Journal of Engineering and Applied Sciences 13 (22): 9771-9776, 2018
ISSN: 1816-949X

Evaluate of Queue Management RED and DT Mechanisms in Wireless Cloud Computing Environment

[1]Mohammed Iqbal Dohan, [1]Alaa Abid Muslam Abid Ali,
[2]Salam Allawi Hussein and [3]Atheer Hadi Issa Alrammahi
[1]Department of Multimedia,
[2]Department of Computer Science,
[3]Department of Computer Information System,
College of Computer Science and Information Technology,
University of Al-Qadisiyah, Al-Diwaniyah, Iraq

Abstract: Now a days, cloud computing has rapid development over the internet, especially in areas that involves high demands on large scale of network computing such as IT. Sharing of resources is the main aim of cloud computing where processing service and network bandwidth needed to access it should be allocated concurrently. This becomes a challenging situation in avoiding congestion which affect the reliability of the cloud computing and in turn, the usability of the application service. Current solutions concerns reducing the size of required resource for congested resource type, rather than restricting all service requests as in the existing networks. Even though those solutions achieve an efficient use of resources in under some congestion conditions but they do not enable the fair allocation of the use of the resources. Furthermore, the effect of congestion collapse in wireless cloud computing environment. This study present a sufficient comparative evaluate of their performance over the commonly used congestion control mechanisms, namely Random Early Detection (RED) and Drop-Tail (DT) queue management mechanism.

Key words: Queue management, DT, RED, wireless cloud, mechanisms, service

INTRODUCTION

Wireless cloud computing environment: A wireless cloud computing environment (Fig. 1) the bandwidth width of the transmitter and buffers works on the principle of network resources and the time it takes to process from the network resources competing network resources for a range of resources, storage, bandwidth and processing time. Cloud computing management is facing threats in the management system that lead to the determination of capacity resulting in an impact on the output leading to an unsatisfactory level (Kadhim *et al*., 2017). Avoiding degradation of data throughput in a cloud computing environment that is overloaded with a specific uptake difficult to avoid in the network (Mahdi *et al*., 2018). The wireless cloud computing environment cannot afford to accept all offered data traffic without some regulations flow management regulates the traffic received from the outside to the cloud computing environment (Kadhim *et al*., 2018a-c).

When the load is greater than the estimated in the wireless cloud computing environment causes congestion problem (Kadhim *et al*., 2018a-c). Frequent redundant buffering causes a major network problem delay in data transmission and loss of packets leads to inefficient cloud computing (Kadhim *et al*., 2018a-c). The congestion of interconnecting networks causes the collapse of the environment. Cloud computing by packet transmission across the network leads to the dropping of the barrier before reaching the destination because of congestion (Kaur *et al*., 2018). When a collapse due to overtaking the high load results in a decrease in congestion as shown in Fig. 2.

For example, the dropped packets are moved back to the source. The process of downloading the network continues continuously sending the packets with the flow of other networks until the data is delivered to the source machine (Wang, 2008). It produces a breakdown in the environment due to congestion as previously indicated. It is possible to avoid overcrowding if the source is detected and the reduction of the modified transient load reduces the congestion (Wang *et al*., 2012). The congestion in the wireless cloud computing environment originates from the router to the source

Corresponding Author: Mohammed Iqbal Dohan, Department of Multimedia, College of Computer Science and Information Technology, University of Al-Qadisiyah, Al-Diwaniyah, Iraq

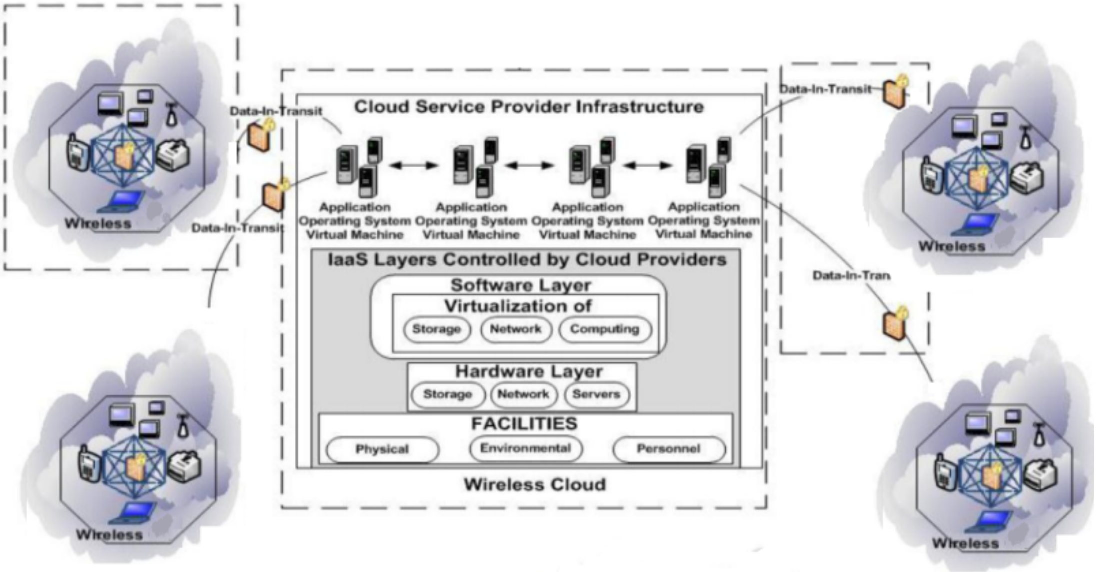

Fig. 1: Wireless cloud computing environment (Ahmed *et al*., 2018)

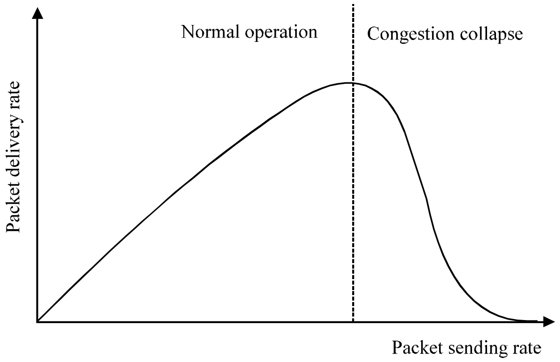

Fig. 2: The effect of congestion collapse in wireless cloud computing environment

(Nzanywayingoma and Yang, 2017). Notice of congestion is present at the source dropping the steering device packed with beams. When there is detected congestion in the network packets the source submits three duplicate resolutions after a time limit to re-transmit again (Roy *et al*., 2017). Signal control when a traffic jam occurs, the steering device sends information to the device in the control of congestion. Congestion overlay causes congestion packets more than usual in a wireless cloud computing environment resulting in inefficient network if signals are controlled in the administration in the network to reduce congestion (Kanagavelu *et al*., 2014). On the other hand, the source can distinguish congestion and manage the problems of devices from signal control. Reducing congestion in computing reduces user downtime, loss of performance and improved cloud performance, providing the user with appropriate performance when minimizing congestion (Yong *et al*., 2017). In order to maintain the work of the cloud environment according to its capacity and potential, it is necessary to understand how congestion occursthe development of high-performance wireless cloud computing and provide extensive service at an efficient time in data processing (Lee and Zomaya, 2012). The highly distributed and collaborative nature of the wireless cloud enables managing the data traffic between the wireless environment and the cloud environment, (i.e., data transported from the cloud to the wireless end-sources and vice versa). Therefore, a pro-active queue management is needed for the wireless cloud computing environment. Internet devices in the cloud computing environment are routed by default, one of the most common Random Early Detection (RED) and Internet Engineering Task Force (IETF) more recommended are (RED) (Liu and Dimitrov, 2018). Control and resolution uses (RED) the average length in the queue that has been in a period of time in the list (Jiang *et al*., 2013). That (RED) uses the length of the list of the average mean interaction occurs with the congestion slowly. There is a significant difference between waiting queues and the efficiency of congestion detection leads to deterioration and delay of queues and loss of beams.

Queue management mechanisms in cloud networks: Mechanisms used for queue size management and congestion control in the networks of cloud data centers include re-active and pro-active mechanisms. As mentioned earlier, Drop-Tail (DT) (Iwatani and Yakoh, 2010). Is the common conventional queue management mechanism in cloud networks which is considered as re-active mechanism as it does not take any action unless the buffer is flooded where all arriving packets will be dropped. On the other hand, pro-active mechanisms such as Random Early Detection (RED). Control the queue size by dropping packets protectively before the buffer gets full. Reactive queue management mechanisms suffer from two main problems (Ryu *et al.*, 2004).

Lock-out which appears when some TCP flows occupy the buffer space due to high transmission rate, resulting in high packet loss of other TCP flows and unfairness.

Full queues where the queue size increases to the maximum buffer capacity as the reactive queue management mechanism does not operate any procedure to prevent the queue from growing and lasting for long time periods, resulting in high queuing delays.

Conversely, proactive queue management mechanisms were designed to prevent (or minimize) the network congestion (Panjanathan and Ramachandran, 2017). This necessitates early congestion detection and notification to the TCP senders, so that, they can help in congestion avoidance or alleviation. Proactive mechanisms were proposed for enhancing the overall throughput by reducing the packet loss, maintaining a small queue size to decrease the end-to-end and queuing delay and fair sharing the link bandwidth to avoid lock-out problem (Adams, 2013). These mechanisms offer preventive procedures to manage the queue size to overcome the problems associated with reactive ones. Thus, they perform a precautionary random packet drops prior to the queue fullness and that the probability of that packet drops is done according to the levels the network congestion. The dropped packet can imply implicit notification to the TCP senders of the occurrence of the congestion (MerLinlagnes *et al.*, 2013). In return, TCP senders decrease their cwnd to moderate the congestion. Hence, if the TCP sender perceives the congestion early enough, it could take a proper action to alleviate and control the congestion. This can help in reducing global synchronization that could take place when many TCP senders reduce their cwnd simultaneously and thus, improving the throughput and link utilization (Tahiliani *et al.*, 2015). In the following subsections, tail-drop and Random Early Detection (RED) are discussed in terms of their advantages and disadvantages.

Drop-Tail (DT): TD drops packets from the end of the queue whenever the buffer is full as shown in Fig. 3. Arriving packets are allowed to the buffer only if there is a room to accommodate them. The size of the queue is the hyphen from dropping all packets is reduced because of the packet's departure from the queue head. TD has less computational overhead and implementation complexity compared to other queue management mechanisms (Ding *et al.*, 2016).

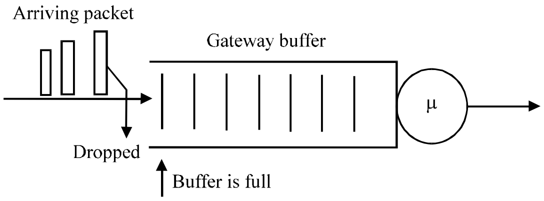

Fig. 3: Tail-drop mechanism

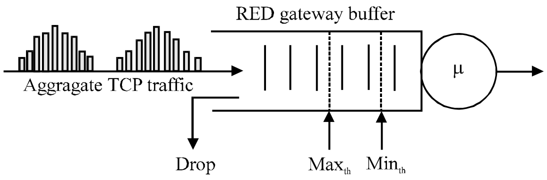

Fig. 4: RED mechanism

However, TD realizes the network congestion after it isalreadyhappened, causing the buffer to overflow. Thus, TD is unable to control the congestion as it does not detect the congestion timely. As TD drops packets only when the queue itis full notificationthe TCP senders not effective In order to avoid congestion. The time the congestion is inferred and having TCP senders decreasing their CWND, many packets are dropped due to buffer overflow. Therefore, TD suffers from high packet loss ratio, unfairness as it does not ensure the packet-dropping rate is proportional to flow isbandwidth share, global synchronization, low link utilization in addition to large queuing delay and large end-to-end delay.

Random Early Detection (RED): Internet Engineering Task Force (IETF) has recommended RED to be used for the Internet routers (Grigorik, 2013). As shown in Fig. 4, RED uses two thresholds (Min_{th}, Max_{th}) to differentiate between different levels of network congestion (Ryu *et al.*, 2004).

With RED, all arriving packets are allowed to the buffer until the queue size reaches the threshold value. It then drops some of arriving packets randomly with a linear distribution function until the queue size reaches the Math threshold value where all arriving packets are dropped.

RED uses average queue size to make its congestion control decisions which help in better distinguishing between temporary and persistent increases in the queue size and to correctly detect the network congestion (Uma and Gurumurthy, 2014). Even though that RED solves global synchronization problem by dropping packets at uniformly spaced interval, the usage of the average queue size derives RED to react to congestion slowly as it is collected over long time. This would cause high fluctuation in the queue size in addition to unseasonable the detection of early congestion and notification of the source of congestion reduces the loss of packets and delays in the queue in the network.

RESULTS AND DISCUSSION

Simulation and evaluate: Present a sufficient comparative evaluate of their performance over the commonly used congestion control mechanisms, namely Random Early Detection (RED) and drop-tail queue management mechanism. As per simulation, at mentioned that evaluate the TCP-Westwood and Data Center TCP (DCTCP) versions of TCP that are widely used in wireless cloud computing. So far, I have evaluated Data Center TCP (DCTCP) to investigate the TCP incast and TCP out cast problems. I have studied the performance in terms of actual (current) queue length and average packet arrival rate and I am going to investigate the performance further in terms of packet loss, link utilization, throughput and bandwidth unfairness as I did in the previous experiments for TCP-Vegas and TCP-Sack. Same as with previous experiments, I first studied the behavior of DCTCP with short-lived connections along with RED and with Drop-tail, respectively. The performance evaluation result of DCTCP over RED in terms of actual queue length is presented in Fig. 5 while for DCTCP over drop-tail, the result is shown in Fig. 6.

From Fig. 5, it is clear that when TCP incast occurs where the queue length reaches 60000 bytes, RED has already started dropping packets with probability proportional to the queue length, allowing the queue length to be moderated and maintained at approximately 18000 byte occupancy level, preventing serious packet loss and timeouts at the sender side. This means that DCTCP can provide less latency and good traffic burst tolerance for the short-lived connection which could contribute to a high throughput.

On the other hand, the queue occupancy level for DCTCP over drop-tail is usually full at 80000 byte queue length. This clearly worsens TCP incast problem where many data packets are dropped as the queue is often maintained full. The expected consequences of such case are the high delay and much reduction in the throughput. This is confirmed through the average arrival rates for DCTCP over RED and drop-tail which are presented in Fig. 7 and 8, respectively. It is clear that, due to the notification given by RED, the packet arrival rate is maintained around 7000 bytes as the senders are aware of the network condition and thus, keep sending manageable amount of data traffic to help in controlling congestion.

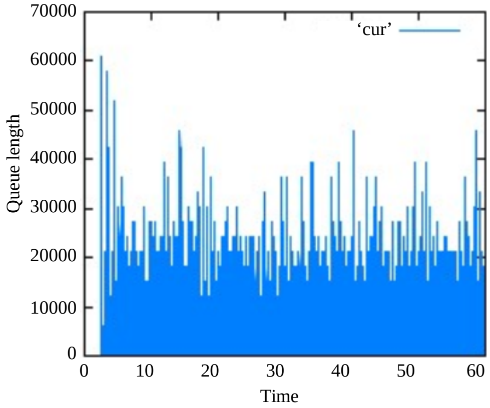

Fig. 5: Actual queue length for DCTCP over RED

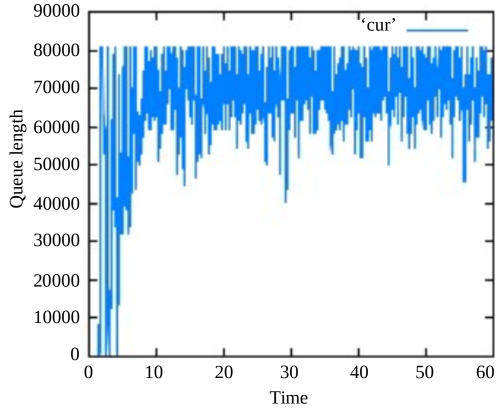

Fig. 6: Actual queue length for DCTCP over drop-tail

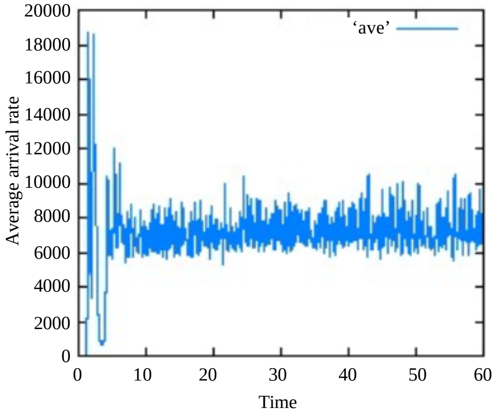

Fig. 7: Average arrival rates for DCTCP over RED

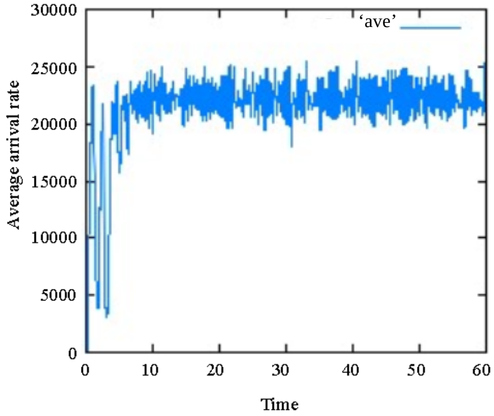

Fig. 8: Average arrival rates for DCTCP over drop-tail

However, DCTCP over RED still suffers from performance reduction and unreliable in meeting the quality of services required by used as RED suffers from slow congestion detection and notification which in turn affect the resources utilization.

The average arrival rates for DCTCP over drop-tail presented in Fig. 8 shows that DCTCP senders keep sending their data and flooding the buffers as they depend on the timeouts only because drop-tail does not provide any notification. It is clear that the packet arrival rate fluctuates gently around 23000 bytes of the queue occupancy level. This can cause high packet loss (as the packets are dropped successively) increase the overall end-to-end delay and reduce the throughput significantly.

CONCLUSION

The simulation and evaluate presented a brief discussion on two of the most well-known queue management and congestion control mechanisms in the cloud networks, namely Tail-Drop (TD) and Random Early Detection (RED) and how they contribute to the performance reduction of TCP in cloud computing environment.

It was concluded that neither TCP variants nor the current queue management and congestion control mechanisms provide a convincing solution to the aforementioned performance issues of TCP. Thus, there is a high necessity for further research to improve the performance of cloud data center networks which focuses on developing a dynamic queue management and congestion control mechanism that interacts with an efficient version of ECN-capable TCP. The recommendations to study the DCTCP performance further in terms of packet loss, link utilization, throughput and bandwidth unfairness. Then evaluate the TCP-Westwood to have clear idea on which one will be used with our proposed queue management mechanism based on their performance and behavior.

REFERENCES

Adams, R., 2013. Active queue management: A survey. IEEE. Commun. Surv. Tutorials, 15: 1425-1476.

Ahmed, S.T., C. Science and B. Education, 2018. Innovative queue management mechanism for congestion control in wireless cloud computing environment. J. Adv. Res. Dyn. Control Syst., 10: 1208-1214.

Ding, D., C. Wang, L. Ding, N. Wang and D. Liang, 2016. Hopf bifurcation control in a FAST TCP and RED model via multiple control schemes. J. Control Sci. Eng., 2016: 1-10.

Grigorik, I., 2013. High-Performance Browser Networking. O Reilly Publishing, Sebastopol, California, USA., ISBN:978-1-449-34476-4, Pages: 379.

Iwatani, J. and T. Yakoh, 2010. A detection method of active queue management in communication paths. Proceedings of the 2010 IEEE International Conference on Industrial Technology (ICIT), March 14-17, 2010, IEEE, Vina del Mar, Chile, ISBN:978-1-4244-5695-6, pp: 1009-1013.

Jiang, X., J. Yang, G. Jin and W. Wei, 2013. RED-FT: A scalable Random early detection scheme with flow trust against DoS attacks. IEEE. Commun. Let., 17: 1032-1035.

Kadhim, Q.K., H.S. Mahdi and H.A. Ail, 2018b. Storage architecture for network security in cloud computing. Diyala J. Pure Sci., 14: 1-17.

Kadhim, Q.K., R. Yusof and S.R. Selamat, 2018c. The cloud computing control in the government services. J. Adv. Res. Dyn. Control Syst., 10: 1136-1147.

Kadhim, Q.K., R. Yusof, H.S. Mahdi and S.R. Selamat, 2017. The effectiveness of random early detection in data center transmission control protocol based cloud computing networks. Intl. J. Commun. Antenna Propag, 7: 1-7.

Kadhim, Q.K., R. Yusof, H.S. Mahdi, S.S.A. Al-Shami and S.R. Selamat, 2018a. A review study on cloud computing issues. J. Phys. Conf. Ser., 1018: 1-10.

Kanagavelu, R., B.S. Lee, N.T.D. Le, L.N. Mingjie and K.M.M. Aung, 2014. Virtual machine placement with two-path traffic routing for reduced congestion in data center networks. Comput. Commun., 53: 1-12.

Kaur, K., S. Garg, G.S. Aujla, N. Kumar and J.J.P.C. Rodrigues *et al*., 2018. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. IEEE. Commun. Mag., 56: 44-51.

Lee, Y.C. and A.Y. Zomaya, 2012. Energy efficient utilization of resources in cloud computing systems. J. Supercomputing, 60: 268-280.

Liu, J. and S. Dimitrov, 2018. Open shortest path first routing under random early detection. Networks, 71: 120-135.

Mahdi, H.S., K. Qusay and I.Y. Khudhair, 2018. Empirical investigation of TCP incast congestion in wireless cloud computing networks. J. Comput. Sci., 14: 663-672.

MerlinLagnes, C., J. Benimol and M.D.J.W. Wise, 2013. Congestion control in wireless networks using AQM algorithms. Intl. J. Adv. Res. Electron. Commun. Eng., 2: 239-246.

Nzanywayingoma, F. and Y. Yang, 2017. Efficient resource management techniques in cloud computing environment: A review and discussion. Intl. J. Comput. Appl., 15: 1-18.

Panjanathan, R. and G. Ramachandran, 2017. Enhanced low latency queuing algorithm with active queue management for multimedia applications in wireless networks. Intl. J. High Perform. Comput. Networking, 10: 23-33.

Roy, S., R. Bose and D. Sarddar, 2017. SSeS: A Self-Configuring Selective Single-Path routing Protocol for Cloud-based Mobile Wireless Sensor Network. In: Proceedings of the 1st International Conference on Intelligent Computing and Communication, Mandal, J., S. Satapathy, M. Sanyal and V. Bhateja (Eds.). Springer, Singapore, Asia, ISBN:978-981-10-2034-6, pp: 655-667.

Ryu, S., C. Rump and C. Qiao, 2004. Advances in Active Queue Management (AQM) based TCP congestion control. Telecommun. Syst., 25: 317-351.

Tahiliani, R.P., M.P. Tahiliani and K.C. Sekaran, 2015. TCP Congestion Control in Data Center Networks. In: Handbook on Data Centers, Khan, S.U. and A.Y. Zomaya (Eds.). Springer, Berlin, Germany, ISBN:978-1-4939-2091-4, pp: 485-505.

Uma, S.V. and K.S. Gurumurthy, 2014. Congestion Control by Heuristics in High-Speed Networks using ANN. In: Emerging Research in Electronics, Computer Science and Technology, Sridhar V., H.S. Sheshardi and M.C. Pandma (Eds.). Springer, Berlin, Germany, ISBN:978-81-322-1157-0, pp: 323-333.

Wang, C., W.H. Hu and N. Bagherzadeh, 2012. A load-balanced congestion-aware wireless network-on-chip design for multi-core platforms. Microprocess. Microsys., 36: 555-570.

Wang, C.C., 2008. Pruning network coding traffic by network coding-a new max-flow algorithm. Proceedings of the IEEE International Symposium on Information Theory, July 6-11, 2008, IEEE, Toronto, Canada, ISBN:978-1-4244-2256-2, pp: 26-30.

Yong, B., G. Zhang, H. Chen and Q. Zhou, 2017. Intelligent monitor system based on cloud and convolutional neural networks. J. Supercomputing, 73: 3260-3276.