DOI: http://dx.doi.org/10.21123/bsj.2019.16.3.0667

PWRR Algorithm for Video Streaming Process Using Fog Computing

Saba Fouad Hassan *Rana Fareed Ghani* [*]

Received 6/5/2018, Accepted 23/1/2019, Published 1/9/2019

Abstract:

The most popular medium that being used by people on the internet nowadays is video streaming. Nevertheless, streaming a video consumes much of the internet traffics. The massive quantity of internet usage goes for video streaming that disburses nearly 70% of the internet. Some constraints of interactive media might be detached; such as augmented bandwidth usage and lateness. The need for real-time transmission of video streaming while live leads to employing of Fog computing technologies which is an intermediary layer between the cloud and end user. The latter technology has been introduced to alleviate those problems by providing high real-time response and computational resources near to the client at the network boundary. The present research paper proposes priority weighted round robin (PWRR) algorithm for streaming operations scheduling in the fog architecture. This will give preemptive for streaming live video request to be delivered in a very short response time and real-time communication. The results of experimenting the PWRR in the proposed architecture display a minimize latency and good quality of live video requests which has been achieved with bandwidth changes as well as meeting all other clients requests at the same time

Key words: Fog Computing, video streaming, WRR algorithm, Priority algorithm, DASH technology.

Introduction:

The Internet passing that used for video streaming is around 70% of the total internet bandwidth, which is considered as the biggest consumer of the internet. Video streaming is predicted to employ as 82% of the total internet traffic in 2020 depending on Cisco statistic (1) Video is a way to communicate which is more popular than the text used for the first contract on the internet. Many internet multimedia applications such as distance learning, corporate communications (earnings calls, human relations, corporate addresses, and press releases), video conferencing, home shopping and amusement (video-on-demand) contain video streaming which is becoming a significant portion of these applications. There are two ways to applied Streaming; first video can be transported to the viewer simultaneously. It records that is live streaming. The second method is that the viewer requests the media from a server contains previously stored media, just like using a library. This is called video-on-demand (2). Cloud computing provides high storage and speed processing for each online service.

Multimedia streaming and emergency notification are real-time services which require speedy response and reduce latency. The latency and the user coverage limitation are due to the fact that data is transported between the end-users and the cloud centers that made the highest quality of interactive service providers to users cannot be realized by cloud services. To alleviate the aforementioned problems, fog computing technologies can be utilized to provide high real-time response and computational resources near to the client at the network boundary (3,1). Kashif and Aiman (2017) (1) claimed that fog computing supports time response reduction, energy, storage requirements and provides extreme bandwidth usage. The authors focused on challenges that limit the interacting media and video streaming and offer an applicable solution using fog technology. Immersive, multi-view streaming and gaming society's problems have been solved by offering a suitable solution from the edge technologies that shows swift real-time responsiveness and interactivity. Chin-Feng et al. (2016) (3) discussed how to manage resources and adapt the video to the existing network conditions through using the fog computing mechanism to achieve QoS-aware Streaming Service, which relieves the traditional content delivery issues. The proposed architecture

Department of Computer Science, University of Technology, Baghdad, Iraq.
[*]Corresponding author: 11415@uotechnology.edu.iq

has three types of nodes: multiple center node which stores the media content and sent it to the user when requested, one index node that responsible of connection and register all nodes in fog infrastructure, and the last is the multiple edge node which is accountable for streaming media segmentation to end user when request it from center node and the display level of the video that depends on available bandwidth. Ning et al. (2016) (4) proposed a real-time vehicle traffic monitoring system where an unmanned plane is used to monitor vehicles then transmits video using a fog computing model to a real-time video streaming. It focuses on the fact that the system meets the real-time monitoring requirements by using a fog model, that shows a high response time and high transmission for the video stream. The proposed control system architecture observing the area of attention by drone, then video data streaming will be sent to the supervising station to find the wanted vehicle in the real-time video. Fog computing node which is closer to site will execute the tracking algorithm where send only sub-area frames that contain the wanted vehicle and excluded from the main frame of the video and transmit. Hongbin Yang et al. (2016) (5), suggested that video streaming in a real-time needs a high response which makes the delay scheduling algorithm (spark schedule) not appropriate for streaming because it has fixed threshold to execute each task. If a task is not complete with a threshold that will lower the execution time of the following tasks and may move some task needed to run locally to another node. This will increase delay, traffic congestion and tasks movement between nodes asking for resources. The authors provided Optimizing of the spark scheduling to work with video streaming in real time through employing the concept of data locality in execution and measuring the tasks execution time dynamically in each node through the weighted method, that will assure tasks running in a time window in each node. As for Luiz F. Bittencourt (6), in Fog computing, end user can get services and applications with minimum connection delay and higher processing rate as a result of distributing the computational resource in the network boundaries. Authors discuss the task scheduling problem that shows in resource allocation in fog environment and how can prepare the task scheduling to deal effectively with various user applications. User applications classified according to user geolocation and edge computing capability into two types: the first one is an application that needs real-time processing and interactive which execute in edge nodes like tractor beam game; while the second type is permissive-lateness where the user is always in the cloud like a video surveillance. This paper presents three scheduling algorithms: Concurrent strategy, First Come-First Served (FCFS) strategy, and Delay-priority. The results show the performance of the algorithms in terms of delay as well as estimate how allocation policies affect network traffic.

In the present paper, an architecture for streaming videos via the fog computing is proposed to reduces the latency and network congestion for interactive applications and executing the priority weighted round robin (PWRR) algorithm on the streaming chunks for scheduling the streaming operation into the fog node. The target is to assure that the given live video gets a preference in streaming beside response for all clients requests at the same time with less latency and maintain video quality according to bandwidth alteration. While in the authors" paper (Kashif and Aiman (2017) (1) they only focused on the possibilities and prospects of using the fog computing to deal with difficulties in interactive media.

The rest of this paper is arranged as follow: Section 2 provides a discussion about video streaming, fog computing and simulation, and task scheduling algorithms. Section 3 explains the proposed architecture with scheduling algorithms. Section 4 has the results of the proposed architecture. Section 5 presents conclusion and future work.

Related Work

This section describes the brief details for the following subsections: video streaming, fog computing and simulation, and the task scheduling algorithms used in the proposed architecture:

Video Streaming

The growth and prevalence of the Internet in the middle of the 1990’s encourages video communication over the internet network despite the fact that there are many complications in video stream such as bandwidth varying, lateness, and losses of packets. Many other issues need to be considered, such as how to make the network resources equally shared amongst many users and how to efficiently implement one-to-many connection for public content (7). Video streaming refers to the real-time transportation of a video. There are two operation processes to transport a video through the Internet. The first one is the downloading mode and the second is streaming mode. The former downloads the entire video file and then plays it back, while real-time is not a necessity. In contrast, for streaming mode, it is a must that the video content needs to be delivered as packets and played in a real time when they arrive at the destination. Video streaming quality of service (QoS) requests height bandwidth, lower loss rate, and rigorous timing restrictions (8). Live Streaming and Video-On-Demand (VoD) are the major kinds of Video-streaming based on the source of the content. In Live Streaming, the video is captured at the source by a camera and is broadcasted to all the client subscribers immediately; video conference is an example of the Live Streaming type. The equipment needed for a Live Streaming service is usually a camera, an encoder to digitize the video, a media publisher, and a network to spread and deliver the video. In VoD services, the video content is stored in a server and is transmission starts when end-users request it. For example, YouTube and Vimeo (9).

Fog Computing and Simulation

Cisco introduced the fog computing as an extended to cloud computing. The concept of fog computing is to place computational and storage capabilities at the network boundary. These capabilities can be accomplished by placing servers and storage devices close to the client. Fog computing is specifically introduced to facilitate latency-sensitive applications and IoT vision, to relieve real-time, mobility, and location consciousness requirements (1,10). Fog computing is a decentralized computing architecture where data is processed and stored between the source and a cloud infrastructure. This leads to data transmission decline overheads, and thereafter, gets better execution of computing in Cloud platforms by reducing the demand to process and store large volumes of unnecessary data. The Fog computing model is largely encouraged by a continued rise in the Internet of Things (IoT) devices (10,11). The fog computing architecture of this paper has been designed as follow: the fog node is as the computing tasks for cloud center node, it is distributed to the network boundary for the real-time services. Cloud computing is the center node that is used for processing the most of computing and storage services for end-devices. Fog node is used for emergency and critical situations to avoid too long response time of the service (3). Figure.1 shows the fog computing model in the proposed architecture.

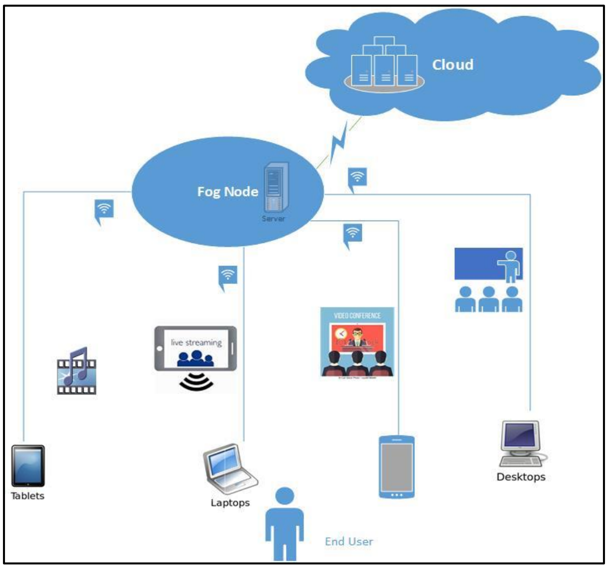

Figure 1. Fog Computing Model

The iFogSim has been chosen to make a simulation for fog computing because it runs on upper of the well- determined CloudSim simulator. It also supports the measurement of application delays, and has been vastly utilized and tested that allows the hierarchical composition of edge devices, cloudlets, and clouds. To evaluate the weighted round robin scheduling algorithm in a fog computing environment a simulation has been performed. It aims to show the performance of the streaming system and the scheduling algorithm in delay as long as rating the change of video quality according to the network bandwidth (12). The scenario contains video streaming demand from the fog node server, the video types are a live video like video conference and chat video. VOD like watches a movie and a scientific lecture.

Task Scheduling Algorithms

1- Weighted Round Robin Algorithm

To schedule the chunk streaming in the fog computing environment, a weighted round robin (WRR) has been used, a commonly used scheduling algorithm because of its plainness and lower computational overhead (13). WRR extends the potential of round robin (RR) algorithm to give different weights to queues. Packets are first categorized into service classes and assigned to specified queues. The scheduler checks each queue and packets are sent from the checking queues. It allows distinguishing in service class treatment (14), that calculation of dynamic time slice can be achieved for the algorithm during the streaming operation by calculating the mean from the client requested videos lengths in the queue, which will reduce the waiting time for all client requests eventually.

2- Priority Scheduling Algorithm

The assignment of priority is constant and has done by the operating system for every process in the system. The higher priority process has caused an interruption to the lower priority process execution when accessing the system and it has a smaller waiting time. The queues for priority algorithm have produce famine for lower priority processes which are waiting for all higher priority processes to be performed first. Priority algorithm uses another scheduling algorithm queues to execute the important jobs as in static and dynamic scheduling algorithm. (15)

The Proposed Architecture with Scheduling Algorithms

The streaming process of the video requires a large server to storage for VoD and powerful processors to reduce latency for Live video, in addition to the availability of bandwidth of the network that controlled the quality of delivered videos. Depending on this, the trend to create an architecture for streaming videos via fog computing for both its kinds in an efficient way with the network available resources to relieve the issues that faced video streaming on the internet. The proposed architecture is client-server that the server is fog compute node, C# language has been used to program the system. The first step is a socket program which uses TCP/IP protocol to make the communication among the server and multi-clients. The protocol for streaming in the proposed architecture performed according to the video type request. If the client requests VoD; that means the use of the simulation method of dynamic and adaptive streaming over http technique to stream video chunk which includes three step of streaming: first, request VoD file that compressed by the JEPG algorithm in three different bitrate levels and stored in cloud node, then determine one of the three compressed levels of the video to be on stream upon request to the client according to the available bandwidth, and finally encapsulate video frames into chunks for streaming. On the other hand, if a live video has been requested by the client, the protocol compressed the video frames by JEPG algorithm according to the client available bandwidth and then encapsulates it into chunks for streaming. Figure.2 clarifies the process. Figure.2 clarifies the propose architecture.

Video streaming has used DASH (Dynamic and Adaptive Streaming Over HTTP) as a predominant criterion for transferring frames through the Internet. DASH has been used particularly for mobile devices which allows the client to control the quality of streaming video when the quality of the network is fluctuating. The video is encoded in various qualities, however; each is divided into a short part to adaptive to network variations. The base transmission protocol in DASH is HTTP/1.1.through sequential downloads of short video segments of different bitrates, DASH supply for the client controlled video content transmission (where higher bitrate means higher quality), DASH submits content over the (TCP) by using the (HTTP), for example video stored in four different qualities in the server (L, M,H,VH) and divided in seven-part to be delivery to client through the network with bandwidth varying, client requests the lowest available quality when bandwidth is very low and when bandwidth has been improved the client request also improves (16).

The proposed protocol in the present paper of the server is determining the bandwidth of each client. When the client sends the dimension of the video displayed to the server; the will extract frames from the video to calculate the quality. Increasing the bandwidth means to increase the streaming chunks to the client with high quality and vice versa. The quality of the frames depends on the existing bandwidth of the client if the bandwidth is = >5000 which considered as a good bandwidth; the compression gives 50% quality to frames. The maximum bandwidth rang when using the wired local network is 52mbps, and when using wireless is 100 mbps. In the proposed architecture the bandwidth range of clients has been assumed to be between (1000-100000) bps which are a randomly generated number using internet network. The result of the algorithm in Figure.3 exemplifies the ratio of compression of frames and that leads to determine the quality of the video. Frames have encapsulated into chunks where each chunk includes 20 frames to be transmitted to the client side.

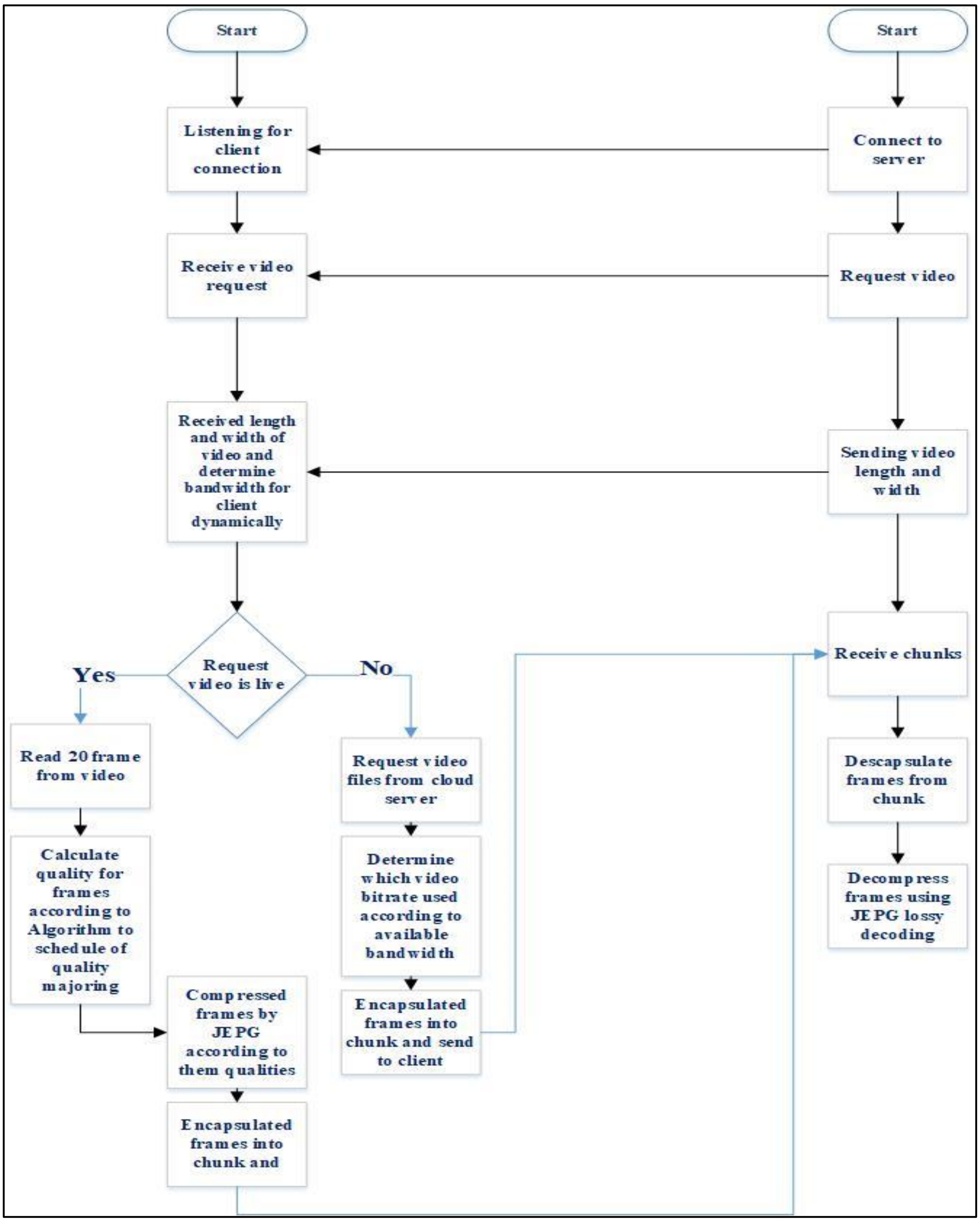
Start
Start
Listening for client connection
Connect to server
Receive video request
Request video
Received length and width of video and determine bandwidth for client dynamically
Sending video length and width
Request video is live
Yes
No
Receive chunks
Read 20 frame from video
Request video files from cloud server
Descapsulate frames from chunk
Calculate quality for frames according to Algorithm to schedule of quality majoring
Determine which video bitrate used according to available bandwidth
Decompress frames using JEPG lossy decoding
Compressed frames by JEPG according to them qualities
Encapsulated frames into chunk and send to client
Encapsulated frames into chunk and

Figure 2. The Proposed Architecture

```
Calculate Quality
INPUT: Bandwidth, Threshold
OUTPUT: Quality
PROCEDURE:
IF ((Bandwidth / Perfect Bandwidth) * 100 >= Full Quality Threshold)
        Quality = Full Quality Threshold
ELSE
        Quality = Bandwidth / Perfect Bandwidth * 100;
END IF
END PROCEDURE
```

Figure 3. Algorithm to Schedule of Quality Majoring

The compression algorithm employed in the proposed protocol to compress frames is JEPG Lossy compression. The International Telecommunication Union (ITU) and International Organization for Standardization (ISO) standardization bodies produced it and considered as one of Video Compression Standards. The DCT transform and a quantization technique are used in

JPEG to eliminate redundant information. This popular standard performs a frame-by-frame compression of video data and gives the flexibility to either choose high picture quality with rightly high compression ratio or to get a very high compression ratio at the expense of a sensible lower picture quality. It shows variable compression ratios, where very high compression ratios result in “blockiness” of the compressed image, (16). JEPG provides a high compression for video through reducing redundancy information in each frame which gives faster streaming operation while keeping the acceptable quality of the video.

Experimental Results and Discussion:

The main purpose of the proposed architecture for video streaming is making live video gets preemptive of the fog server resources in transmission and amount of bandwidth from the server to the client. That can be achieved by applying the scheduling algorithm on the streaming chunks to measure the quality of the streaming process in the proposed architecture. The QoS parameters have been performed on the streaming chunks request from the fog server, the parameters which have been used to show the impact in the video streaming are latency, bandwidth variation relation with video quality, and the RMSE measurement. The proposed architecture uses four videos (Lecture, Movie, Live video, Conference video) with two scheduling algorithms; the WRR and PWRR. The aim is to get the best quality with low latency for the real-time video request to get the user acceptability.

Dynamic time slice has used in the scheduling algorithms to execute the streaming for each client request which provides equal time to transmit all the clients' video chunks. In WRR algorithm the real-time video (Live video, Video conference) don't get interactive processing that shows in high latency because it has to wait in the queue until the other requests served, besides fixed bandwidth that assign from the server. While in PWRR algorithm when real-time video request received at the server; it gets the highest preemptive to stream and gets the highest bandwidth from the server to provide chunks transmission which reduces the bandwidth to the other VoD requested video (non-preemptive). Figure.4 and Figure.5 respectively display the values for latency and buffer size that varies with the bandwidth of eight tests to four video types through the two scheduling algorithms.

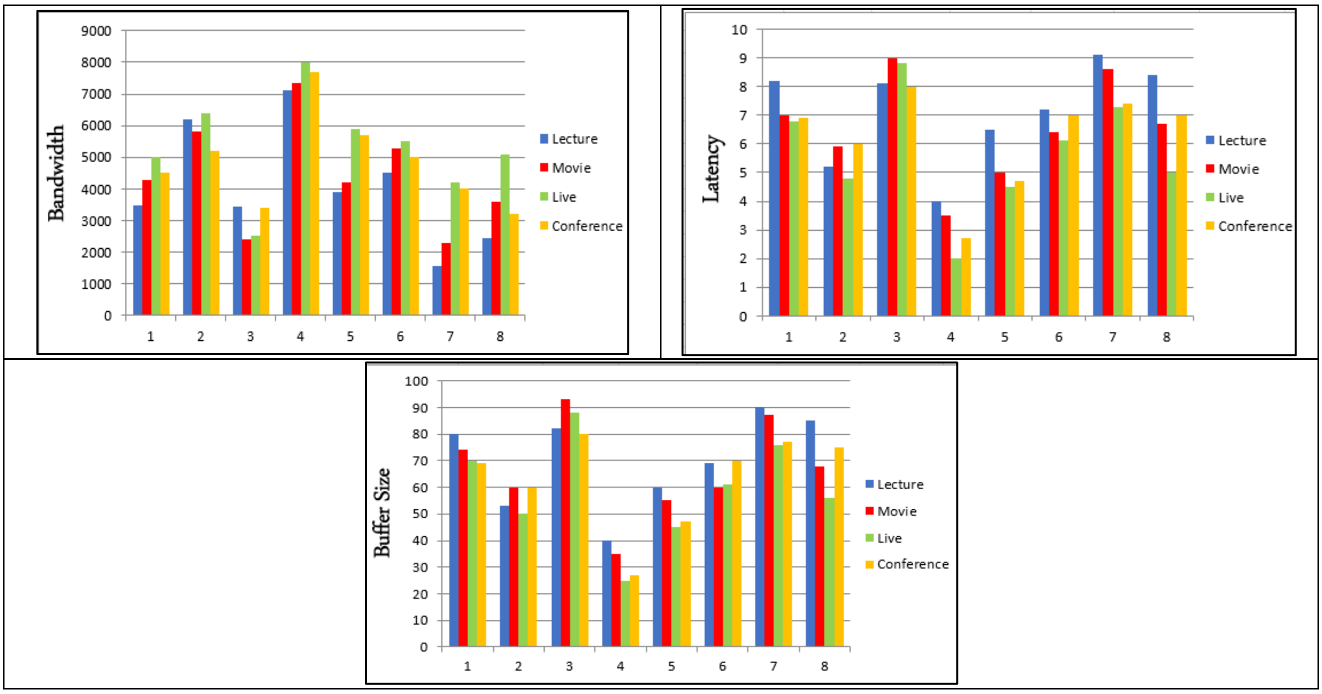

Figure 4. Weighted Round Robin results

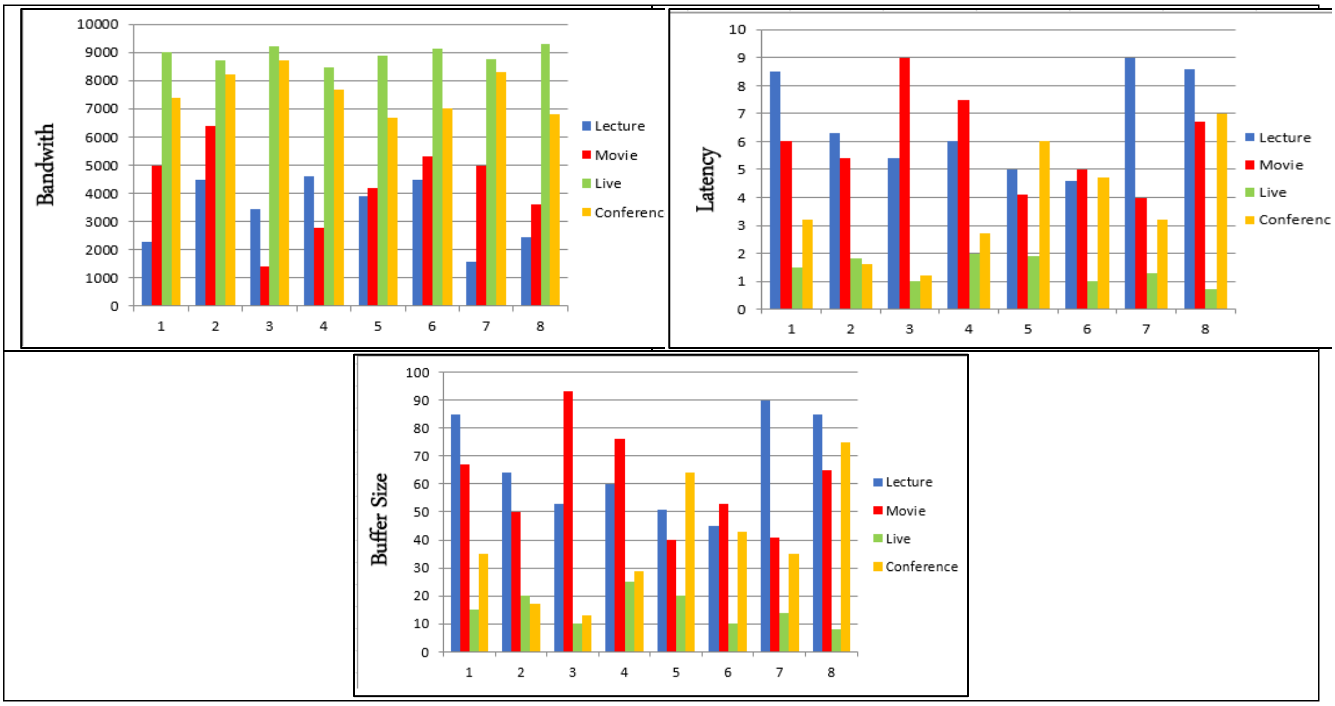

Figure 5. Priority Weighted Round Robin results

Figure.4 and Figure.5 exemplify the relation between bandwidth, latency, and the buffer size; where increasing the bandwidth leads to reduction of the latency and the buffer size in the client side, that means the video chunks will immediately be displayed when arrived at the client player. In WRR the real-time video request has a high latency in spite of its high bandwidth because the time slice of the algorithm enforce each request of the video waits until the performance of the other request in the queue (that arrived before) fulfilled. This does not serve requests of delay-sensitive. While in PWRR, when the real-time video request arrived to the fog server, it makes an interruption to the other video request streaming VoD, and gets the highest priority for streaming with low latency and good quality.

The RMSE is one of the qualities of service measurements that mean Root Mean Square Error (RMSE) is a mathematical method used to calculate the difference quality between the original video frames and the received video frames at the client side through the streaming process. The RMSE equation is clarified as follows:

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Where y^i represent the streaming video frames while yi represent the original video frames. This method compares each frame between the original and streaming video by subtraction operation to find the root of the square of the difference between frames. In the proposed architecture when perform both WRR and PWRR. The RMSE values in the PWRR is reduced for the real-time videos request which means that this videos frames have a high quality as the original video frames and has good streaming features, whereas in WRR the real-time videos are affected by variable qualities and latency based on the available bandwidth. In PWRR when the request with high priority (real time video) is received at the server side, this makes the quality and the bandwidth of the other request (VoD) with lower priority became less quality and increased the RMSE for them. Table.1 and Table.2 respectively clarify the results of measuring the RMSE of the four types of video streaming in eight tests in both scheduling algorithm WRR and PWRR, also see Figure.6.

Table 1. RMSE for video samples in WRR algorithm

Video name	Frame dimensions	Number of frame	Bandwidth	RMSE in WRR
Lecture	400 x 224	5953	3500	7.5
			6200	5.4
			3450	7.6
			7100	4.2
			3900	6.9
			4500	6.4
			1560	9.1
			2450	8.2
Movie	640 x 344	13099	4300	6.9
			5800	6
			2400	8
			7340	4
			4200	6
			5300	6.5
			2300	8.7
			3600	7
Live video	1280 x 720	4918	5000	7
			6400	5.7
			2540	8.1
			8000	3.6
			5900	5.8
			5500	5.9
			4200	7.4
			5100	6.8
Video conference	640 x 360	4820	4500	7.1
			5200	5
			3400	7
			7700	3.9
			5700	5.9
			5000	5.7

Table 2. RMSE for video samples in PWRR algorithm

Video name	Frame dimensions	Number of frame	Bandwidth	RMSE in PWRR
Lecture	400 x 224	5953	2300	7.2
			4500	6.4
			3450	6.8
			4600	6.7
			3900	7.8
			4500	6.4
			1560	9.1
			2450	8.2
Movie	640 x 344	13099	5000	7
			6400	5.9
			1400	9.2
			2800	8.1
			4200	6.4
			5300	6.5
			2300	7
			3600	3.6
Live video	1280 x 720	4918	9000	7
			8700	5.7
			9200	8.1
			8450	3.6
			8900	5.8
			9120	5.9
			8750	7.4
			9300	6.8
Video conference	640 x 360	4820	7400	7.1
			8200	5
			8700	7
			7680	3.9
			6700	5.9
			7000	5.7
			8300	7.7

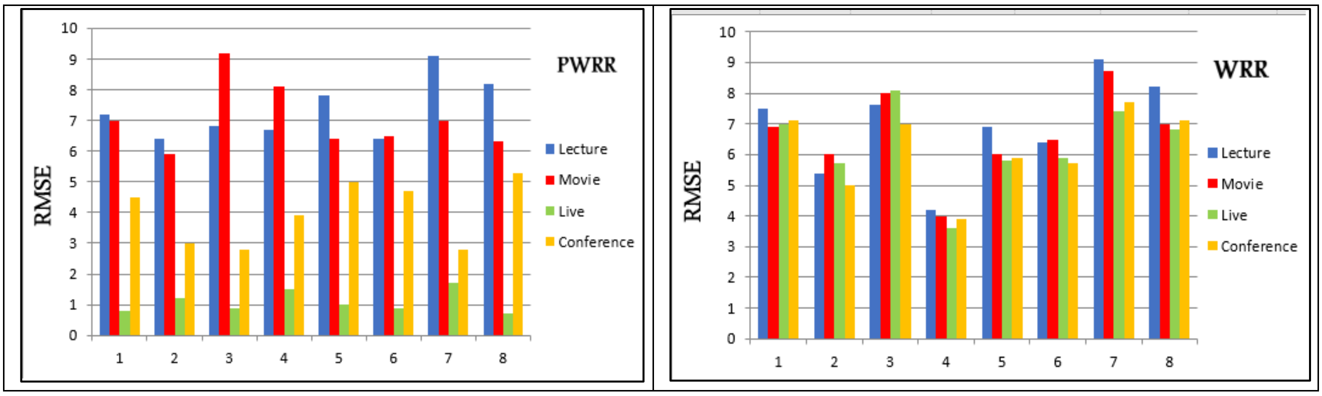

Figure 6. RMSE for video streaming in the proposed architecture

The aim of proposed architecture is to execute streaming video to every client request at the same time with preemptive streaming for real-time video, and make the video quality varying according to client bandwidth. Latency value is impacted by client bandwidth alteration which decreases by high computational of fog computing and the increase of bandwidth. Kashif and Aiman (2017) (1) discussed the increased demand for using multimedia, video streaming, and free- Interactive Multi-View by people on the internet. They provided statistics of uploading and requesting those applications. The authors use the benefits of fog computing to economize and to reduce real-time response, that considered as a suitable solution for the challenges and problems in those applications. While this paper displays execute the streaming of videos over fog computing environment and dispose of some challenges that limit streaming videos on the internet today.

Conclusion:

Fog computing provides interactive and high processing to the sensitive delay application that makes it a suitable solution to video streaming processing in the real-time. The goal of using PWRR scheduled algorithm of the proposed architecture is to give preemptive streaming to the live videos which reduce latency for streaming this video type and at the same time served all other clients' requests in equal time depended on selected time slice. The video frames qualities that have been streamed from the server to the client are measured by QoS measurements. These measurements are used to determine the degradation between the video frames in the server and the frames that received by the client in the streaming process to evaluate the display video quality in the client side. This paper aims to prove that the fog computing is very appropriate to perform streaming for interactive media, which can be seen in the results. It can be declared that this field has not been tackled in other surveys of the edge technology. For the future work, the proposed architecture can be executed on the internet by working in a real fog computing environment, it can be done by reserving on a fog computing domain and display the results from real performance.

Conflicts of Interest: None.

Reference:

1. Kashif B, Aiman E. Edge Computing for Interactive Media and Video Streaming. Second International Conference on Fog and Mobile Edge Computing (FMEC); 2017 May 8-11; Valencia, Spain, IEEE Xplore: IEEE; 2017. 68-73 p.
2. David A. The Technology of Video and Audio Streaming. # Second Edition. United States: Elsevier;2005.357p.
3. Chin FL, Dong YS, Ren HH, Ying XL. A QoS-aware Streaming Service over Fog Computing Infrastructures. Digital Media Industry & Academic Forum (DMIAF) ;2016 July 4-6; Santorini, Greece, IEEE Xplore: IEEE; 2016.94-98p.
4. Ning C ,Yu C, Yang Y, Haibin L, Pengpeng L, Roger Z. Dynamic Urban Surveillance Video Stream Processing Using Fog Computing. 2016 IEEE Second International Conference on Multimedia Big Data; 2016 July 4-6; Santorini, Greece, DMIAF: IEEE; 2016. 105 –112p.
5. Yang H., Guo J., Liang C., Lei Z., Wang C. An Optimization of the Delay Scheduling Algorithm for Real-Time Video Stream Processing, n: Hung J., Yen N., Li KC. (eds) Frontier Computing. Lecture Notes in Electrical Engineering,2016, vol 375. Springer, Singapore
6. Bittencourt LF, Diaz-Montes J, Buyya R, Rana OF, Parashar M. Mobility aware Application Scheduling in Computing. IEEE Cloud Computing. 2017April 26;4(2):2635P.https://ieeexplore.ieee.org/document/7912261/.
7. John GA, Wai- tian T, Susie JW. Video Streaming: Concepts, Algorithms, and Systems. Mobile and Media Systems Laboratory; 2002 September 18th, Mobile and Media Systems Laboratory, HP Laboratories Palo Alto.

8. Padmanabhan S, Chinnagounder C. Study and Simulation of Video Streaming. International Journal of Computers and Applications (IJCA). 2015 July 11; 32(4) (1206-212X): 428-434.
9. Carlos A. Self-healing Techniques for Video-streaming Applications. Coimbra: University of Coimbra; September 2015.
10. Saad K, Simon P, Yongrui Q. Fog computing security: a review of current applications and security solutions. Journal of Cloud Computing: Advances, Systems and Applications.2017 August 16 ;6(1) (2192113X):1-22P.
11. Amir VD, Rajkumar B. Fog Computing: Helping the Internet of Things Realize its Potential. IEEE Cloud Computing. 2016 Aug 15; 49(8): 112 – 116P.
12. Luiz FB, Javier D-M, Omer FR, Manish P. Mobility-aware Application Scheduling in Fog Computing. IEEE Cloud Computing. 2017April 26; 4(2):26- 35P.
13. Ibrahim S, Shamala S, Azmi J, Zuriati AZ. A load-aware weighted round-robin algorithm for IEEE 802.16 networks. EURASIP Journal on Wireless Communications and Networking.2014 December 19; 1-12p.
14. Tomáš B, Denisa L, Martin M. Performance of Round Robin-based Queue Scheduling Algorithms. 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service; 2010 June 13-19; Athens/Glyfada, Greece, IEEE; 2010156-161p.
15. Qureshi I. CPU Scheduling Algorithms: A Survey, Int. J. Advanced Networking and Applications (IJANA), 2014, Volume: 05, Issue: 04, Pages:1968-1973, ISSN: 0975-0290.
16. Lejla R. DASH-based Network Performance-aware Solution for Personalised Video Delivery Systems. Dublin: Dublin City University; January 2016.

خوارزمية الجدولة (PWRR) لمعالجة تدفق الفديو باستخدام حوسبة الضباب

صبا فؤاد حسن **رنا فريد غني**

قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق.

الخلاصة:

يُعد بث الفيديو أكثر الوسائط شيوعًا التي يستخدمها الأشخاص على الإنترنت اليوم ويستهلك الكثير من عمليات نقل الإنترنت. يتم استخدام كمية هائلة من استخدام الإنترنت لتدفق الفيديو الذي ينفق ما يقرب من 70٪ من الإنترنت اليوم. ومع ذلك ، توجد قيود على الوسائط التفاعلية ممثلة في زيادة استخدام النطاق الترددي وتأخره ، مثل بث الفيديو المباشر الذي يتطلب الإرسال في الوقت الفعلي. تستخدم تقنيات حوسبة الضباب للتخفيف من حدة هذه المشكلات من خلال توفير استجابة عالية في الوقت الحقيقي وموارد حاسوبية قريبة من العميل عند حدود الشبكة ، والضباب عبارة عن طبقة وسيطة بين السحابة والمستخدم النهائي ، تقترح هذه الورقة خوارزمية Weighted Round Robin (PWRR) ذات الأولوية لجدولة عمليات التدفق في بنية الضباب لإعطاء استباقية لبث طلب فيديو مباشر لتقديم وقت استجابة أقل للغاية وتواصل في الوقت الفعلي. تعرض نتائج تجربة PWRR في البنية المقترحة لتدفق الفيديوعبر حوسبة الضباب ، تقليل وقت الاستجابة ونوعية جيدة لطلبات الفيديو المباشرة التي تم تحقيقها مع تغييرات النطاق الترددي بالاضافة الى تلبية كل طلبات الاخرى للزبائن في نفس الوقت.

الكلمات المفتاحية: الحوسبة الضبابية، تدفق الفيديو، خوارزمية الاولوية، خوارزمية (WRR) , تقنية DASH