

Probability Admission Control in Class-based Video-on-Demand System

Sami Alwakeel and Agung Prasetijo

Department of Computer Engineering
College of Computer and Information Sciences, King Saud University
Riyadh, Kingdom of Saudi Arabia
swakeel@ksu.edu.sa, agungbp@ksu.edu.sa

Abstract— Admission control is applied to maintain a predefined Quality of Service in online streaming services such as Video-on-Demand system. In common admission control, a new request to a specific class will be rejected when the ports dedicated to such a class are fully-occupied. This proposed system is different from previous works as follows: (1) we experimented on the advent of probability for this fully-occupied class so that it might use other class' ports which are still available. (2) Every higher priority class (popular class) has higher probability to get admitted in lower priority class (less-popular class). Conversely, lower priority class has lower probability to get admitted in higher priority class. It is expected that our proposed system will not only increase the performance of popular class but also the performance of overall system. The proposed admission control policy is validated through simulation using NS-2 simulator. In general, with such experiments, it is shown that blockage has significantly been reduced for popular class and also for overall system.

Keywords— admission control; probability; Quality of Service; Video-on-Demand

I. INTRODUCTION

Given a limited resources for serving clients of a Video on Demand service, provider must have a predefined Quality of Service (QoS) in order for maintaining the quality of video distributed to clients. Rejection of client request must be conducted as traffic must be maintained not exceeding maximum available resources. We consider here that the resources are related to available ports. However, the resources mentioned here can be of data transfer bandwidth of disks, or of disk parameters such as rotational latency and seek-time, or even communication bandwidth [1, 2].

The admission control proposed here is categorized as deterministic admission control, as we have to maintain every client that have already been served to have sufficient resources, and do not degrade it even a little with the advent of a new admitted request. However, the use of probability in every class will change the rejection rate (blocking probability) compared to of normal admission control.

In normal admission control, every class (say, popular class and less-popular class) is considered has its own maximum number of ports to service clients. Suppose the popular class has now been serving maximum number of clients (all ports are

occupied). Then, when there is another client requesting for a popular movie, simply it will be rejected as there is no available ports given belong to that class. Conversely, with the advent of probability in admission mechanism, ports for every class can be shared to handle requests from different classes with a predefined probability assigned to each class. Again, recall that popular class' ports are fully occupied. Then, the system will seek the other available class' port, which is here less-popular class. If less-popular class still has ports available, then we apply a probability mechanism here. We simply “flip a coin”, to allow or to reject the request if we consider that the probability for acceptance or rejection of less-popular class to different class request is of 0.5. The amount of ports shared on every class is directly connected with the probability applied within one class to accept client requests from the other classes.

The organization of the paper is the following: background of the importance of admission control is introduced. This also deals with classification of admission control and techniques available to date to be used in admission control scheme. A method is then proposed for a deterministic admission control to work in a more flexible manner for admitting requests from clients to reduce the chance of underutilization. Some experiments are conducted with the use of NS-2 simulator and results are given, followed by conclusions.

II. RELATED WORK

Online access to continuous media has now been increasing in demand. With this, resources, such as disks bandwidth, memories, processing speed, and communication bandwidth must be calculated well to provide a guaranteed level of service. Likewise, Video on Demand is now become commonplace. Users or clients request videos and providers must provide the service with high quality of service to maintain customer's satisfaction. As video system is intolerant of packet loss, usually industry standard gives the value of 10^{-6} packet loss or better as suggested in [3]. For this goal, resource management is paramount to maintain customer's satisfaction. Hence, admission control is the only way of managing the resources which now comes in some ways.

Admission control mechanism can be classified into two categories: measurement-based and parameter-based [2]. In measurement-based admission control, resources of servers or communication channel are continuously measured and the

results are used for admission decision purposes. However, even it is continuously monitored and directly fed back to the system, the length of average service duration will affect the resource utilization in general. Meanwhile, parameter-based admission control provides a hard Quality of Service guarantees which has a specific deterministic bounds on the delay or image quality for video transferred to clients [4]. Parameter-based approach calculates network resources required before accepting a request [5]. If it is sufficient, such a request is then accepted. Conversely, if one or more parameters used are considered as insufficient the request will be rejected.

Parameter-based admission control is further divided into deterministic admission control and best-effort [6]. Best-effort admission control itself can be in the form of statistical admission control [2, 7] and some may call such an admission control as predictive admission control [8]. For deterministic admission control, guaranteed of service for every client that currently being serviced is paramount. No requests will be admitted when resources are not enough to spawn a new service. Deterministic quality is achieved by considering worst-case assumptions for the resources granted to a client for a service. However, this to some extent may lead to an extreme under-utilization [1, 9].

Meanwhile, best-effort admission control tries to eliminate this potentially underutilized resource by sacrificing service quality to currently served clients. This quality gradation, however, may affect user perception [9]. Even worse, estimating resources available cannot lead to accuracy due to the complexity of estimating disk response time in general purpose operating system [10]. Some of techniques used in the statistical methods are as follows: record mean and deviation standard of past usage as suggested in [1], using video information stored offline as in [11], or using exponential weighted average to guess future resource needs as in [8]. More complex methods are also found such as using Markov Chain for current connection as in [12], applying different blocking probability for every class using recursion [13] and using combination of statistical method and caching technique as in [9] and in [12] respectively, to name a few.

III. METHODOLOGY

We use NS-2 simulator to conduct simulation on probability admission control for Video on Demand service. We have two types of video requests λ_1 and λ_2 : popular video requests and less-popular video requests, respectively. This two class scenario has been introduced in [6]. The total capacity of video server is distributed among these two classes, such that popular class has more ports as compared to less-popular class. Both λ_1 and λ_2 are different exponential arrival rates for popular and less-popular classes such that λ_1 has higher rate than λ_2 .

In normal admission control, requests to a specific class will be served only by its predefined class ports. For example, when a new request to popular class arrives, the system will only admit it if there is available ports in such popular class. However, in the proposed method, if a client would like to request a popular movie but all the ports in popular class are fully occupied, then the client still has the chance of getting

the service with the use of other class ports. This chance depends heavily on the probability of requests previously set. For example, if popular class requests have the probability of 0.8, means that when there are no ports available in popular class, such requests will have 80% chance to get ports from other classes. Practically, for admitting the requests using other class ports, the system will generate random number between 0 and 1 for the client and see if it is less or equal to the probability number set to such specific class requests (i.e. 0.8). If so, than the client deserves the service. The mechanism can be extended to accommodate three or more classes system and probability numbers are applied to requests for every class. Requests of popular class have higher probability number as compared to of less-popular class. This means that popular requests are treated superior to less-popular requests. Fig. 1 describes the algorithm used in the experiment.

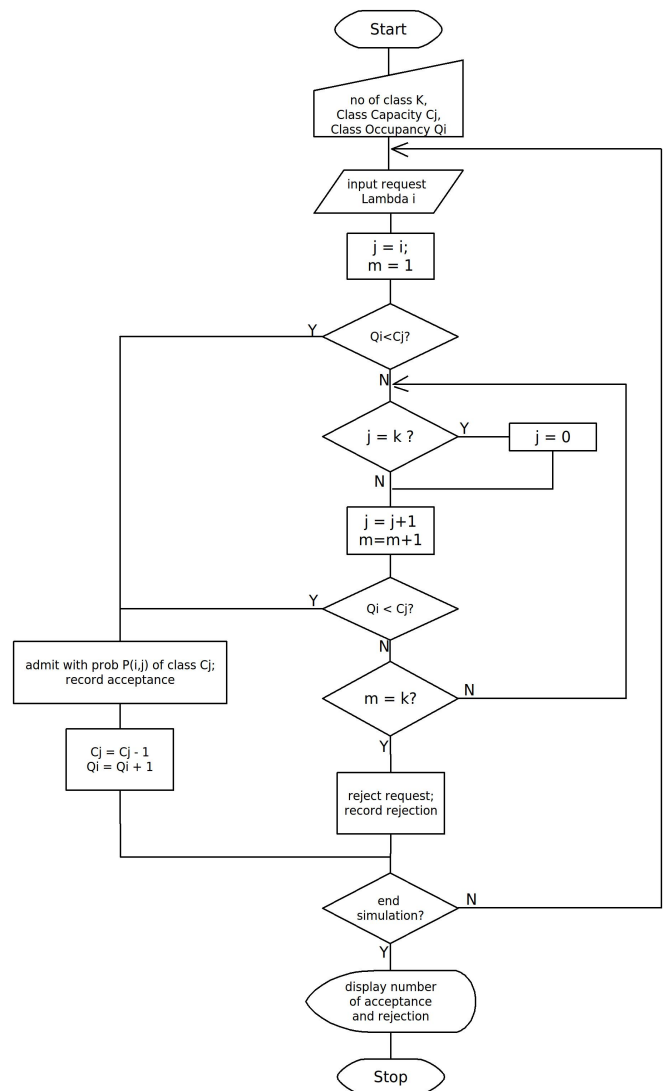


Figure 1. Flowchart of probability admission control

The summary of the algorithm is the following: Suppose that a video server has a total bandwidth capacity of C ports, distributed into k classes of services for video streams (C_j ,

where $j = 1 + k$, each with rate of λ_i where $i = 1$ to k . When a new request arrives with exponential arrival rate of λ_i , the system checks the port occupancy for class j (Q_i). If $Q_i \leq C_j$ then the system admits the request with probability $P_{ij} = P(i, j) = 1.0$. Conversely, when the ports of such a class are fully occupied, then the system tries to look at its next available class for available ports as long as $Q_{i+1} < C_{i+1}$ with probability $P_{ij} = P(i, j)$ of less than 1.0. After admitting a request, the occupancy of the associated class will be set to $Q_{i+1} = Q_{i+1} + 1$. If the request cannot be admitted by every k class, then the server will discard the request.

With such an algorithm, it is expected that the system will reduce blockage and will increase traffic by simply putting more flexibility on separated classes with the use of probability. In addition, during simulation, the arrival rate ratio of popular and less-popular classes is, in general, to be set proportionally to the capacity or ports assigned to such classes.

IV. RESULTS

Some scenarios have been investigated to see the effect of probability variations applied to the classes. Variations on probability number have been applied along with variations on arrival rate, class capacity, and on average video duration. The results of the experiment are as follow.

A. Variations on Arrival Rate

Firstly, the arrival rate of popular class is varied from 1 to 8 requests per minute, while keeping the other class arrival rate to 3, and set popular and less-popular capacity to 500 and 250 and average video duration to 90 minutes with its deviation standard of 15 minutes. The simulation is run for the period of 180 minutes.

The probabilities of popular and less-popular classes are set to 0.9 and 0.1 respectively. Fig. 2 shows that when the system is highly populated with clients' requests, popular class has smaller blockage when the algorithm is applied. Some amount of ports is taken from less-popular class, dedicated for accepting requests on popular movies. Hence, in a busy server, at popular arrival rate of 6 per minute and above, less-popular class has higher blocking probability.

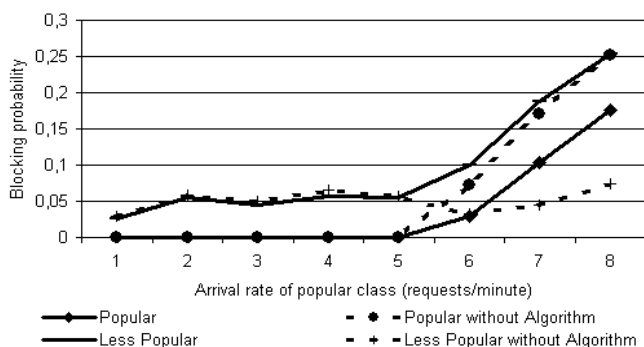


Figure 2. Blocking probability to variations on popular arrival rate class with probability P1 and P2 of {0.9, 0.1}

However, in underutilized area, say when the arrival rate is less than 5, the effect of the algorithm is not significant. Simulations with probability of {0.7, 0.3} and of {0.5, 0.5}, are shown in Fig. 3 and Fig. 4. The use of the algorithm generally reduces the blockage (see Fig. 5). This result is obtained when the probability is set to {0.5, 0.5}, under which the simulation has its maximum effect. A reduction of 2% blockage is achieved in average, from 3% to 1% in underutilized situation.



Figure 3. Blocking probability to variations on popular arrival rate partition with probability P1 and P2 of {0.7, 0.3}

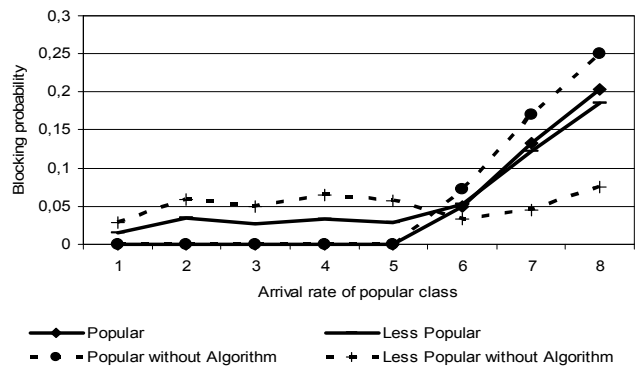


Figure 4. Blocking probability to variations on popular arrival rate partition with probability P1 and P2 of {0.5, 0.5}

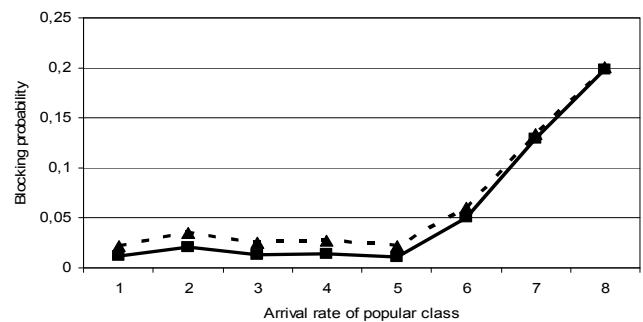


Figure 5. Average blocking probability for popular and less-popular class with and without algorithm with probability P1 and P2 of {0.5, 0.5}

Secondly, the experiments are conducted for the less-popular class is variably set to the arrival rate of 1 to 8 requests per minute while keeping the popular class arrival rate to 8 requests per minute. It is shown that in underutilized situation, the algorithm gives less blocking to popular class. However, in populated conditions, popular class goes higher in blockage. Meanwhile, less-popular class is in reversed conditions. It is found that changing the probability of popular and less-popular classes from $\{0.9, 0.1\}$ to $\{0.5, 0.5\}$ will merely slide the graph's twist points somewhere as shown in Fig. 6 and Fig. 7.

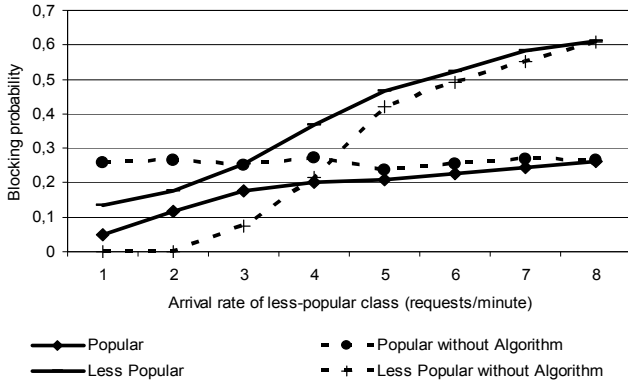


Figure 6. Blocking probability to variations on less-popular arrival rate partition with probability P1 and P2 of $\{0.9, 0.1\}$

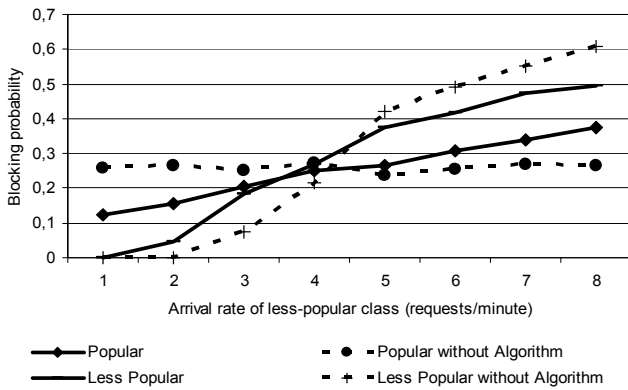


Figure 7. Blocking probability to variations on less-popular arrival rate partition with probability P1 and P2 of $\{0.5, 0.5\}$

B. Variations on Class Capacity

The capacity of both classes are changed but with similar total number of ports. During experiment, arrival rates are held constant to 8 and 3 requests per minute, while popular class port capacity is changed from 375 to 550 while keeping the total number of ports to 750. The average video duration is set to 90 minutes. It is shown in Fig. 8 that blocking probability for popular class is decreasing along with the addition of ports. Conversely, with reduction on number of ports to less-popular class, blocking probability of the associated class is increasing.

The algorithm itself has significant impact on reducing the blockage of popular class when the load (requests) is set to

normal (of 8 and of 3 requests per minute). Conversely, the algorithm increases the less-popular class blockage. Changing in probability composition has less meaning in this class capacity variation as shown in Fig. 9 and Fig. 10. In average, blockage is maintained constant which is of 20% for popular and less-popular classes as shown in Fig. 11.

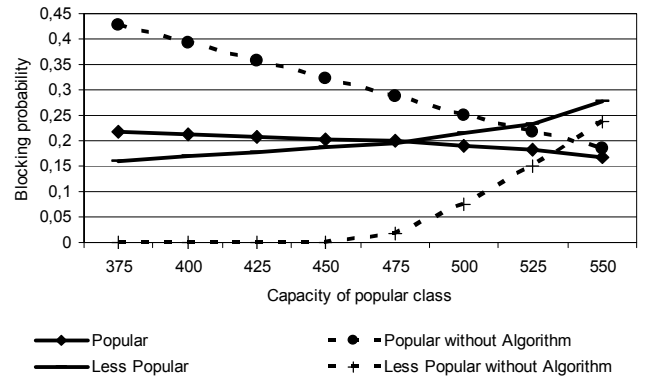


Figure 8. Blocking probability to variations of popular class port with probability P1 and P2 of $\{0.7, 0.3\}$

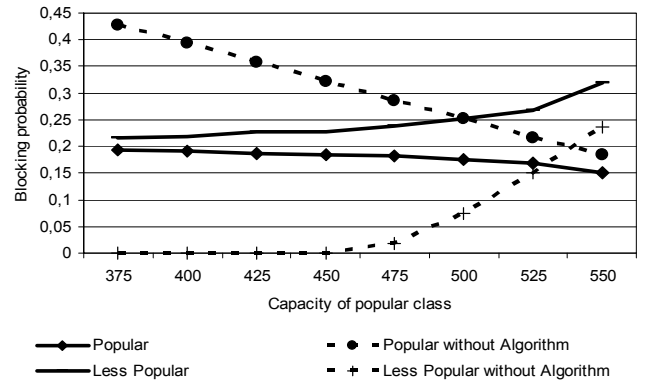


Figure 9. Blocking probability to variations of popular class port with probability P1 and P2 of $\{0.9, 0.1\}$

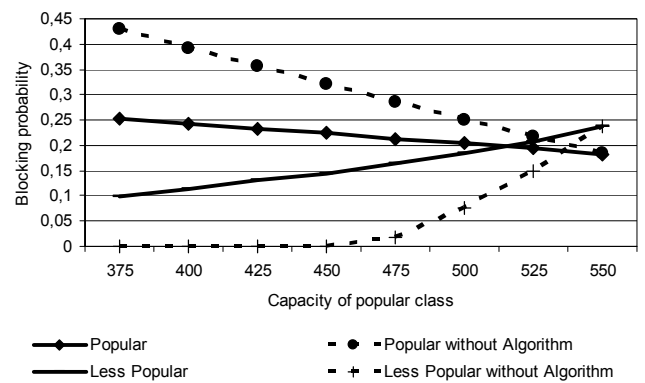


Figure 10. Blocking probability to variations of popular class port with probability P1 and P2 of $\{0.5, 0.5\}$

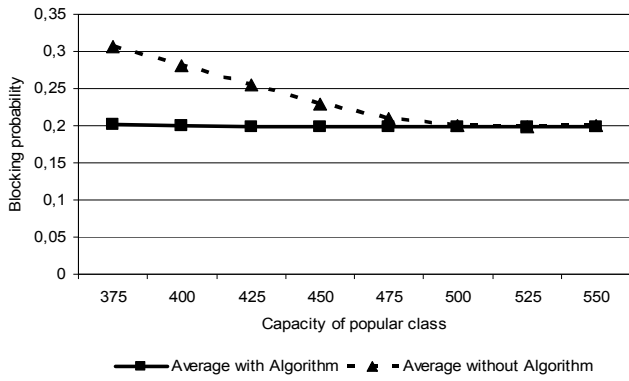


Figure 11. Average blocking probability to variations of popular class capacity with probability P1 and P2 of {0.7, 0.3}

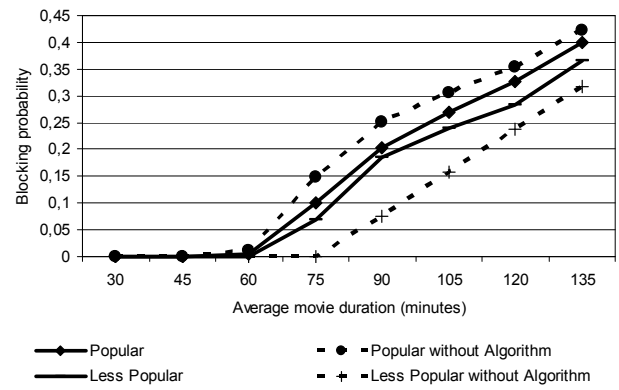


Figure 14. Blocking probability to variations on average video duration with probability P1 and P2 of {0.5, 0.5}

C. Variations on Average Duration of Movies

In this experiment, arrival rates and capacities are held constant. Movie duration is then varied from 30 to 135 minutes for its average. It is shown in Fig. 12, the algorithm has reduced the blocking probability of popular class and has increased less-popular class in blocking. Fig. 13 and Fig. 14 show that class which has less probability will suffer much. However, in average, there will be no significant benefit to use the algorithm (see Fig. 15).

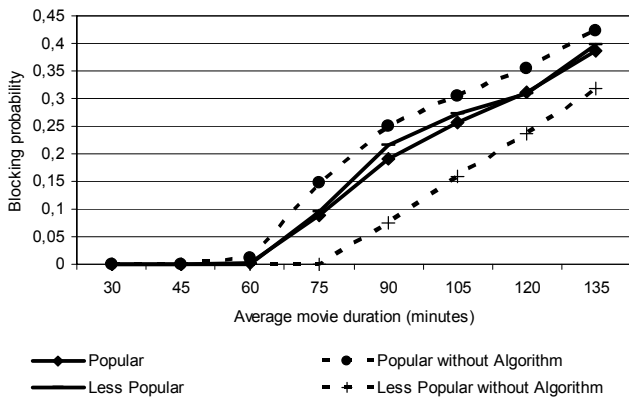


Figure 12. Blocking probability to variations on average video duration with probability P1 and P2 of {0.7, 0.3}

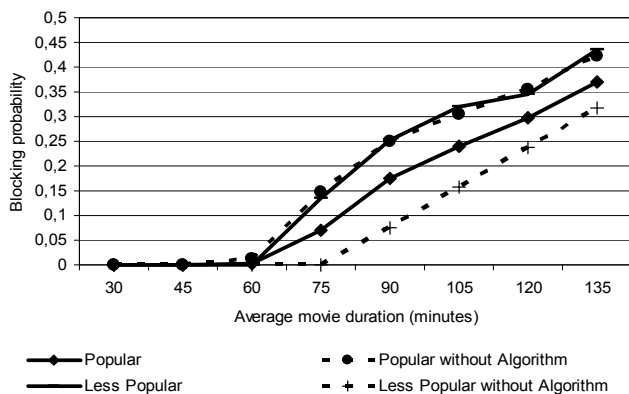


Figure 13. Blocking probability to variations on average video duration with probability P1 and P2 of {0.9, 0.1}

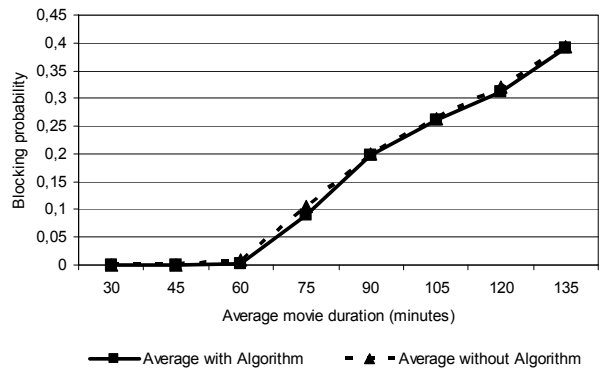


Figure 15. Average blocking probability to variations of movie duration with probability P1 and P2 of {0.7, 0.3}

The interaction between classes also works for systems having more than two classes. Consider that class 1 is popular and the rest of the classes are less popular. In our simulation, such an interaction of classes creates less blockage within underutilized area in average and the algorithm will have less power to do when the system is nearly or fully populated (arrival rate of 6 and above). However, class 1, which can be popular class of our interest, will suffer at underutilized area when the algorithm is used. This is because, during simulation, other less popular classes have run out their ports first and start taking ports from class 1 which still have some. The higher the probability of less-popular classes, the higher blockage that popular class will suffer under the underutilized area.

One way of reducing this issue is by changing the algorithm a little, in order that classes with have less popularity cannot take ports from more popular classes. Only requests from more popular classes deserve the chance of having ports from other classes when its own class ports are fully occupied. Fig. 16 and Fig. 17 present the result of such modification in the algorithm.

Nevertheless, with such a modification, the significant difference of having algorithm applied and without algorithm within underutilized area has no longer existed. These results

also conform to [13] that in clients which have more priority must have more chances to be served to maximize system's profits.

during a certain period of service in order for the system optimally manages its resources. Adaptive change on probability algorithm is considered important for future works.

ACKNOWLEDGMENT

The authors would like to thank to Mohammad Ammad, for sharing ideas and conduct proof reading of this material.

REFERENCES

- [1] H. Vin, P. Goyal, A. Goyal, and A. Goyal, "A Statistical Admission Control Algorithm for Multimedia Servers". Proceedings of the 2nd ACM International Conference on Multimedia. San Fransisco: ACM, 1994, 33-40
- [2] R. Zimmermann and K. Fu, "Comprehensive Statistical Admission Control for Streaming Media Servers", Proceedings of the 11th ACM International Conference on Multimedia. Berkeley, CA: ACM, 2003, 75-85.
- [3] Cisco Systems, Integrated video admission control for the delivery of a quality video experience - White Paper, 2006, Retrieved from http://www.cisco.com/en/US/prod/collateral/routers/ps368/prod_white_paper0900aecd804a05bd.html
- [4] M. Fidler and V. Sander, "A Parameter-Based Admission Control for Differentiated Services Networks". Computer Networks, Vol. 44, No. (4), 2004, 463-479. doi:10.1016/j.comnet.2003.12.
- [5] S. Jamin, S.J. Shenker, and P.B. Danzig, "Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service", Proceedings of the 16th Conference of the IEEE Computer and Communications Societies. Kobe: IEEE Computer Society Press, 1997, 973-980. Doi:10.1109/INFCOM.1997.631035
- [6] I-R. Chen and C-M. Chen, "Threshold-Based Admission Control Policies for Multimedia Servers". The Computer Journal, Vol. 39 No. (9), 1996. 757-766. doi:10.1093/comjnl/39.9.757
- [7] E. Biersack and F. Thiesse, "Statistical Admission Control in Video Servers with Variable Bit Rate Streams and Constant Time Length Retrieval", in Proceedings of the 22nd EUROMICRO Conference. Prague: IEEE Computer Society, 1996, pp. 633-639.
- [8] B. Qazzaz, J. Moreno, J. Xiao, P. Hernandez, R. Suppi, and E. Luque, "Admission Control Policies for Video on Demand Brokers". Proceedings of International Conference on Multimedia and Expo. Baltimore, MD: IEEE Computer Society, 2003, 529-532. doi:10.1109/ICME.2003.1221670
- [9] J.B. Kwon and H.Y. Yeom, "A Statistical Admission Control Scheme for Continuous Media Servers using Caching". Multimedia Tools and Applications, Vol. 19, No. (3), 2003, 279-296. doi:10.1023/A:1023229414510
- [10] I-H. Kim, J-W. Kim, S-W. Lee, and K-D. Chung, "Measurement-based Adaptive Statistical Admission Control Scheme for Video-on-Demand Servers". Proceedings of International Conference on Information Networking. Beppu City, Oita: IEEE Computer Society, 2001, 471-478. Doi:10.119/ICOIN.2001.905467
- [11] F.Y-S. Lin, "Optimal Real-time Admission Control Algorithms for the Video-On-Demand (VOD) Service". IEEE Transactions on Broadcasting, Vol. 44 No. (4), 1998, 402-408. doi:10.1109/11.735901
- [12] S. Kang and H.Y. Yeom, "Statistical Admission Control for Soft Real-time VOD Servers". Proceedings of the 2000 ACM Symposium on Applied Computing. Como, Italy: ACM, 2000, 579-584. doi:10.1145/338407.338504
- [13] Y. Zhi, Z. Zhu, X. Ma, and B. Wang, "Client-class based admission control for distributed Video-on-Demand systems". International Conference on Wireless, Mobile and Multimedia Networks. Hangzhou, 2006, 1 - 4. doi:10.1049/cp:20061519

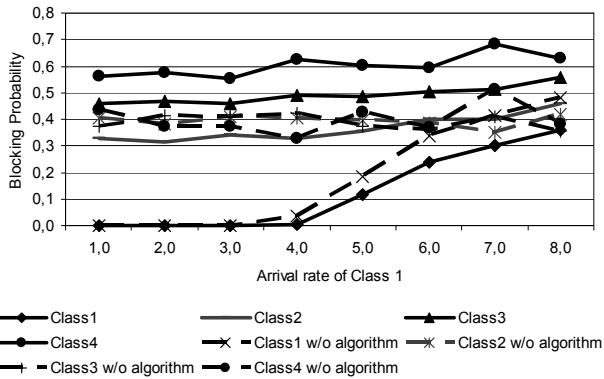


Figure 16. Blocking probability to arrival rate of class 1 with probability of {0.9, 0.7, 0.5, 0.1} after modification

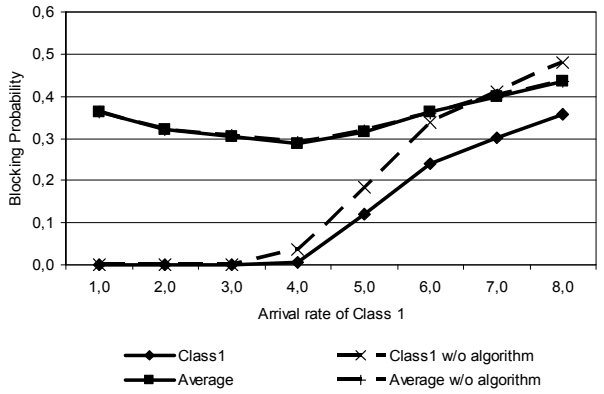


Figure 17. Average blocking probability to arrival rate of class 1 with probability of {0.9, 0.7, 0.5, 0.1} after modification

V. CONCLUSIONS

In this paper, the performance of Video on Demand system employing probability algorithm has been tested for its variation on request arrival rate, capacity, as well as movie duration, so we can distinguish the performance as compared to the system without such an algorithm applied. Through simulation, it is shown that the algorithm works very well on popular class for decreasing the blockings when the system is nearly or fully-utilized. Hence, the overall system performance is also increasing, in particular when the system has been set to have a fifty-fifty chance for every class available. We have shown that this algorithm helps very much on utilizing system resources better when the system goes nearly or fully-utilized. Furthermore, this algorithm can be used for making adjustment when there is a change in any of the system parameters such as request arrival rate and class capacity