International Journal of Computing and Network Technology
ISSN (2210-1519)
Int. J. Com. Net. Tech. 6, No. 1 (Jan-2018)

http://dx-doi.org/10.12785/ijcnt/060101

A New Unified Communication Approach to comply Bandwidth Optimization Technique using Dynamic Channel Allocation

Md. Julkar Nayeen Mahi[1], Milon Biswas[2], Obonee Kushum[3], Md Whaiduzzaman[4] and Shamim Al Mamun[5]

[1,2]Master of Science, Institute of Information Technology, Jahangirnagar University, Savar – Dhaka; 1342, Bangladesh
[3]Bachelor of Science, Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka; 1216, Bangladesh
[4]Associate Professor, Institute of Information Technology, Jahangirnagar University, Savar – Dhaka; 1342, Bangladesh
[5]PhD Research Assistant, Department of Information and Computer Sciences, Saitama University, Saitama - Japan

Received: 2 Sept. 2017, Revised: 17 Aug. 2017, Accepted: 20 Aug. 2017, Published: (1 Jan. 2018)

Abstract: Expanding requests on a system data transfer capacity is a natural event on genuine circumstances. In diverse cases, the accessible blockage control instrument within reach cannot meet these requests appropriately because of packet misfortunes occurred at client end which results call dropping or buffering. As an answer of these issues the authors have proposed an adjustment of the leaky bucket algorithm referred as proposed algorithm, which is an ameliorate choice for end to end communication. In this proposed algorithm the primary bucket can get the information bundles and after that, it can flow the packets as indicated by the output rate. In the worst case, if the information or input rate surpasses the output rate the container begins to store the packets briefly. At that point when the packets are reached at the farthest point of the accumulated bucket then the call drops which is not achievable for the clients of low bandwidth capacity distributions. Here, the authors depicted about combined communication at the both end of a transmission channel (e.g. wires, ethernet links) through displaying data transmission streamlining or bandwidth optimizing procedure providing dynamic channel allocation. This appropriate proposed approach is a mix of two subsystems that controls transmission rate in a system by means of separating the approaching packets. Later, the procedure handles the transmitted packets with a view to pass all requests via agreeing the yield or output rate. The primary objective of this approach is to guarantee consistent communication unless of the possibility that the output rate goes down than the approaching packet rates because of low system availability. The authors recommend that, this approach is viably competent to decrease the rate of packet misfortunes and call dropping on diverse operating system managements.

Keywords: Bandwidth, Consolidate, Solicitations, Buffering, Packet losses, Leaky Bucket, Dynamic channel

1. INTRODUCTION

The correspondence frameworks by means of web are hindered by restricted transfer speed and loss of packets. The vast majority of the present correspondence framework are running by the leaky bucket algorithmic process. This process is the technique for briefly putting away a variable number of demand and sorting out them into a set-rate yield of packets in a non-concurrent exchange mode network [1]. From this idea it can be expected that the measure of transferred bundles, cells or packets are constantly settled. The leaky bucket workflow depends on the utilization of transmission tokens for every cell getting to the system. In this algorithm the fundamental bucket gets the info parcels or packets and go them through as indicated by output rate. On the off chance that the output rate is not as much as the information rate the basin begin to store the packets briefly [2]. At the point when the packets flood the capacity and reaches the furthest of the principle bucket, it declines or drops any recently arriving solicitations or processed packets and results call dropping [3]. Also the

E-mail: mahi.1992@gmail.com, milonbiswas4702@gmail.com, oboneek@gmail.com, wzaman@juniv.edu, shamim@juniv.edu

execution of leaky bucket algorithm acts truly poor at connected movement with the in-ware of real time [4]. The approach we have utilized as a part of our paper present another framework which is more reliable than leaky bucket algorithm. The proposed algorithm is a blend of two subsystems: i) Separating the approaching solicitations and ii) Consolidating and passing the processed packets as per the yield rate. An essential contrast between the proposed algorithm and the leaky bucket algorithm is a recommended approach that the proposed algorithm initially limits the video packets when the pail is topped off the greater part of it and the process declines just the video bundles or packets when the container is going to overflow, thus will be happened just in the most pessimistic scenario to guarantee a consistent correspondence [5]. However the proposed process never disposes of all packets while leaky bucket disposes of every single approaching solicitation or packets when the bucket or pile is full [6].

A. Dividing the incoming requests

In the proposed algorithm authors have utilized three aggregate pails named video pail, Audio pail and primary pail or bucket for holding the approaching solicitations or parcels [7]. In the algorithmic process the approaching solicitation bundles the Audio packets and the video packets and then do an isolation between each other for sending them into their separate buckets [8]. In the meantime these two pails pass their parcels to the fundamental buckets and stored them if necessary.

B. Merging and passing the requests

After getting packets from the initial two containers, the primary bucket consolidates them and pass them through as indicated by the yield rate of the opposite side. If the output rate is less than the input rate the primary bucket begins to store the approaching parcels. On the off chance that the bundles top off 1/2 of the capacity size of the principle bucket the video container begins to resize the resolution of the video parcels [9]. When it tops off to 3/4 of the fundamental bucket, it denies the video parcels by killing the video bucket therefore the video correspondence will be halted yet the audio correspondence will continue as before as it was some time recently [10]. It is the most pessimistic scenario of this algorithm to stay away from call dropping. This criteria may give a consistent correspondence or communication [11].

At that moment, when the yield rate of the principle bucket will be equivalent to the input rate then it will get again the video parcels from its particular pail with a limited determination or resolution until the stored packets compasses to 1/2 of the primary basin or bucket size [12]. Furthermore, when the capacity estimate goes down to the 1/2 of the primary basin size, it will get the video packets from the video bucket with its real determination and gives normal, constant, feasible and easy to use correspondence [13].

C. Description of working process and figure of the proposed algorithm

This algorithm incorporates three pails named audio bucket, video bucket and the last one is principle bucket. From the approaching solicitations the audio bundles and the video packets [14] will be isolated from each other and sent to their separate containers. In the meantime these two buckets pass those bundles to the primary one. On the other hand, getting these parcels from the following employment of the fundamental bucket are blended and send them to the client end concurring the output rate. In the event that, if the input rate surpasses the yield or output rate, fundamental bucket will begin to store the additional packets briefly until it is going to flood.

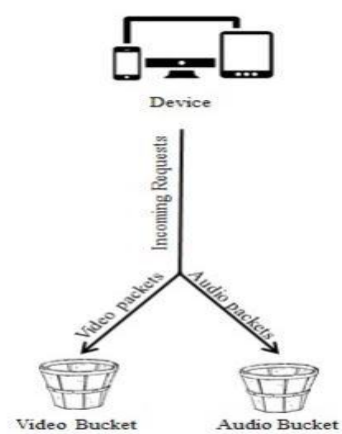

Figure 1. Dividing the incoming requests

If the packets top off more than 1/2 of the capacity size of the primary bucket, the video bucket [15] begins to resize the resolution of the video packets until the point when it cross the 3/4 of the fundamental bucket or tumbles down to the 1/2 of it. The audio packets will continue to its remaining sizes as it was before. After that process the principle bucket will combine them and exchange those packets concurring to the output or yield rate.

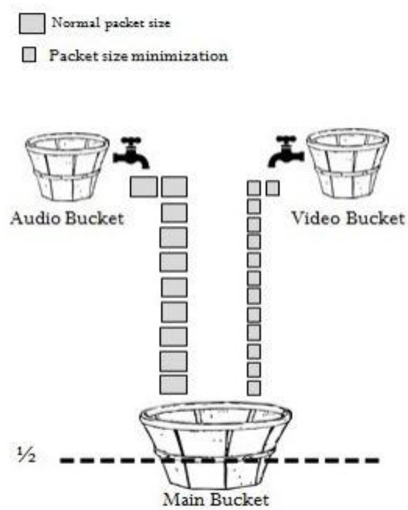

Figure 2. When incoming data packets fill up to ½ of the main bucket

At the point when the packets will top off to 3/4 of the principle bucket, the bucket then will kill the process into and out of accepting video packets from the video bucket. This determinative performance itinerary technique can be termed as dynamic channel allocation.

Afterwards, the video communication will be unavailable on that particular time [16]. But the audio communication won't be obstructed. It is the worst case scenario of this algorithm to keep away from call dropping [17] and to give a steady correspondence for quite a while.

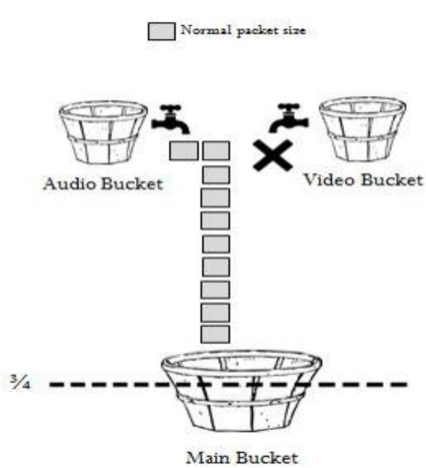

Figure 3. When stored packets fill up to 3/4 of the main bucket

At the point when the yield rate will be expanded and the stored packets will go down and achieves the 3/4 of the primary bucket, it will begin the process with a view to get the video packets from its container or bucket, and in addition the video correspondence then will be additionally accessible [18].

Since every last time, the approaching packets check as far as possible or the circumstance of the fundamental bucket, Aftermath of, it will be easy to donate appropriate service [19] according to the output or yield rate.

2. Related works

Shah, Junaid Latief, and Javed Parvez et al. describes three queuing mechanism in traffic shaping section considering IPv6 protocol [20]. They have mentioned three types of queuing mechanism like- FIFO, priority queue, weighted fair queue. Authors also consider the attributes in delay types, like – end to end delay, jitter buffer, packet delay variation, total data send and received under different type of delay measurement within a runtime simulation of 600 seconds in case of voice and video conferencing. The authors have described the difference among the queuing mechanisms. FIFO, considers less data transfer than priority queue and weighted fair queue [21] [14] in case of voice transmission. But priority queue and weighted fair queue occur very high percentage of packet loss considering voice. Priority queue [22] performed well compared to FIFO and weighted fair queue in case of video conferencing. FIFO considers less packet loss among all these queuing mechanism. In case of both voice and video the delay variation is far less in priority queue. In case of jitter buffer the rate and delay for voice is high. The main objective of this work is to initialize the differences between queuing algorithms [23].

Kozacinski, Hrvoje, and Petar Knezevic et al. describes a comparative study by developing QOS (quality of service) mechanisms over computer networks through building a software system. The software gives a statistical review by establishing the algorithms stated as – buffer strategy round robin, token bucket, packet delays for specific buffer time under considering the parameters like- packet delay, packet priority, packet interval time throughout the system [24]. The main goal is to establishing and enriching the QOS within computer networks that will occur less delay, less packet loss throughout the system by building higher performance considering network traffic. The QOS has main three attributes- guaranteed QOS, QOS network managing policies, QOS mechanism selection [25]. The software uses QOS for less network traffic congestion and it has the function of dump bell topology, different QOS with track buffer capacity, packet loss and time delay. Simulations are done with retransmission, buffer, buffer strategies, packet priorities, and token bucket algorithm properties [26] [15]. The simulation also done considering TCP, VoIP, video traffic, Adhoc network protocols for more realistic comparison. The main results of the paper shows that, a clear solution to QOS in communication networks does not exists and the solution greatly depends on the network itself and the individual needs of the particular network. The paper, describes if fast packet transfer needed regardless of packet priority, the round robin buffer strategy [27] shows better result. If packets need different priorities or packets that are more sensitive to delay then the packets have to use the combined token bucket shaping algorithm. The main motive of this paper is to show that network traffic improves by improving QOS mechanisms and making the simulations of complex system with real life application to emphasize that QOS affect packet delay [28] and buffer congestion properties.

3. BUCKET SIZE AND OUTPUT RATES

Authors have utilized three buckets in this new way to deal with a consistent correspondence for the clients by limiting the rate of packet's misfortune and keeping the odds of call dropping. In this algorithm [29], it is truly vital to comprehend and pick up a reasonable idea about the span of these three buckets and their individual yield or output rate [30].

The span of these buckets might be the same or the primary bucket may will be biggest than the others, but the yield rate follow some dictated requirements (e.g. packet sizes for both video and voice, memory allocation on random access memory and cache, packet estimation relay on cache etc.) to accomplish the objective or the necessities [31].

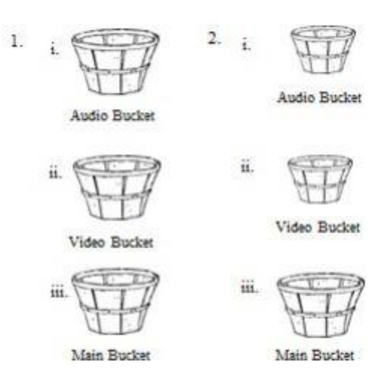

Figure 4. Bucket sizes comparing from Audio to Video and Main

As indicated by our algorithmic procedure, the measure of the principle bucket (e.g. variable characterized functions for parameters) is greater in extent because if the aggregate size of audio and video container surpasses the fundamental bucket size then it will topped off rapidly and that may intrude on the correspondence procedure [32]. The yield or output rate of the fundamental bucket is only the blend of the other two containers, named audio container and video container [33].

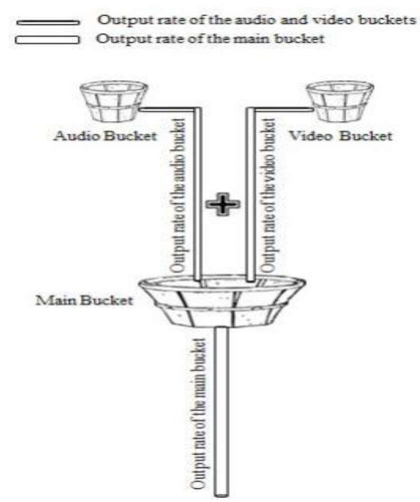

Figure 5. Scenario of output rates

If the all three buckets have the same output rate then it is obvious that those packets are to be overflown. That is the reason for which the authors determined the output rate to lessen the chance of packet's flooding during the communication.

4. PROPOSED ALGORITHM AND FLOWCHART

The algorithm works through measuring specific parameters for both the audio and video packets. For audio the parameter ranges from 0 to 2000 and after 2000 – 4000 it serves for the video packets. But in the worst cases, the algorithm reshapes the video packets and mixes with audio packets to provide steady communication. The authors used specific parameter modelling in this work through remembering CPU workloads. To give a reasonable consideration of this proposed algorithm, here authors described a flow chart. It will visualize the entire communication framework that is specified in this paper.

Here,

Main Output = R	Video Bucket $= Y$
Voice Bucket = X	Main Bucket $= M$
Iteration = n	Storage size $= N$

Threshold 1 = T Threshold 2 = H

Threshold value 1 = TO Threshold value 2 = TH

Packet counter = C Voice packet priority = VP

First the algorithm checks for voice packet sessions and gather the algorithmic defined packets.

Therefore the relation can be described by set builder notation that is -

$$C \cap X = \{ x: x \in C \text{ and } x \in X \text{ where, } C \leq 2000\}$$

In second step, the algorithm checks for video packet sessions and gather the algorithmic defined packets.

Therefore,

$$C \cap Y = \{ x: x \in C \text{ and } x \in Y \text{ where, } C \leq 4000\}$$

In third step when all the packets gathered and send towards the main bucket for calculating the TO value, (M/2 < N) and processed the next session, then relation of the elements depends on -

$$TO = M \supseteq (X*Y) = \{x: x \in M \text{ and } x \notin (X*Y)\} ,$$

$$\{(x , y): x \in X \text{ and } y \in Y\}$$

In fourth step while the main bucket still on the nature of processing packets to deliver at the end point and calculate threshold value 2, [3*M/4 (bucket session) < N] then the relation also remains the same which is,

$$TH = M \supseteq (X*Y) = \{x: x \in M \text{ and } x \notin (X*Y)\} ,$$

$$\{(x , y): x \in X , y \in Y\}$$

In high load when $M = X$, which means voice packet priority scheme is ongoing at that moment the elementary set relation can be expressed as -

$$VP = \{x: x \in M \text{ and } x \in X\},$$

$$\{x: x \in M \text{ and } x \in Y\} \text{ where } X > Y$$

After merging and estimating all the packets the algorithmic relation of the overall variables and elements of this process redirects in set theory as-

$$R \subseteq M = M \supseteq (X*Y) ,$$

$$M \supseteq R = \{x: x \in M \text{ and } x \notin R\}$$

Let 'M' is the main bucket and the sum of other two bucket output rates are considered as X (voice) and Y (video) then the mathematical equation can be predicted as,

$$\sum_0^n \{M*N\} = \sum_0^n \{X*N\} + \sum_0^n \{Y*N\}$$

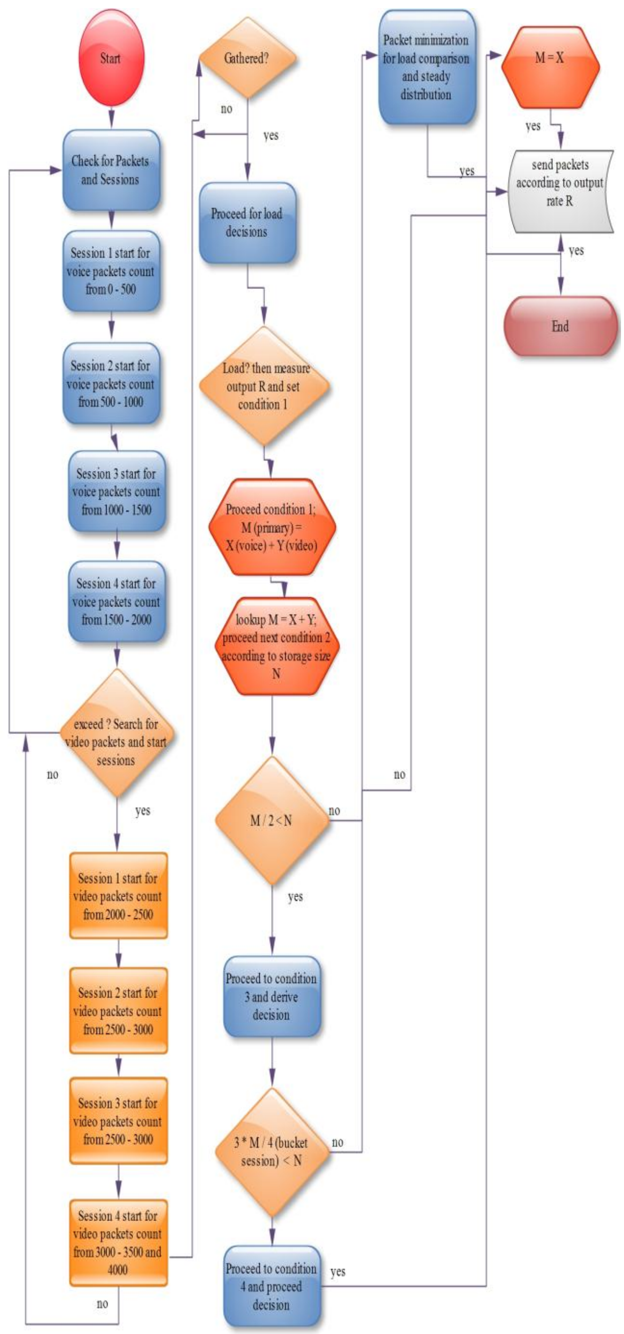

Figure 6. Workflow chart of the proposed algorithm

In this stream chart, authors demonstrated that when packets are send from the video and audio bucket then converged into the fundamental bucket [34], the process checks firstly if the joined packets are surpassing the 1/2 of the primary bucket's size or not, if not then it transfers the combined packets as indicated by the output rate. On the other hand, if yes then it checks that unless the joined packets surpasses the 3/4 of the principle bucket's size or not. In the affirmative processing of the algorithm, the immediate and next packet combines

[35] through upholding only the unchanged audio packet and refuse the video packet for the next transitions. In the negative processing, it reduces the resolution of the next video packets merging with the upcoming unchanged audio packet in order to transfer the combined packet to the user-end [36].

In this algorithmic process, each time every packet counts in a particular session and checks the present circumstance of the main bucket [37]. After that the process delivers the audio and video packet through merging in act and sends to the end point of the communication system. As a result, the possibility of call dropping, buffering and packets misfortune are very lower according to specific packet size or call duration on particular environment loads. (e.g. Windows 8.1 Pro, Kali-Linux, Debian Server, Red-Hat Linux Server, Ubuntu 14.04 LTS, Microsoft Windows Server 2012)

5. Leaky Bucket Algorithm vs Proposed Algorithm

In leaky bucket algorithm, if the output rate is not as much as the input rate of the bucket then after certain measure of packet storing, flooding attempt of packets [38] occurs that means there are no free space for the new approaching requests, called as packet loss. Reasoning this the steady communication hinders and drops the call by decimating the session.

In leaky bucket algorithm if the output rate of the bucket is not as much as the input rate then after certain measure of packet storing it comes about packets flooding that implies there are no free space for the new approaching requests, called as packet loss. Reasoning this the steady communication hinders and drops [39] the call by decimating the session. The authors claimed in this work that, the chance of call dropping is very rare and lower in specific criteria. The fundamental focus of this algorithm is to proceed with the discussion or the communication as far as possible [40]. Despite of the fact that when the input rate surpasses the output rate [41] or the output rate descends. Just even under the least favorable conditions [42] case, when the primary bucket is close to flood it rejects the video packets and gets just the audio packets to proceed with the communication framework [43] or process. In leaky bucket algorithm [42] [43] packets misfortune are expanding with time.

6. Experimental Results

The experiments were performed and throughput and memory complexities were measured using TCP (SIP) and UDP (SIP) protocol in an IEEE 802.3i customize network. Iterations were gradually increased in numbers from 1000 to 7000 for both TCP and UDP on long term system run. The resulting throughput measurement, time delay, memory complexities on both RAM (random access memory) and cache values were plotted in histograms. These results were gathered from Windows 10, Windows 7, Kali Linux, Ubuntu 14.0 LTS server, Red hat Fedora Linux Server etc.

Expressed underneath in Figure 7. the Y hub portrays “Packet Loss” and the X pivot delineates “Time (minutes)”. The amount of packets misfortune is 30 in first second. Subsequent to following 8 minutes, the amount of packets setback reaches to 360. Be that as it may, in the proposed algorithm [44], the rate of packet setback have little changes with the time.

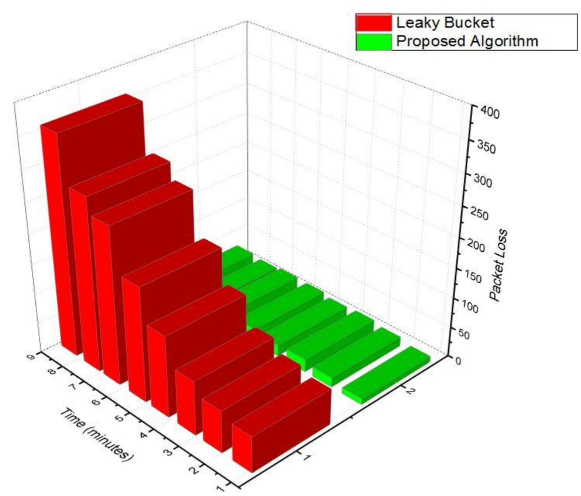

Figure 7. Bandwidth or throughput measurement scenario at 512 kbps

In Figure 8. the Y pivot delineates “Packet Loss” and the X hub portrays “Time (minutes)”. Going through this underneath outline, authors show the circumstance if the information exchange limit speed reaches to 1Mbps, by then the rate of packet misfortune for leaky bucket calculation still goes higher than the developed algorithm.

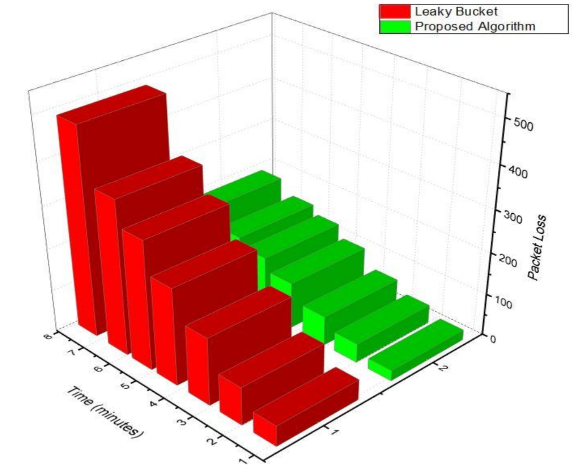

Figure 8. Bandwidth or throughput measurement scenario at 1Mbps

In Figure 9. the Y hub portrays “Packet Loss” and the X pivot delineates “Category of Intel Processors”. The authors have made a custom program in this case for checking the MMU (memory management unit) addressable memories and secondary memories (RAM) to process real-time packets on different Intel family microprocessors. In Figure 9. we can see that from sixth to fourth generation i7, i5, i3 processors at each case in leaky bucket, the packet drop [45] is higher in category contrasting with the proposed algorithm.

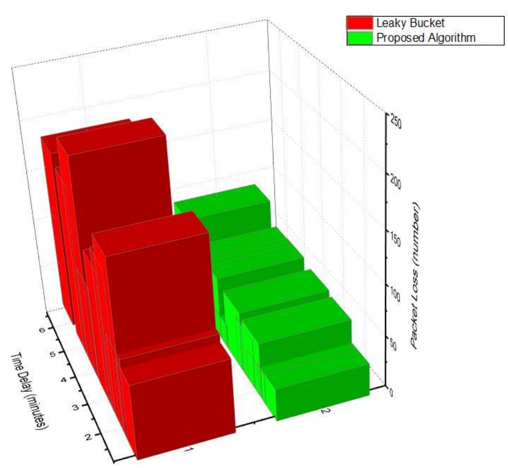

Figure 9. Algorithmic memory management on processor caches

In Figure 10. the Y pivot delineates “Packet Loss” and the X hub portrays “Memory Engage [Cache + RAM]”. Here, various operating systems aggregates with different memory managements based on their processing techniques (e.g. middleware). In below Figure 10. different operating systems [46] are coupled with primary and secondary memory mainly based on omnipresent fashion. As per our committed MMU programming process it is clearly portrayed that Linux systems offers less memory utilization comparing to the Windows systems.

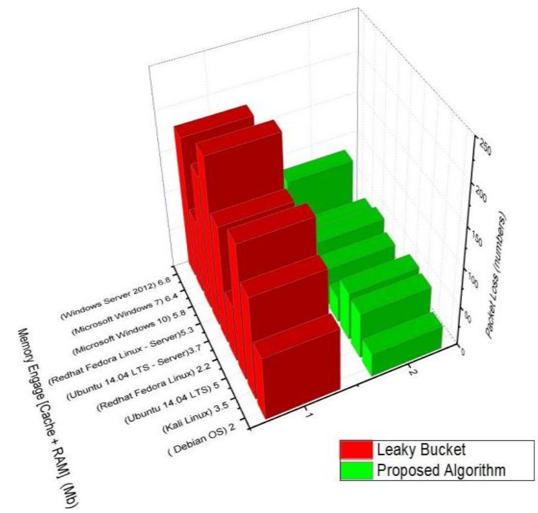

Figure 10. Memory engagement on both cache and RAM

In Figure 11. the Y pivot delineates “Packet Loss (number)” and the X hub portrays “Time delay (minutes)”. Figure 11. describes that in different operating system the algorithmic process takes various time delay on the basis of processing strategy. In this session, Windows server shows more time delay among the Linux systems for processing 1000 packets. Actually the time delay depends on various criteria’s of processing each session [47] work in an operating system. Graphical real-time data causes extra time delay instead of streaming a binary text of large string. Apparently it can be told that, in leaky bucket the packet losses are higher than the proposed algorithmic scheme.

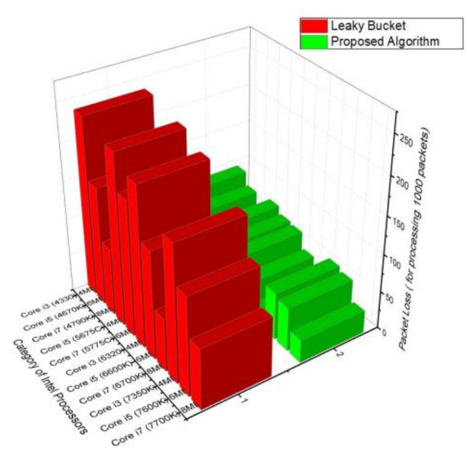

Figure 11. Time delay on processing 1000 packets

The distinction between space complexity and time complexity is that space (memory) can be reused (e.g. amount of computer memory required during the program execution, as a function of the input size). Here, authors have utilized central processing unit benchmark properties for getting the outcome. The property relies on "Dhrystone" benchmark that includes the program execution process in MIPS (million instruction per second). In Figure 12. the three Y hub delineates “MIPM (million instructions per minutes)”, “Secondary Memory (RAM)”, “Time (per minutes)” and the X pivot portrays “Iterations in OS”. The overall work follows our custom made direct voice packet priority queuing usually and also in the worst case scenario. It is a nondeterministic time memory property that utilizes O (n) space. From the first attempt to the second attempt in Figure 12. shows that, whatever the attempt or emphasis is, MIPM begun to ascend at an augmenting level as more processes are closed by.

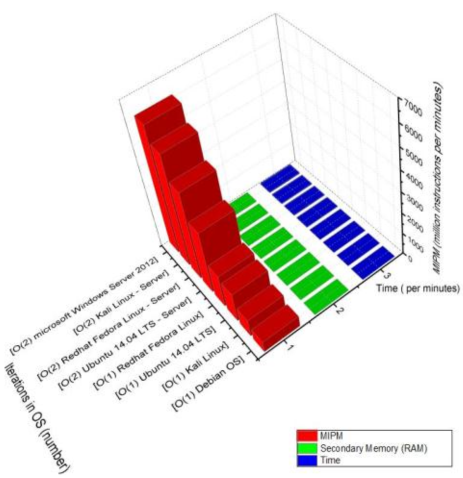

Figure 12. Space complexity in secondary memory management

Contrasting with the emphasis of processes, processing time and secondary memory engagements are very lower, usually 2 to 8 minutes and 3 to 8 GB secondary memory engagement for this specific proving ground situation. That implies, that the process will continue until feasible addressable memory locations [48] are accessible to take a shot at particular sessions.

Considering the results a performance table is given below showing three major specific attributes:

TABLE I. PERFORMANCE COMPARISON

Criteria	Comparison Among Three Specific Attributes		
	Leaky Bucket Algorithm	***Proposed Algorithm***	***Cache and RAM Management (Proposed Algorithm)***
Time Delay in OS (1500 to 3000 Packets/hour)	93.6 %[a.]	33.8 %	9.09 %
Call Dropping in OS at 512 Kbps throughput (1000 Packets in 8 minute)	1.48 %	1.31 %	0.10 %
Call Dropping in OS at 1 Mbps throughput (1000 Packets in 8 minute)	2.48 %	1.58 %	0.064 %
Memory Engadgement in OS (processing 1000 packets on 24 GB)	0.37 %	0.90 %	2.65 %

a. **Performance comparisons are shown in this table is got from above stated bar graphs .**

Figure 7. to Figure 12.

7. Limitations And Future Work

The significant impediment of this work is, the algorithm separates the voice and video packet to alleviate the movement stacks or traffics on particular sessions. But it cannot provide voice and video packets simultaneously on network channels having loaded traffic. The authors provide here a customized digital channel method estimating packet levels for particular sessions (e.g. 1000 packets for first voice session, 1500 from second voice, 2000 from third voice and starring video packets from 3000 then second session video packets from 3600, third session from 4000 packets). In any case, in particular channel with heavy traffic this algorithm, processed video packets through lessening sizes to maintain steady communication. In this work, memory utilization is a major issue. For this work we appraise the secondary memory measure as 24 GB yet in higher operation the memory may lead twofold or triple to overcome constant information memory handling lags [49]. The memory utilization can be a never ending phenomena in this situation. Despite what might be expected, this circumstance can emerge enormous equipment costing. An efficient memory allocation is a desperate requirement for the time being and furthermore later on to do these sort of works [50]. As time passes, we need more devices to communicate efficiently with each other persons in that case, proper memory allocations or allotments are another name of necessity.

In this work the authors have built up a FIFO queuing based voice packet priority mechanism to follow high traffic loads on digital channel assignment plans [51]. In this manner the framework gives consistent transmission. In future priority on session basis queuing, shortest job first queuing priority, non - preemptive priority on session basis, last come fast serve, preemptive priority with quick sort, bucket sorting with shortest job first preemptive are some planning procedures that can be obtain through the algorithm to diminish the memory complexities [52] of the system. Using some specific priority based scheduling one can make a less memory usage property that can be a better criteria for future to reduce memory utilization.

8. CONCLUSION

In this Paper, authors proposed another approach for joined communication with data transfer capacity advancement or bandwidth optimization strategy through performing dynamic channel circulation. The issues of traditional communication pattern can be utilized by this algorithmic structure. The created procedure can redesign the proficiency and plausibility of the existing systems. Additionally it guarantees a consistent communication strategy without being hindered in audio region. The rule target of this algorithm is to continue with the communication regardless of the possibility that the output rate goes down than the approaching packets rates, because of low network connectivity. In this paper, it is additionally depicted that how this approach can reduce the rate of packet misfortunes, buffering and call dropping in dedicated benchmark criteria.

ACKNOWLEDGMENT

Foremost, I would like to express my sincere gratitude to my advisor Shamim Al Mamun for the continuous support, his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of research and writing of this paper.

Besides my advisor, I would like to thank BUBT chairman Prof.M. Ameer Ali, Redwanul Islam, Milon Biswas and Obonee Kushum for their instantaneous support during the ongoing work.

REFERENCES

[1] What is leaky bucket algorithm? - Definition from WhatIs.com. (n.d.). Retrieved August 21, 2017, from http://whatis.techtarget.com/definition/leaky-bucket-algorithm. (reference web)

[2] J. N. (n.d.). "Limitations" and "Pitfalls" of Leaky Bucket A Study with Video Traffic.
doi:10.1107/s0108270100002845/bm13933bsup6.hk. (reference web)

[3] Berry, Alfred E., et al., "Integrated audio and video agent system in an automatic call distribution environment", U.S. Patent No. 6,404,747.11 Jun. 2002.

[4] Gough, Michael, "Video Conferencing over IP: Configure, Secure, and Troubleshoot: Configure, Secure, and Troubleshoot", Elsevier, 2006.

[5] Medhi, Deepankar, "Network routing: algorithms, protocols, and architectures", Morgan Kaufmann, 2010.

[6] Elhanany, Itamar, "High-performance packet switching architectures",Springer, 2007.

[7] Campbell, Robert Craig, et al., "Videophone and method for a video call", U.S. Patent No. 7,404,001. 22 Jul. 2008.

[8] Tryfonas, Christos, et al., "Algorithms for Deterministic Call Admission Control of Pre-stored VBR Video Streams", Journal of Multimedia 4.4 (2009): 169-181.

[9] Nakibly, Gabi, "Traffic Engineering Algorithms for IP and MPLS Networks: Novel and practical algorithms for routing optimization of large operational networks", Scholar's Press, 2014.

[10] Farzanegan, Mahmoud Daneshvar, Hossein Saidi, and Mehdi Mahdavi,"A Scheduling Algorithm for Bursty Traffic: Controlling of Service Rate and Burst", Arabian Journal for Science and Engineering 39.6 (2014): 4753-4764.

[11] Grimm, Christian, and Georg Schlchtermann, "IP-traffic Theory and Performance", Springer, 2008.

[12] Akyildiz, Ian F., et al., "A roadmap for traffic engineering in SDN Open Flow networks", Computer Networks 71 (2014): 1-30.

[13] Dainotti, Alberto, Antonio Pescape, and Kimberly C. Claffy, "Issues and future directions in traffic classification", Network, IEEE 26.1 (2012): 35-40.

[14] Shah JL, Parvez J. Evaluation of queuing algorithms on QoS sensitive applications in IPv6 network. InAdvances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on 2014 Sep 24 (pp. 106-111). IEEE.

[15] Kozacinski H, Knezevic P. An approach using simulation techniques to estimate quality of service parameters in communication networks. In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on 2014 May 26 (pp. 1642-1647). IEEE.

[16] Sinha, Amritanshu, Tapas Saini, and S. V. Srikanth, "Distributed computing approach to optimize road traffic simulation", Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference on. IEEE, 2014.

[17] Nguyen, Hung, et al., "How to build complex, large-scale emulated networks", Parallel, Distributed and Grid Computing(PDGC), Testbeds and Research Infrastructures. Development of Networks and Communities. Springer Berlin Heidelberg, 2011. 3-18.

[18] Amaldi, Edoardo, et al., "Energy management in IP traffic engineering with shortest path routing", World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a. IEEE, 2011.

[19] Xie, Haiyong, Guangyu Shi, and Pengwei Wang, "TECC: Towards collaborative in-network caching guided by

traffic engineering", INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.

[20] Hakusui, Shigeaki, "Virtual PBX based on feature server modules", U.S. Patent No. 8,279,857. 2 Oct. 2012.

[21] Guy, Pujolle, "Metamorphic Networks", Journal of Computing Science and Engineering 7.3 (2013): 198-203.

[22] Scholler, Marcus, et al., "Resilient deployment of virtual network functions", Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on. IEEE, 2013.

[23] Conway, Kelly, et al., "Method and system determining the complexity of a telephonic communication received by a contact center", U.S. Patent No. 8,023,639. 20 Sep. 2011.

[24] Quislant, Ricardo, et al., "Conflict Detection in Hardware Transactional Memory", Transactional Memory. Foundations, Algorithms, Tools, and Applications. Springer International Publishing, 2015. 127-149.

[25] Schmitz, Michael, et al., "Interaction modalities for digital product memories", SemProM. Springer Berlin Heidelberg, 2013. 261-279.

[26] Ke Xu, Wenlong Chen, Chuang Lin, Mingwei Xu, Dongchao Ma, and Yi Qu" Toward a Practical Reconfigurable Router: A Software Component Development Approach" IEEE Network September / October 2014.

[27] Arash Farhadi Beldachi*, Mohammad Hosseinabady**, Jose L. Nunez-Yanez" Configurable Router Design for Dynamically Reconfigurable Systems based on the socwire noc" International Journal of Reconfigurable and Embedded Systems (IJRES)Vol. 2, No. 1, March 2013, pp. 27~48ISSN: 2089-4864.

[28] Ying-Cherng Lan, Student Member, IEEE, Hsiao-An Lin, Shih-Hsin Lo, Yu Hen Hu, Fellow, IEEE, and Sao-Jie Chen, Senior Member, IEEE" A Bidirectional noc (binoc) Architecture with Dynamic SelfReconfigurable Channel" ieee transactions on computer-aided design of integrated circuits and systems, vol. 30, no. 3, march 2011

[29] Sinduri. M, N. Nagaraja Kumar" Design and Implementation of Reconfigurable Router Architecture for Dynamic noc's" IJESC Issue No: 10 Volume No: 1 ISSN: 2250-1371 Paper IJESC116

[30] Malathi Veeraraghavan, Senior Member, IEEE, and Xuan Zheng, Student, IEEE" A Reconfigurable Ethernet/SONET circuit-based metro network architecture" ieee journal on selected areas in communications, vol. 22, no. 8, october 2004

[31] Yong-Kee Yeo, Member, IEEE, Jianjun Yu, and GeeKung Chang, Senior Member, IEEE" A Dynamically Reconfigurable Folded-Path Time Delay Buffer for Optical Packet Switching" ieee photonics technology letters, vol. 16, no. 11, november 2004.

[32] Zhiliang Qian, Student Member, IEEE, Syed Mohsin Abbas, and Chi-Ying Tsui, Senior Member, IEEE "fsnoc: A Flit-Level Speedup Scheme for NetworkonChips Using Self-Reconfigurable Bidirectional Channels" ieee transactions on very large scale integration (vlsi) systems, vol. 23, no. 9, september 2015.

[1] Stoica I, Shenker S, Zhang H. Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. ACM; 1998 Oct 1.

[2] Jain R. A timeout-based congestion control scheme for window flow-controlled networks. IEEE Journal on Selected Areas in Communications. 1986 Oct;4(7):1162-7.

[3] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," IEEE/ACM Transactions on Networking.

[4] (October 2006), "Is IEEE 802.11 ready for VoIP?" 8th International Workshop on Multimedia Signal Processing (IEEE MMSP), Victoria, Canada.

[5] E. W. Fulp (2008). "Congestion Control", Department of Computer Science Wake Forest University, North Carolina. Available:www.cse.wfu.edu/~fulp/csc343/congest.pdf (reference web)

[6] Semeria, C., Supporting differentiated service classes: queue-scheduling disciplines, Juniper networks, 2001.

[7] Ray Williams (nd)," Tackling push-to-talk challenges in handsets",Email: ray.williams@ttpcom.com (reference web)

[8] Rahman MT, Mahi MJ, Biswas M, Kaiser MS, Al Mamun S. Performance evaluation of a portable PABX system through developing new bandwidth optimization technique. In Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on 2015 May 21 (pp. 1-5). IEEE.

[9] Rahman MT, Mahi MJ. Proposal for SZRP protocol with the establishment of the salted SHA-256 Bit HMAC PBKDF2 advance security system in a MANET. In Electrical Engineering and Information & Communication Technology (ICEEICT), 2014 International Conference on 2014 Apr 10 (pp. 1-5). IEEE.

[10] Mahi MJ, Rahad KA, Biswas M, Islam R, Chowdhury ZI. An Accident Detection System for a Single VANET at Low Cost Module. IEEE TechSym, volume 4, issue 1, page 54, december 2016.

[11] Mozumder, Deba Prasead; Mahi, Md. Julkar Nayeen; Whaiduzzaman, Md. Cloud Computing Security Breaches and Threats Analysis. International Journal of Scientific & Engineering Research, volume 8, issue 7, page 1287 - 1297), july 2017.

[12] Whaiduzzaman M, Haque MN, Rejaul Karim Chowdhury M, Gani A. A study on strategic provisioning of cloud computing services. The Scientific World Journal. 2014 Jun 15;2014.

[13] Ahmed E, Akhunzada A, Whaiduzzaman M, Gani A, Ab Hamid SH, Buyya R. Network-centric performance analysis of runtime application migration in mobile cloud computing. Simulation Modelling Practice and Theory. 2015 Jan 31;50:42-56.

[14] Qi H, Shiraz M, Gani A, Whaiduzzaman M, Khan S. Sierpinski triangle based data center architecture in cloud computing. The Journal of Supercomputing. 2014 Aug 1;69(2):887-907.

[15] Whaiduzzaman M, Gani A, Anuar NB, Shiraz M, Haque MN, Haque IT. Cloud service selection using multicriteria decision analysis. The Scientific World Journal. 2014 Feb 13;2014.

[16] Bhuiyan KA, Whaiduzzaman M, Nasir MK. Efficiency and Performance analysis of routing protocols in WSN.

[17] Arefin AS, Rahman MA, Kabir MM, Alam AM, Al Mamun S, Rahman ML. An Efficient Approach for Handling Large Arrays in Parallel Computing Environment Using Bitonic Sorting. Asian Journal of Information Technology. 2005;4(11):1027-30.

[18] Al Mamun S, Kaiser MS, Ahmed MR, Islam MS, Islam MI. Performance analysis of optical wireless communication system employing neuro-fuzzy based spot-

diffusing techniques. Communications and Network. 2013 Sep 27;5(03):260.

[19] Kaiser MS, Al Mamun S, Ahmed KM. Performance analysis of ANC based MIMO cooperative relaying network over Nakagami-m fading channel. InElectrical & Computer Engineering (ICECE), 2012 7th International Conference on 2012 Dec 20 (pp. 393-396). IEEE.

[20] Al Mamun S, Suzuki R, Lam A, Kobayashi Y, Kuno Y. Terrain Recognition for Smart Wheelchair. InInternational Conference on Intelligent Computing 2016 Aug 2 (pp. 461-470). Springer International Publishing.

Md. Julkar Nayeen Mahi has successfully completed his B.Sc. degree in Information Technology, Jahangirnagar University, Savar – Dhaka; 1342, Bangladesh and currently studying in M.Sc at the same institute. He has two conference papers at IEEE engineering community and the databases are indexed at ieeeXplore. In Sep, 2016 he attended the IEEEtechSym conference that was driven by IIT, Kharagpur, West Bengal 721302, India on the banner of "Transforming lives through the power of technology. He is a passionate researcher. His current research interests are mainly in Distributed computer networks, Embedded systems, IoT, Data mining, Cloud computing, Prediction and analysis of Big Data, Operating Systems Scheduling approach, Wireless Sensor Networks, Computer and microprocessor architecture, but not in limited to, Digital logic design, Software defined networking, Image processing, Computer vision, Stochastic process, Probability estimations, Data structures and Bio - informatics.

Milon Biswas has successfully completed his B.Sc. degree in Information Technology, Jahangirnagar University, Savar – Dhaka; 1342, Bangladesh and currently studying in M.Sc at the same institute. He has one conference paper at IEEE engineering community. His current research interests are mainly in Computer networks, Image processing and Cloud computing.

Obonee Kushum is currently a B.Sc. second year student in Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT), Dhaka; 1216, Bangladesh. Her current research interests are mainly in Communication networks, Image processing, Ubiquitous computing.

Md Whaiduzzaman completed his undergraduate degree in Electronics and Computer Science with first class from Jahangirnagar University and he obtained MSc degree in Telecommunication and Computer Network Engineering from London South Bank University, UK. He also completed his PhD degree and research assistant in a High Impact Research (HIR) in Mobile Cloud Computing Research Group, Faculty of computer science and Information Technology, University of Malaya, Malaysia. Presently, he serves as an Associate Professor in the Institute of Information Technology (IIT), Jahangirnagar University. His research interests are Mobile Cloud Computing, Vehicular cloud computing, Fog Computing, Mobile and wireless communications, Security issues. Already Md Whaiduzzaman has published in high impact factors journals including Elsevier, Springer and IEEE Transactions. He received the Elsevier JNCA best paper award 2016 in Paris, France recently.

Shamim Al Mamun completed his B.Sc. in Electronics and Computer Science from Jahangirnagar University and obtained the M.Sc. degree in Optical Fiber Networks from Bangladesh University of Engineering and Technology, (BUET); Bangladesh. He is an Assistant Professor in Institute of Information Technology (IIT), Jahangirnagar University. Currently he is pursuing his PhD from Graduate School of Science and Engineering, Saitama University, Japan. His current research interests are mainly in Computer vision, Robotics, Image processing, Wireless optical communications but are not limited to Communication networks and Big data managements.