Stefan Grulović (20150280)

Hirstos Hristodoulou

Computer Science 325: Distributed Applications

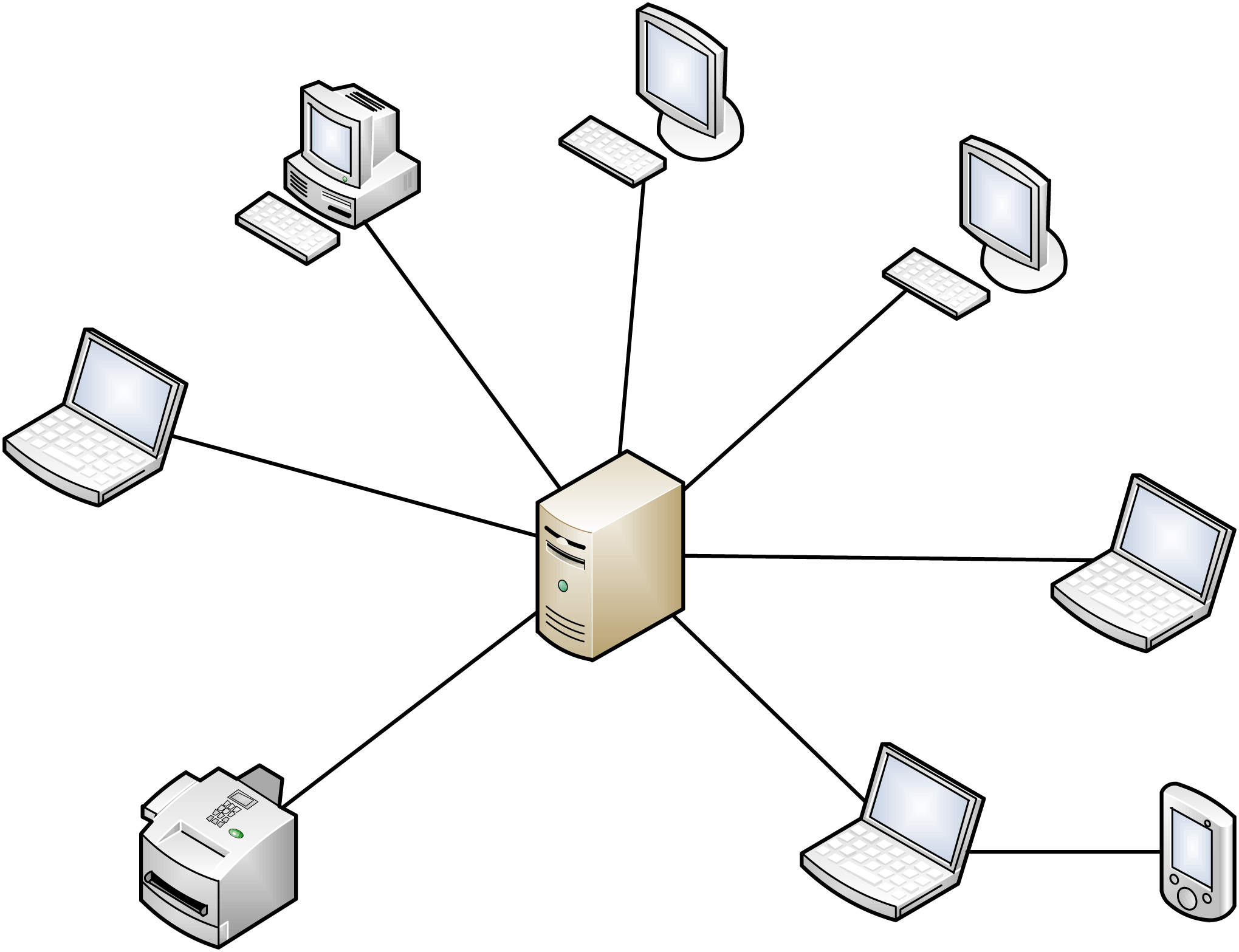
Spring 2018

**Distributed applications and Massively-Multiplayer Online Games (MMOs)**

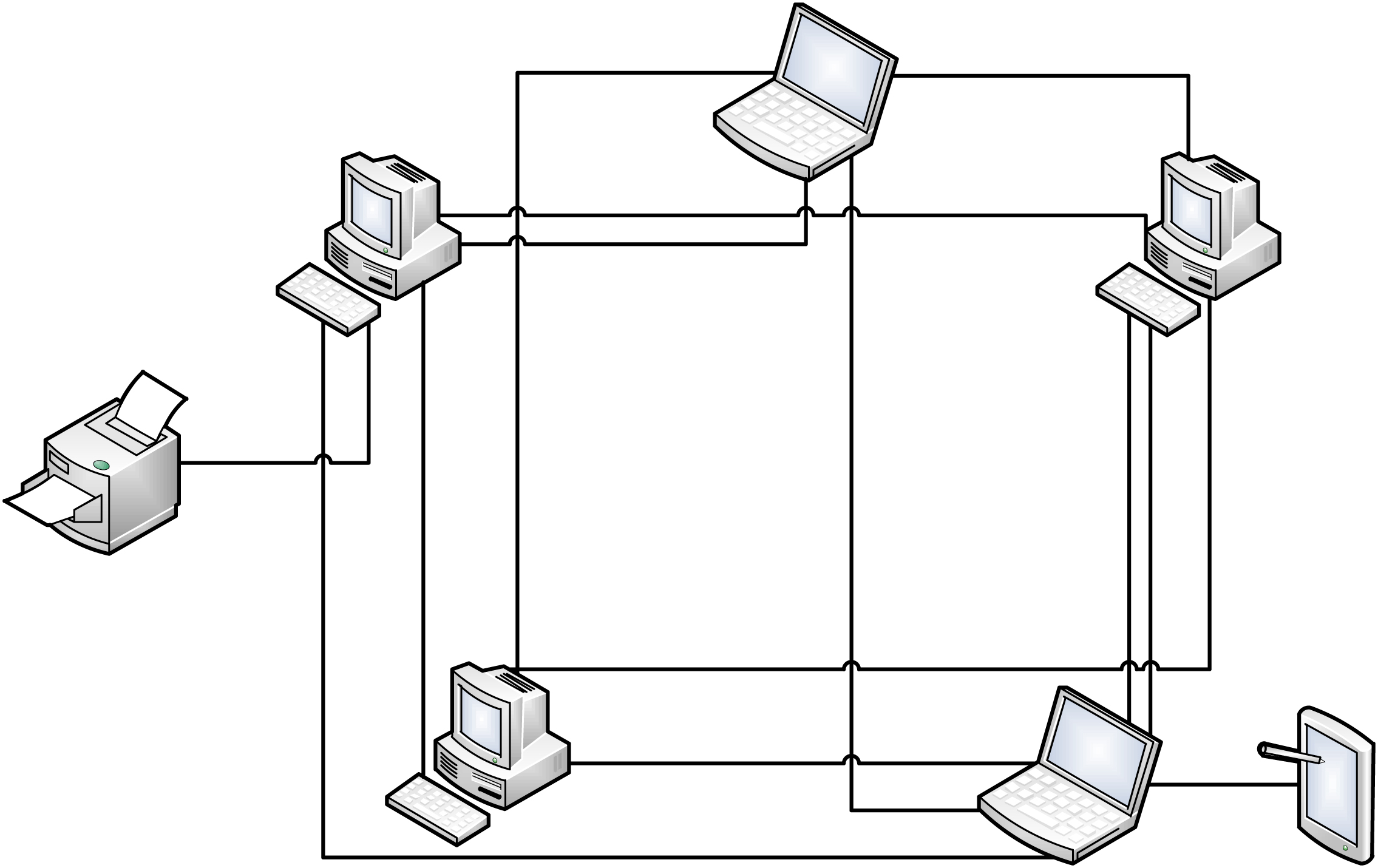
Distributed application is software which runs on multiple machines in order to achieve a common goal [7] [9]. The machines are connected through a network creating a link between the server and client [3]. Example of such distributed application are massively multiplayer online games (MMOs) MMOs are internet based video games which accommodates hundreds to millions of players across the globe [2]. Run indefinitely meaning they have no end they run until the servers run [4]. The player can join or quit at any point, constantly saving the latest state of the player. Such as so whenever they return they return from the last state they left, creating constantly running virtual world [4]. Also, the main challenge on creating a MMOs are keeping up with the growth of players and requirements of hardware and software approaches to fixing in dealing with those numbers [5] [6].

Creating a scalable game architecture which would run well for every player is hard, as it needs to assist large numbers of players requiring a short response time for their actions in order to see a consistent virtual world [10]. Three main architectures of game server are client-server, peer-to-peer, and hybrid of both [10] [6].

Client servers are the most commonly used, the architecture has a main server which is in charge of accommodating all user requests and providing a result which in return updates the virtual world [10]. This allows for more security as the response is generated on the server end rather than the client preventing user from meddling with the logic of the game. There may be more than one server and or machine which can balance and divide the load among them [10]. Server is responsible for creating a response, however a cluster of user could create localized congestion slowing down the server. Plus, the server and its response is limited by its computational power allowing only a specific number of players [4]. Client-server architecture therefore is not scalable as it has a one centralized system which is bounded to its computational power and if the server crashes all players lose connection [10]. Furthermore, making it harder to achieve a fluent, virtual world, which MMOs require [1]. Client-server is preferable for a small number of users as it allows more security due to response being generated only on the server side, and overall performance is on the server while the front end may be on user providing fluent experience [8]. A solution to the limitation could be having more servers, disk farms and faster internet, however this solution is expensive.



Peer-to-peer architecture incorporates the server logic on to the client side, allowing to overcome the server only computation and storage increasing scalability [4] [6] [10]. Alongside it is also cheap to employ and distribute pushing scalability even further [11]. Moving the computation requirement, processing power and storage capacity on to the user’s CPU and network, contributes to the overall server power and fluency [10] [4]. Direct connection between users reduce the delay of response among users [4] [11]. The world can be also divided into sections which can making a specific user responsible for a small region dividing also the computation load [1]. However, with server logic distributed to the user, the user is put into a position from which they can create their own virtual world and bypass the monthly subscription which is the way the games manage to run and continue to evolve [10]. This also opens the door to the user’s ability to cheat and additionally makes it harder to handle such actions from the developer’s side [6] [10] [11].



Combination of client server and peer-to-peer is possible, in this architecture usually the events and situations are divided among the server and client [6]. Critical events are handled by the client-server to prevent cheating and peer-to-peer to lower the bandwidth but yet still keep a controllable environment [6]. Such architecture is harder and more expensive to employ as it requires more knowledge, planning and resources in order for everything to work properly.

The main complication of MMOs are that they require a constant and fluid experience in order to achieve a believable virtual world making response time essential to the equation [10]. Some of the software approaches in achieving this is zoning, sharding, instancing and cloning [10]. Zoning splits the world into geographical regions where each one is hosted on its own machine and or server. However, user may encounter waiting time when switching zones and it can vary dramatically depending on the total player allocation among the world [10]. Sharding creates duplicates of the world but groups the users rather than the world. However, there is no communication among the groups of players limiting the connectivity among the players by not allowing players to play together [10]. Instancing has multiple zones which can be on different servers and each zone has smaller amounts of players. However strange situations can occur, for example if two players want to meet at a certain location but are on a different player group, it will result in not seeing each other [10].

To conclude, the rates continue to increase across the globe as popularity catches on within players and so does the demand of the computing powers of MMOs. The client-server architecture will have difficulty in keeping up with the increase in users as the fixes can get quite expensive, further introducing a peer-to-peer solution. This development is more efficient however protection becomes an issue because server tools are available to client. And Hybrid is complex to implement and develop needing further research to make more accessible.

**Bibliography**

1. Assiotis, M. and Tzanov, V., 2005, May. A distributed architecture for massive multiplayer online role-playing games. In *NetGames' 06 Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games, Article* (No. 4).
2. Chu, H.S., 2008. Building a Simple yet powerful MMO game architecture. *Verkkoarkkitehtuuri. Part*.
3. Docs.oracle.com. (2018). *What Is a Distributed Application*. [online] Available at: https://docs.oracle.com/cd/E13203\_01/tuxedo/tux71/html/addist2.htm.
4. GauthierDickey, C., Zappala, D. and Lo, V., 2004, April. A Distributed Architecture for massively multiplayer online games. In *ACM NetGames Workshop*.
5. Hampel, T., Bopp, T. and Hinn, R., 2006, October. *A peer-to-peer architecture for massive multiplayer online games. In Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games* (p. 48). ACM.
6. Jardine, J. and Zappala, D., 2008, October. A hybrid architecture for massively multiplayer online games. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games* (pp. 60-65). ACM.
7. Momoh, O. (2018). *Distributed Applications (ĐApps)*. [online] Investopedia. Available at: https://www.investopedia.com/terms/d/distributed-applications-apps.asp [Accessed 25 Mar. 2018].
8. Sanati, J. (2011). *Top 10 Reasons to Setup a Client-Server Network - IT Peer Network*. [online] IT Peer Network. Available at: https://itpeernetwork.intel.com/top-10-reasons-to-setup-a-client-server-network/.
9. Techopedia.com. (2018). *What is a Distributed Application? - Definition from Techopedia*. [online] Available at: https://www.techopedia.com/definition/23971/distributed-application [Accessed 25 Mar. 2018].
10. Turchini, G., Monnet, S. and Marin, O., 2015, September. Scalability and availability for massively multiplayer online games. In *11th European Dependable Computing Conference (EDCC 2015)*.
11. Yahyavi, A. and Kemme, B., 2013. Peer-to-peer architectures for massively multiplayer online games: A survey. *ACM Computing Surveys (CSUR)*, *46*(1), p.9.
12. gameranx 2016*, What Is A Dedicated Game Server & Why Is It IMPORTANT?* , video recording, YouTube, viewed 24 Mart 2018,<<https://www.youtube.com/watch?v=IoOrY-MPVzw>>.