Journal of the Nigerian Association of Mathematical Physics
Volume 31, **(July,** ***2015*****), pp 95 – 100**

Taxonomy of P2P Networks and their Associated QoS Parameters

[1]Adamu Aminu and [2]Ahmed Jamilu Bashir

Umaru Musa Yar'adua University, Katsina. P.M.B 2218, Katsina State, Nigeria.
Mathematics and Computer Science Department.

Abstract

Presently, many networks provide real time and on-demand services to users over the Internet. Initially client/server model was used for rendering these services. Due to scalability constraint of the client/server model among other reasons, many networks have now adopted Peer-to-Peer model (P2P model). In P2P model, each user is simultaneously a consumer and a provider of services, as such, more users means better performance. The paper describes the general architecture of P2P systems in detail. It further classified P2P networks based on Location-awareness, Overlay construction and Service delivery. Each class was comprehensively explained and examples for each class were provided. We further identified and explicitly defined the Quality of Service (QoS) parameters for P2P networks. The paper however, highlights the need to channel more effort on how to improve the identified QoS parameters in the paper, especially those that directly influence users' experience.

Keywords: P2P Systems, QoS Parameters, Location Awareness, Overlay Construction, Service Delivery.

1.0 Introduction

On demand and real time video services, such as online broadcast of News and sporting events were traditionally provided using the client/server model. The client/server model centralizes all resources at server locations. Host devices known as clients that are in need of any service provided by a system tender a request for such service to the system's server. The server then replies to such a request with the demanded service if it has not exhausted all its available resources for servicing clients' requests. As most people all over the globe are now having access to high speed Internet connectivity and with the proliferation of mobile devices such as tablets, laptops, smart phones, etc, the demand for online streaming services as well as the demand for files from the Internet are expected to rise. In that regard, client/server model can be considered as not a promising model for the provision of services to users because of its short comings, such as single point of failure, centralization of resources (when all the resources are exhausted at the server, no more services can be provided), more expensive servers are needed to serve multiple users, maintenance cost of servers is enormously high, etc., thus colossal resources are required to support or to provide services to ever increasing population of users. Examples of systems that deliver their services based on client/server model are Google TV, YouTube [1], etc. In many cases to overcome the above stated short comings associated to the client/server model in the provision of services such as real time and on Demand services, many networks adopt peer-to-peer model (P2P model). In modern P2P networks, each user acts as both client and server, therefore, users not only consume the resources of the network, but they also provide additional resources to it, and this is one of the advantage of P2P model over the client/server model, where all the resources came only from a single server. In P2P networks the more the number of users connected to it, the more the resources the network will have, this property of P2P networks makes them to be functional, efficient and scalable at any combination or quantity of users getting services from a network. Numerous P2P networks were successfully deployed and they provide live and on-demand video services to thousands of clients simultaneously, examples of such P2P applications include PPLive [2], PPStream [3], FeiDian [4], CoolStreaming [4], GridCast [5], BiTos [1], Zattoo [6], Joost [7], BitTorrent [8], etc. P2P networks are widely use for the exchange of different sort of data, e.g. elastic or stream traffic. Traffic measurements have shown that, P2P traffic consumes more than 60% of the whole Internet traffic [9].

Corresponding author: Adamu Aminu, E-mail: aminu@mail.ru, Tel.: +2348031581658 & 7067712786 (AJB)

The performance of traditional client/server model deteriorates rapidly as the number of clients increases, but in well-designed P2P network more users generally means better performance.
A detailed description of the general architecture of P2P systems is provided in section II. Taxonomy of P2P systems is presented in section III. We focus on the classification of P2P systems by the type of services and functions they deliver, because such classification is all-encompassing and reflecting the fundamental design and implementation differences. Section IV describes the Quality of Service (QoS) parameters for P2P systems based on service delivery. Section V concludes the paper. In the paper the terms system and network are used interchangeably.

2.0 P2P Architechture

In a simple P2P system, the basic elements are the server, tracker and a set of host devices that are currently sharing/downloading the same file or watching the same video. Each host device is called a peer, and every peer acts as both a client and a server [1, 9, 10, 11, 12], i.e. each peer consumes and provides resources to the system (Figure.1a), unlike the client/server model, where all the resources come only from the server (Figure.1b).
The server generates real time data or hosts on-demand data such as stored movies, software packages, books, etc. The tracker, as its name implies, keeps track of a subset of active peers that are currently downloading the same file or watching the same video [1, 9].
The basic idea in P2P architecture, consist of the followings: A whole file or video is divided into small blocks of data elements of equal sizes, known as chunks, and peers exchange these chunks among themselves. If a user is missing some chunks of the requested file, it request and download those missing chunks from neighboring peers in the network downloading the same file or watching the same video. A new peer joins the system by undertaking three (3) steps [2] as shown in Figure 1a.

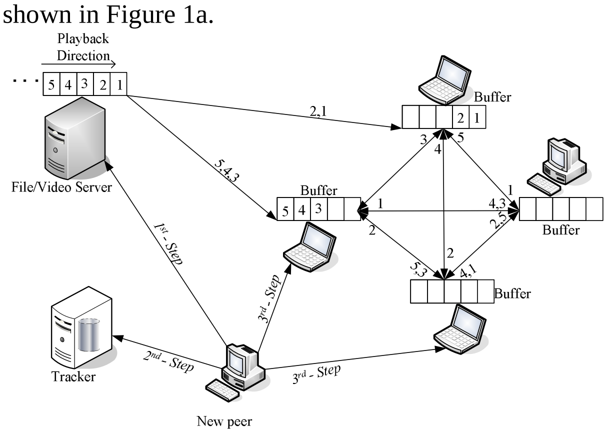

Figure 1a: P2P Architecture

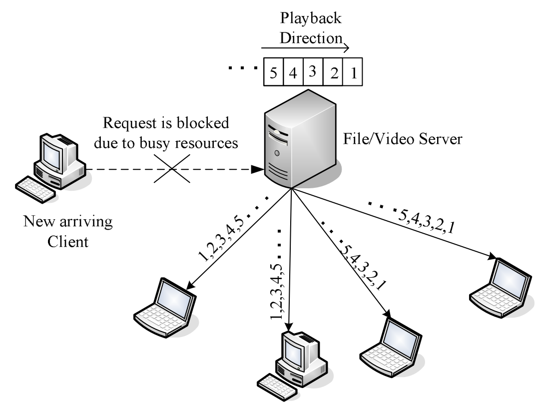

Figure 1b: Client/Server model

In the first step, a new peer sends a query message to the file/video server. The file/video server returns with an up-to-date list of files/videos (i.e. file list or video list) from which the new peer selects its favourite file/video. The second step begins after a peer has selected the file/video it intends to use. The peer sends its information such as IP address, Port number, etc to the tracker. The tracker then returns with a list of active subset of peers that are currently downloading the chunks that make up the selected file/video. After receiving the list of peers from the tracker, the peer begins its final stage of joining the P2P system by sending a connection request to peers from the given list. Once such request is accepted by peers from the list, the peer under consideration will form a list of its neighbouring peers with which it will collaborate. Having being accepted by some peers for collaboration, chunks of file/video are then shared between the peer under consideration and its neighbours.
Figure 1a shows how the first five (5) chunks of a file/video are being downloaded by peers. Early arriving peers obtain desired chunks from the server and later arriving peers source their chunks from the early arrived peers and possibly from the server. Each peer posses a buffer of defined size where chunks are cached prior to use (Figure 3). Each chunk has a sequence number that serves as an identifier of its position in the file/video. Chunks are obtained from the network at random and are automatically re-arranged sequentially upon arrival at the buffer prior to use. Though peers join and leave the network system in an unpredictable manner, the probability of obtaining chunks remains high because if a peer is short of neighbours, it can send collaboration requests to other peers.

3.0 P2P Systems Taxonomy

P2P networks are classified based on the following criteria:

(i) Location-awareness
(ii) Overlay construction
(iii) Service delivery

(i) Location-awareness: Peers may or may not be aware of the exact neighbouring peers with which it shares chunks. On this basis P2P systems can be classified as either structured or unstructured. Each peer in a structured P2P system is assigned a unique peerID; similarly, each content or data element is assigned a unique key and mapped to the peer that harbours it. Generally, P2P systems keep track of all the peerID's as well as the keys. To download content from the network, a requesting peer must use the pair {Key, peerID} to efficiently locate and download the required content. Examples of algorithms designed for the implementation of structured P2P systems include: CAN [5], Viceroy [13], Kademlia [13], Chord [14], Tapestry [15], Pastry [15], FreeNet [14], etc. Unstructured P2P systems allow participating peers to locate their desired content from neighbouring peers without any knowledge of the system topology. Peers in unstructured P2P systems employ flooding technique in locating contents from neighbouring peers. Each peer will continue to broadcast or flood its request directly to neighbouring peers, until all or some of the collaboration requests are granted or the maximum number of permitted flooding steps are exhausted [11, 15, 16]. In such case, flooding is known to be of limited scope and thus not so efficient especially in locating contents that are rarely available. Examples of unstructured P2P systems include: BitTorrent [17], Napster [18], Gnutella [18], FastTrack [8] and Overnet [19].

(ii) Overlay construction: Peers may source for their required chunks from one or multiple number of neighbouring peers simultaneously. As such, peers would therefore form different topology in order to accommodate the number of neighbouring peers that serve them with required chunks. According to the overlay construction of peers, P2P systems can be classified as either tree–based or mesh–based systems. In tree-based P2P systems, peers obtain chunks from their parent peers. Peer churn affects tree-based P2P networks especially when a parent peer leaves the network thereby disconnecting its children peers from receiving their desired chunks. Though disconnected children peers get reconnected to a new parent peer after some time, the children peers would experience poor quality (called freezing) of the video they are watching as a result of the missing chunks whose playback deadline elapsed before downloading them. Some examples of tree-based P2P systems include: End System Multicast (ESM), OverCast, etc. Alternatively, mesh-based P2P systems allow peers to form and obtain chunks from multiple neighbouring peers simultaneously. Peers can further add, change and remove any of their neighbouring peers whose chunks have already been downloaded. This independent ability to increase, change and decrease the number of neighbouring peers ensures that at any given time, a peer can still download desired chunks irrespective of the rate of peer churn in the system. Some examples of mesh-based P2P systems include: SopCast [20], CoolStreaming, PPLive, etc.

(iii) Service delivery: With respect to the service P2P system provides to users, P2P systems can further be classified as either file sharing or streaming systems. In P2P file sharing systems, peers exchange chunks among themselves. If a peer is missing some chunks of the requested file, it request and download those missing chunks from neighbouring peers in the network downloading the same file (Figure 2). The entire chunks that make up a file must be downloaded by a user before that file can be put to use. Examples of successfully deployed P2P file sharing systems are BitTorrent, Napster, Gnutella, etc.

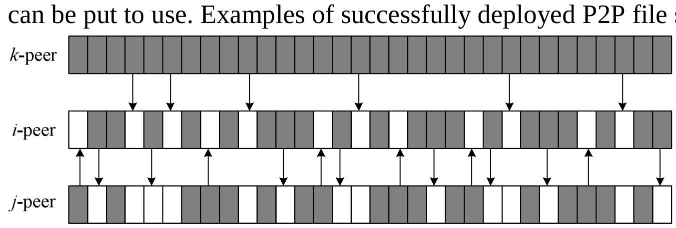

Figure. 2: File sharing systems

In P2P streaming systems, peers are simultaneously downloading and watching the successfully downloaded video chunks from the network (Figure 3). A new peer begins watching video after downloading some sufficient number of video chunks from the network enough to be played for some time. Each peer in P2P streaming system posses a playback buffer where it stores the successfully downloaded video chunks. Each chunk in P2P streaming has a playback deadline, and any chunk that arrived at the user's buffer after its playback deadline is useless and will be discarded.

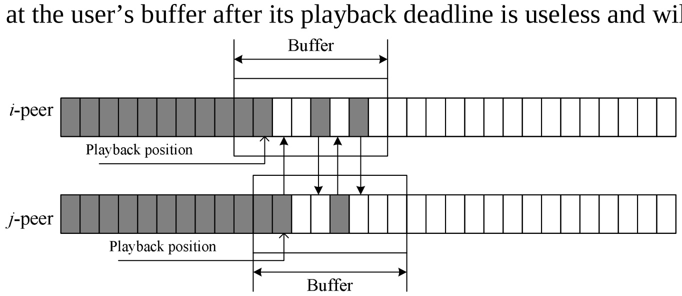

Figure. 3a: Live streaming systems

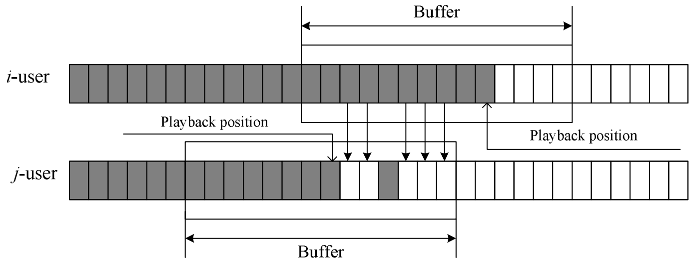

Figure. 3b: On demand systems

Figure.3: Live and on-demand streaming systems

Under this classification however, P2P streaming systems can be further classified as either P2P Live streaming systems (Figure 3a) or P2P Video-on-Demand (VoD) streaming systems (Figure 3b). Live streaming P2P systems broadcast live videos to all viewers in real time, this makes video playbacks on all users to be synchronized as such; all users watch almost the same position of the same video. Examples of P2P Live streaming systems are: Joost [21], Zattoo [21], PPLive, CoolStreaming, etc. In P2P VoD streaming systems, users may choose any video of their interest and begin to watch any part of the video at any time. The playback positions of the same video on different users are therefore not synchronized. Examples of P2P VoD streaming systems are: GridCast, BiToS, etc.

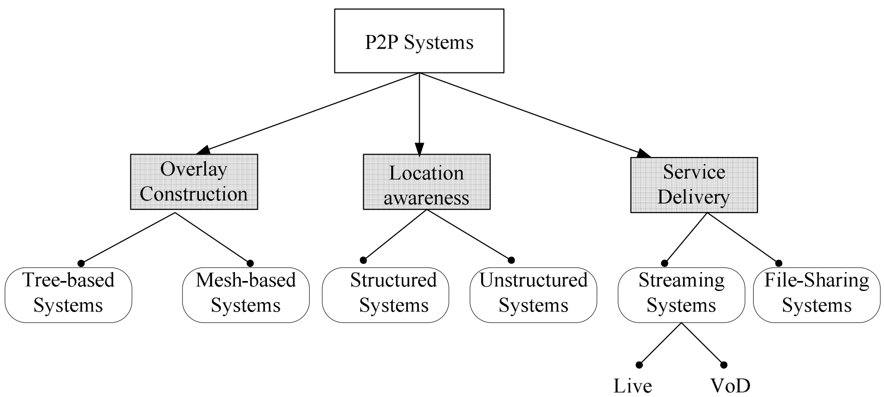

Figure 4: P2P Systems and their classifications

4.0 Quality of Service Parameters for P2P Systems

Certain performance measures define the quality of service delivery in P2P file sharing and streaming systems. The parameters are shown on Figure 5.

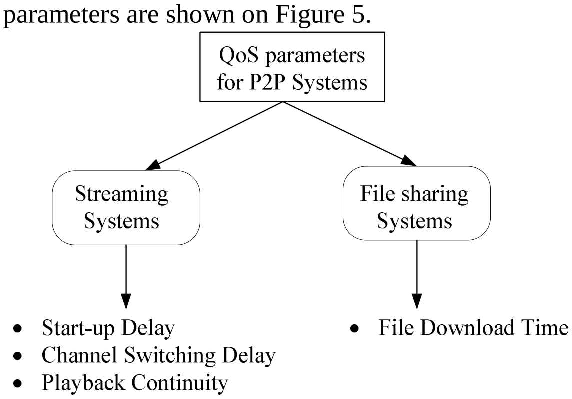

Figure 5: QoS parameters for P2P streaming and file sharing systems

In P2P file sharing systems, a file can only be used after all its component chunks have been successfully downloaded by a peer. Therefore, the basic QoS parameter for P2P file sharing systems is the total time needed or taken to download a complete file (*file download time*). The file download time is the time interval from the moment the file download started to the moment the file download completed.

P2P streaming systems have the following QoS parameters: *start-up delay*, *channel switching delay* and *playback continuity*.

The *start-up delay* is the time difference in seconds between the moment a peer joined streaming session to the actual moment the playback started. This is experienced by peers because sufficient video chunks must be cached in the playback buffer before users can begin watching video. Traditional cable and satellite TV systems have start-up delay within the range of 5.0 – 8.0 seconds. In comparison, our analysis revealed high start-up delay for P2P TV systems. For instance, we observed start-up delay for Joost P2P TV and other P2P TV systems is within the range of 30.0 – 35.0 seconds.

Channel switching delay is the total time in seconds it takes for the system to respond to a channel change request made by a peer. Usually, a peer requesting for a channel change must send collaboration request and wait for subsequent acceptance from a group of peers watching the same channel. The time taken by a peer in facilitating collaboration requests and subsequent downloading of chunks contribute to longer switching delays. Traditional cable and satellite TV systems have a channel switching delay within the range of 3.0 – 4.0 seconds. Similar to start-up delay, our analysis revealed high channel switching delay for P2P TV systems. For instance, we noticed that, PPLive and other P2P live streaming systems have Channel switching delay within the range 10.0 – 60.0 seconds.

In P2P streaming systems, each chunk has an associated and fixed playback time and must be at a peer's buffer before that time so that users can have good viewing experience. *Playback continuity* points to the state, in which a peer experiences an uninterrupted video playback. When every chunk arrived at the buffer before its playback deadline elapsed, smooth playback will be experienced by a peer otherwise, the peer will experience video freezing or pause at instances where chunks are not received before their playback deadline. A peer with numerous and distinct chunk sources simultaneously has a higher chance of experiencing smooth playback continuity than one with fewer chunk sources.

5.0 Conclusion

In this paper, a detail description of P2P system is provided. Several criteria used in classifying P2P systems were also presented. We have identified and explicitly defined associated QoS parameters for both P2P file sharing and streaming systems while highlighting their features, differences and characteristics. In light of the conducted study, we come to a conclusion that much needs to be done to standardize and improve the Quality of Service metrics of both P2P file sharing and streaming systems. Our future work will first focus on how to improve QoS parameters of P2P streaming systems such as minimization of start-up delay and channel switching delay and maximization of playback continuity. We choose streaming systems because of their popularity, vast deployments, wider acceptance and their QoS parameters have a direct impact on the user QoE (Quality of Experience).

References

[1]. Liu, Y., Guo, Y., and Liang, C. (2008) A Survey on Peer-to-Peer Video Streaming Systems. Journal of Peer-to-Peer Networking and Applications, 1: 18 – 28

[2]. Hei, X., Liu, Y., and Ross, K. W. (2008) Understanding the Start-up Delay of Mesh-pull Peer-to-Peer Live Streaming Systems

[3]. Hei, X., Liu, Y., and Ross, K. W. (2007) Inferring Network-Wide Quality in P2P Live Streaming Systems. IEEE Journal on Selected Areas in Communications, 25, 9: 1640 – 1654

[4]. Kumar, R., Liu, Y., and Ross, K. (2007) Stochastic Fluid Theory for P2P Streaming Systems. 26TH IEEE International Conference on Computer Communications. pp. 919 – 927

[5]. Cheng, B., Liu, X., Zhang, Z., Jin, H., Stein, L., and Liao, X. (2008) Evaluation and Optimization of a Peer-to-Peer Video-on-Demand System. Journal of Systems Architecture 54: 651 – 663

[6]. Alhaisoni, M., Liotta, A., and Ghanbari, M. (2009) Performance Analysis and Evaluation of P2PTV Streaming Behaviour. IEEE Computers and Communications pp. 89 – 94

[7]. Agboma, F., Smy, M., and Liotta, A. (2008) QoE Analysis of a Peer-to-Peer Television System. IADIS International Telecommunications, Networks and Systems pp. 114 – 119

[8]. Qui, D., and Srikant, R. (2004) Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. ACM SIGCOMM Computer Communication Review, 34(4): 367 – 378

[9]. He, Y., and Guan, L. (2010) Peer-to-Peer Streaming Systems. Intelligent Multimedia Communication: Techniques and Applications; Studies in Computational Intelligence, 280: 195 – 215

[10]. Veciana, G. D., and Yang, X. (2003) Fairness, incentives and performance in Peer-to-Peer Networks. In 41ST Annual Allerton Conference on Communication, Control and Computing

[11]. Wang, C., and Li, B. (2003) Peer-to-Peer Overlay Networks: A Survey

[12]. Gupta, A., and Awasthi, L. K. (2011) Peer-to-Peer Networks and Computation: Current trends and Future Perspectives. Computing and Informatics, 30: 559 – 594

[13]. Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. (2005) A Survey and Comparisons of Peer – To – Peer Overlay Network Schemes. IEEE Communications Surveys, 7: 72 – 93

[14]. Theotokis, S. A., and Spinellis, D. (2004) A survey of Peer – To – Peer Content Distribution Technologies. ACM Computing Surveys, 36: 335 – 371

[15]. Milojicic, D. S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., and Xu, Z. (2002) Peer-to-Peer Computing. HP Laboratories, Palo Alto

[16]. Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J., and Towsley, D. (2003) Modelling Peer-Peer File Sharing Systems. IEEE INFOCOM 22nd Annual Joint Conference of the IEEE Computer and Communications and IEEE Societies, 3: 2188 – 2198

[17]. Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., and Zhang, X. (2007) A Performance Study of BitTorrent-like Peer-to-Peer Systems. IEEE Journal on Selected Areas in Communications, 25: 155 – 169

[18]. Saroiu, S., Gummadi, P. K., and Gribble, S. D. (2002) A Measurement Study of Peer-to-Peer File Sharing Systems. Proceedings of the International Society for Optics and Photonics, Multimedia Computing and Networking 4673: 156 – 170

[19]. Fessant, F. L., Handurukande, S., Kermarrec, A. M., and Massoulie, L. (2004) Clustering in Peer-to-Peer File Sharing Workloads. Proceedings of the Third International Conference on Peer – to – Peer Systems. Springer – Verlag Berlin, Heidelberg. pp. 217 – 226

[20]. Lu, Y., Fallica, B., Kuipers, F. A., Kooij, R. E., and Mieghem, P. V. (2009) Assessing the Quality of Experience of SopCast. International Journal of Internet Protocol Technology, 4: 11 – 23.

[21]. Krieger, U. R., and Schwebinger, R. (2008) Analysis and Quality Assessment of Peer-to-Peer IPTV Systems. IEEE International Symposium on Consumer Electronics pp. 1 – 4.