwidth. It is clear unfair that the joined players have latency while the host does not. Therefore, in P2P model, game producer could force the host to experience some certain of delay to synchronise with others through coding. In this case, all of the player will experience almost same game environment in a low latency. On the other hand, P2P network model has a fantastic performance on local area network. Because there is no need for internet in local area network, the latency is reduced significantly.

Therefore, if a multiplier online game is hosted on a local area network, peer-to-peer model is the best choice.

The disadvantages of peer-to-peer network are obviously. This model requires the player has a large internet bandwidth in order to reduce the balance. If a host does not have a high-quality internet environment, the joined player will experience extremely bad latency. On the other hand, because of there is no dedicated servers, if the host shutdown suddenly, the joined player will lost connection immediately. In this case, the gaming experience will be affected a lot.

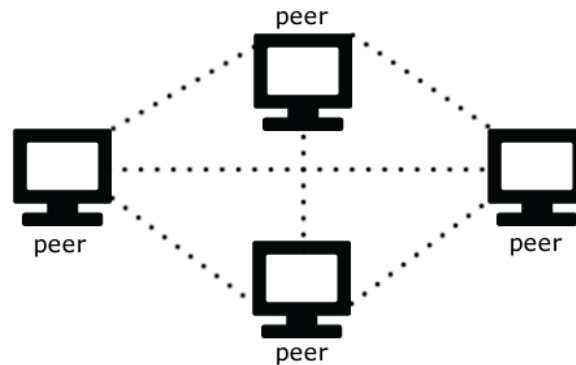


Figure 4 Peer-to-Peer model [4]

Conclusions and future directions

In conclusion, it is no doubt that mobile game will getting more popular and significant to our life in today as well as the future. While playing a real-time mobile game, a little period delay will bring a negative gaming experience to the player. That is to say, the latency problem is the first element that will affect the player's gaming experience. Online gaming players are more likely to quit a game when they feel the network delay over 500 milliseconds [5]. Therefore, how to optimize the gaming experience is the most important thing to consider. While playing a real-time game, the precondition is that the client should not feel latency as less as possible in order to enjoy the content. In this essay, it has indicated three general methods or directions to reduce the latency issue as much as possible. The first one is to reduce the latency problem at programming level. Through different programming methods to bring a fluently gaming experience to player. Besides that, in network layer, using different communication protocol, TCP or UDP, for different types of real-time mobile game will also optimize the network connection between client and server, which will lead to a low delay gaming environment. Furthermore, by using of different game model based on the situation could also reduce the real-time latency problem.

Mobile game has proved that it was occupied an important position in electronic gaming market. And mobile game will bring a bunch of profits to the gaming industry. In 2015, mobile game had earned over \$29 billion to the gaming industry. In 2018, it is predicted that about \$45 billion potential profits the mobile game will make for the gaming company [6]. In the future, with the development of technology, there will be more

kinds of interesting mobile games published. To remain competitive for a gaming industry, how to let the client enjoy the mobile real-time game thoroughly will be getting more significant. A low latency environment is the baseline for every mobile real-time game, it will decide whether the player could enjoy the game content or not. The purpose of this essay is to introduce several effective methods to reduce the mobile real-time latency. In the future, these methods will be used widely to produce a outstanding mobile real-time game.

Reference List

- [1] Tanenbaum, A. S., & Wetherall, D. (1996). *Computer networks (ed.)*. ISBN: 0-12-066102-3.
- [2] TechTerms. (2016, June 17). *Client-Server Model Definition*. Retrieved from https://techterms.com/definition/client-server_model
- [3] Neumann, C., Prigent, N., Varvello, M., & Suh, K. (2007). Challenges in peer-to-peer gaming. *ACM SIGCOMM Computer Communication Review*, 37(1), 79-82.
- [4] Fernando Bevilacqua. (2013, Aug 12). *Gamedevelopment Building a Peer-to-Peer Multiplayer Networked Game*. Retrieved from <https://gamedevelopment.tutsplus.com/tutorials/building-a-peer-to-peer-multiplayer-networked-game--gamedev-10074>
- [5] Pete Mastin, Cedexis. (2016). *How latency is killing online gaming*. Retrieved from <https://venturebeat.com/2016/04/17/how-latency-is-killing-online-gaming/>
- [6] Tricia Duryee. (2015, May 4). *Mobile gaming sales to surpass console gaming sales for first time this year*. Retrieved from <https://www.geekwire.com/2015/mobile-gaming-revenue-to-surpass-console-gaming-sales-for-first-time-this-year/>