Journal of Engineering and Applied Sciences 13 (19): 8063-8074, 2018
ISSN: 1816-949X

Innovative Bursty Traffic Control Technique to Overcome Outcast Problem in Cloud Computing Environment

Atheer A. Kadhim
Department of Computer Science, Al Yarmouk University College (YUC),
32001 Baqubah, Diyala, Iraq

Abstract: The promising services offered by cloud computing environments have led to huge amount of data that need to be processed and stored. Wireless cloud networks rely on TCP/IP protocols for reliable transfer of data traffic between the cloud end-users and servers and vise-versa. Even though, TCP has been successful for several applications it, however, does not perform well in wireless cloud environments. The many-to-one communication pattern used in such environments with such huge amount of data resulted in TCP incast problem. TCP outcast problem happens when traffic that consists of connections of many incoming flows and few flows at two ports of a switch/router compete for one common output port, the throughput of the few flows is affected implicitly. It occurs in the cloud environment where routers and switches of the data centres operate Tail-Drop (TD) queue management mechanism as it has less implementation complexity and less computational overhead compared to other queue management mechanisms. However, the use of TD leads to frequent and multiple successive packet drops from the queue of a specific port which contributes to frequent timeouts and severe global synchronization occasions. TCP outcast problem can be mitigated by minimizing the queue occupancy level at the routers and switches of the cloud networks. This study presents an innovative technique for controlling TCP bursty traffic to overcome outcast problem in cloud computing environment. The performance evaluation of the proposed technique shows that it improves the performance of the cloud networks in terms of fairness represented by less delay and queue length and comparable data throughput achieved.

Key words: Tail-Drop (TD), frequent timeouts, cloud networks, TCP bursty traffic, protocols, Iraq

INTRODUCTION

Overview of cloud computing environment: The rapid growth of the internet and the emergence of client server computing and have led to the idea of cloud computing environment which becomes an important part of future internet. Data centres are considered the heart of the cloud environment and the backbone of the internet hosting various ranges of applications.

Data centres are basically categorized into two categories (Zhang *et al*., 2013a-c), online services to users and online resources to users. Data centres of the first category provide timely accessible services ranging from social networking to web search to the subscribed users such as Google, Facebook and Yahoo while in the second category, online resources ranging from web hosting to advertisements are provided to the users such as Amazon Elastic Compute Cloud (EC2) and Microsoft Azure to name a few. Data centres are usually referred as mission critical systems where reliability is the main factor that should be achieved at any cost.

In regards to the network infrastructure where 15% of the cost goes in the current wireless cloud data centres networks (Greenberg *et al*., 2008), there are four types of networking based on the purpose of the network:

Client server network: Allows remote internet users to connect to the cloud data centres via wired or wireless ethernet technologies. Once the requests from Internet users are received, they will be distributed to the servers within the cloud data centres according to a specialized load balancer.

Server-server network: Allows high-speed connectivity and communication between the servers of the cloud data centres using ethernet technologies. The servers can collaboratively work to process the internet user's requests.

Server-storage network: Allows high-speed connectivity and communication between the servers and storage devices within the cloud data centres using fiber link or ethernet technologies.

Data centre management network: Allows managing the data centre network for sustainable availability of the services provided. Such network usually uses Ethernet technologies with different cabling.

In regard to the network topology, two-tier network architecture is common in cloud now a days where access switches/routers depend on switching and routing functionalities provided by aggregation switches/routers. This architecture has many advantages the design becomes simpler as only few switches/routers can be sufficient and thus, the network latency and aggregate power consumption can be reduced. Nevertheless, this architecture presents limited scalability. Once the input and ports on a particular aggregation switch/router are fully used, then adding another aggregation switch/router would present a high complexity. A mesh connection between a pairs of aggregation switches/routers should be established with high bandwidth, thus, avoiding having bottlenecks in the cloud network. But these aggregation switches/routers perform routing protocols, this means that the more switches/routers connected in the network the more routing interfaces and information are provided, hence, mesh design would increase the complexity inside the cloud networks.

In regard to data traffic and applications requirements, the nature of traffic in cloud data centres imposes some restrictions. As the cloud environment is dedicated to achieve high bandwidth and low latency, Round-Trip Time (RTT) in some cases is <250 microseconds without considering the queuing delay (Hammid *et al*., 2017a, b; Vasudevan *et al*., 2009). There are three types of traffic coexist in cloud data centre networks:

Short traffic (mice or short-lived connections) such as Google search and Facebook updates in which transmission rate of the connections is <100 kB. Such connections require short response times. Medium traffic (cat or medium-lived connections) such as small-and medium-sized file downloads using YouTube and also Facebook photos. The transmission rate of such connections varies from 100 kB to 5 MB and they usually require low latency.

Long traffic (elephant or long-lived connections) such as large software updates (antivirus updates) and video on demand (large movie downloads). The traffic of these connections usually exceeds 5 MB and they require high throughput.

In regard to the performance of the cloud data centre networks, there are three essential requirements: high bursty traffic tolerance, low delay and high throughput (Adams, 2013; Hammid and Sulaiman, 2017). These performance requirements are imposed by the aforementioned traffic types that coexist in cloud networks where each traffic type requires certain application quality of service that differs from the other. The main issue related to the performance of cloud networking is the use of Transmission Control Protocol/Internet Protocol (TCP/IP) stack that was specifically, designed for internet. TCP/IP suffers from several performance issues when it is traditionally and genuinely used in cloud data centre networks without modification according to the diverse requirements of several coexist traffic types of the cloud environment.

As these cloud data centre networks deal with and process huge amount of data at any given time, it is important to have an efficient and reliable cloud networking environments, considering performance, scalability, resilience and sustainability. Furthermore, since, the demand on cloud computing is increasing, it is essential to meet the requirements of the scaling cloud users by designing fast, reliable and efficient networks for communication within cloud data centres to manage the increasing load of the cloud user's traffic. While the reliability of the cloud networks relies on their transport layer, it is worthy to understand the performance related issues in cloud data centres networks which are presented in the next section.

Literature review

Transmission Control Protocol (TCP) issues in cloud networks: Transmission Control Protocol (TCP) is the most common transport layer protocol that has been widely used by internet applications (Alizadeh *et al*., 2010; Forouzan, 2002; Morandin, 2010) and dominates the majority of the traffic in cloud data centre networks (Prakash *et al*., 2012).

Due to its congestion control mechanisms, TCP has recognized to be scalable, robust and capable of dealing with different network conditions. Nonetheless, it has been realized that TCP fails to fulfil the aforementioned three essential requirements because of the characteristics diversity of the cloud data centre networks and the variety of requirements of their applications. Such diversity and requirements imposed many challenges to TCP congestion control mechanisms and make TCP suffers from several deficiencies (Prakash *et al*., 2012) such as TCP incast (Vasudevan *et al*., 2009), TCP outcast (Prakash *et al*., 2012), queue build-up (Alizadeh *et al*., 2010), buffer pressure (Alizadeh *et al*., 2010) and pseudo-congestion effect (Cheng *et al*., 2013).

Several enhancements of traditional TCP have been developed for cloud computing environment to alleviate aforementioned problems. Table 1 presents a summary of these enhancements and whether they can solve or

Table 1: Summary for TCP enhancements for cloud computing environment

TCP enhancement	Researchers and years	Evaluation method	Modification	TCP incast mitigation	TCP outcast mitigation	Buffer pressure mitigation	Queue build-up mitigation	Pseudo-congestion mitigation
FG-RTO	(Vasudevan *et al*., 2009)	Testbed and NS-2	Sender	Yes	No	No	No	No
FG-RTO-delayed ACK disabled	(Vasudevan *et al*., 2009)	Testbed and NS-2	Sender	Yes	No	No	No	No
DCTCP	(Alizadeh *et al*., 2010)	Testbed and NS-2	Sender/ receiver/ router/switch	Yes	No	Yes	Yes	No
ICTCP	(Wu *et al*., 2013)	Testbed	Receiver	Yes	No	No	No	No
IA-TCP	(Hwang *et al*., 2012)	NS-2	Receiver	Yes	No	No	No	No
D^{2}TCP	(Vamanan *et al*., 2012)	Testbed and NS-3	Sender/receiver/ router/switch	Yes	No	Yes	Yes	No
CP-FITDC	(Zhang *et al*., 2013a-c)	Modelling and NS-2	Sender/receiver/ router/switch	Yes	No	Yes	Yes	No
TDCTCP	(Das and Sivalingam, 2013)	OMNeT++	Sender/receiver/ router/switch	Yes	No	Yes	Yes	No
TCP-GIP	(Zhang *et al*., 2013a-c)	Testbed and NS-2	Receiver	Yes	No	No	No	No
PVTCP	(Cheng *et al*., 2013)	Testbed	Sender/Receiver	Yes	No	No	No	Yes

mitigate the problems of TCP incast, TCP outcast, buffer pressure, queue build-up and pseudo-congestion effect or not.

Common queue management mechanism in cloud networks: Tail-Drop (TD) 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. Reactive queue management mechanisms suffer from two main problems (Hammid, 2016; 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.

Therefore, a proactive queue management mechanism is needed 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; Hammid *et al*., 2016). 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 (Hammid, 2013; 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 (Hammid *et al*., 2017a, b; Lee *et al*., 2014).

In the following subsections, TD is discussed in terms of its advantages and disadvantages. TD drops packets from the end of the queue whenever the buffer is full as shown in Fig. 1. Arriving packets are allowed to the buffer only if there is a room to accommodate them. All arriving packets are dropped until the queue size 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 (Hammid *et al*., 2013; Wang *et al*., 2015) .

However, TD realizes the network congestion after it is already happened 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 is full, this late notification to the TCP senders is not effective for congestion avoidance. By the time the congestion is inferred and having TCP senders decreasing their Congestion Window (CWnd), a lot of 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's bandwidth share, global synchronization, low link utilization in addition to large queuing delay and large end-to-end delay.

Therefore, it is important to have a mechanism that solves global synchronization problem by dropping packets at uniformly spaced interval and does not cause

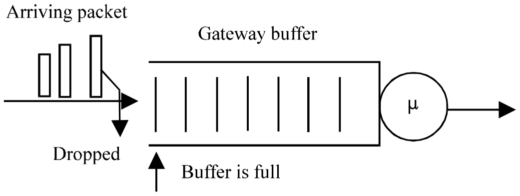

Fig. 1: Tail-Drop mechanism (Hassan and Jain, 2004)

high fluctuation in the queue size in addition to unseasonable untimely congestion detection and notification which reduce the network performance due to high packet loss and large queuing delays.

MATERIALS AND METHODS

TCP outcast problem: TCP outcast phenomenon is that when traffic that consists of connections of many incoming flows and few flows at two ports of a switch (or router) and compete for one common output port, the throughput of the few flows is affected implicitly (Hammid, 2016). It occurs in the cloud environment where routers and switches of the data centres operate Tail-Drop (TD) queue management mechanism as it has less implementation complexity and less computational overhead compared to other queue management mechanisms. However, TD does not guarantee the proportional packet drop ratio of each flow to the flow's share of the bandwidth. TD drops packets from the end of the queue when the packet arrival reaches the maximum queue length. TD leads to frequent and multiple successive packet drops from the queue of a specific port which contributes to frequent timeouts and severe global synchronization occasions. This case is known as port blackout which causes a noteworthy reduction in the performance of few flows (in terms of response times) due to high delay resulted from frequent timeouts. Figure 2 illustrates the port blackout situation where data packets are arriving at input ports (A and B) and competing for one output port (C). As shown in Fig. 2, data packets that have arrived at port B are buffered successfully while the ones arrived at port A are being dropped successively.

As throughput of a TCP connection is proportional to its RTT in inverse manner, this could lead to RTT bias. This means that TCP connections that have low RTT can attain higher share of bandwidth than other connections. Nonetheless, it has been perceived low RTT TCP connections are overtaken by high RTT TCP connections as TCP outcast drives TCP to present inverse RTT bias (Hammid, 2016). Since, many-to-one communication model is the groundwork of several applications in the cloud and that networks of the data centres utilize TD queue management mechanism in their routers and switches, TCP outcast is also the common exist problem the cloud computing environment which could not be solved by the TCP enhancements as shown in Table 1.

TCP outcast problem can be mitigated by using an effective congestion control mechanism that cooperates with TCP end-hosts that could minimize the queue occupancy level at the routers and switches of the data centres networks.

Proposed technique: The aim of the proposed technique is directing the actual queue length to a manageable level over a period of Time T which is used for computing the drop Probability P of a packet (pkt_i). As bursty data traffic is expected in this environment, thus, the rate of incoming packets is assumed to change during T. The decision on whether to drop the packet belong to certain connection or allowed to the queue buffer depends on output of the drop probability function which in turns relies on the average rate of arriving packets of that connection and actual queue length are which compared to a fixed threshold (which equals to half of the buffer size B).

For every arranging packet, compute the standard deviation of the sample means of the packet arrival rate and the queue length. If drop probability output equals or less than zero, then it implies that expected increase in the queue size over the next T period is below the available space of the buffer, hence, the packet will be allowed to the queue. In contrast, if the output greater than zero, it means that the packet will be dropped with probability proportional to the number of packets arriving during T.

Upon the arrival of a new packet, the average arrival rate (R_{avg}) and its acceleration (data burst) based on standard deviation of it sample means with respect to the packet interarrival time. Then, the drop probability is determined as:

$$P(pkt_i) = \frac{(((\partial^*T)+R_{avg})-\gamma)^*T)-(Q_{thr}-Q}{\left((\partial^*T)+R_{avg}\right)^*T}$$

Where:

∂ = The data burst which is the acceleration of average arrival rate with respect to the packet interarrival time
T = The Time period during which the drop probability is computed
R_{avg} = The average rate of the incoming traffic
γ = The rate of the transmission link (packets/sec)
Q_{thr} = The Queue length threshold (B/2)
Q = The actual Queue length at any given time

That is to say, it defines the fraction of the decrease in the queue size resulted of random drop to the increase resulted of packet arrivals over the period of time T that should be randomly dropped to enforce the fairness between connections. Figure 3 presents the flowchart of the process of the proposed technique.

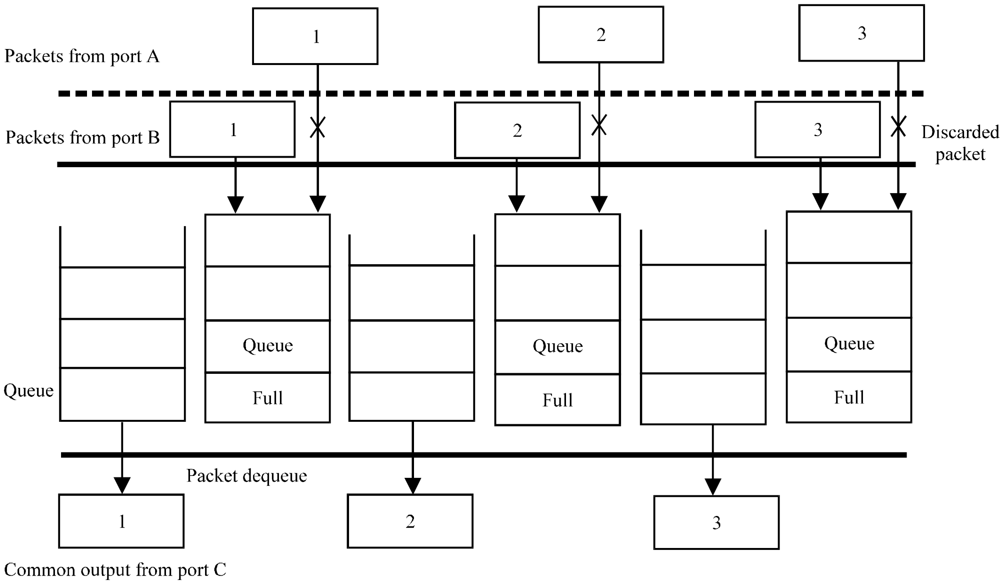

Fig. 2: Port blackout case (Hammid, 2016)

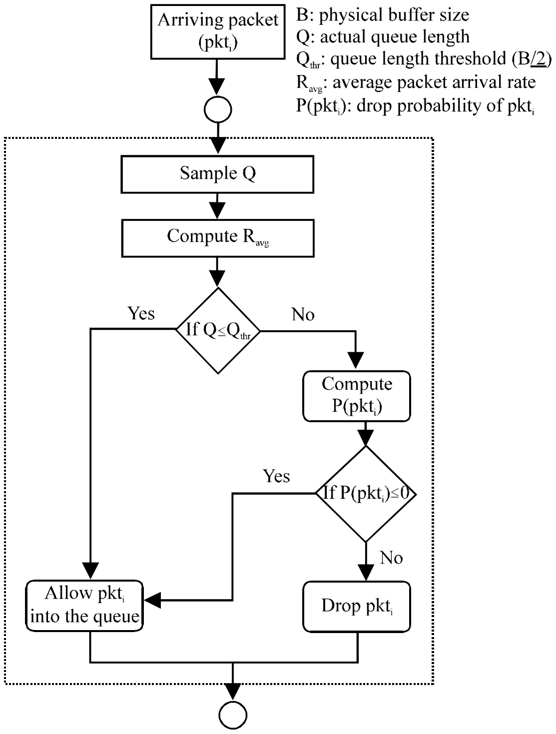

Fig. 3: Flowchart of the proposed technique

Performance evaluation

Simulation experimental scenario: Aiming at investigating the performance of the proposed technique in cloud computing networks, we have simulated short-lived TCP connections (mice) where thousands of senders (clients) that effectively and simultaneously sending data for processing at data center in a cloud with several servers connected to a gateway (router) that implement TD mechanism. The purpose of the scenario is to explore and study the TCP outcast problem where 500 users concurrently use web search (as one of the large-scale applications in the cloud) through a router that is connected to the data centres over 100 Mbps link with delay of 25 μsec and how the proposed technique alleviates the problem. This scenario presents cases such as Google search and Facebook updates where users need short response times. The average size of the transferred request file is 10 kbytes and the packet size are 552 bytes. DCTCP version of TCP has been used in this evaluation. The arrival of new TCP sessions flows Poisson process. The arrival time between new sessions is exponentially distributed and selected to be 40 msec which means 25 sessions would be arrive at each node every 1 sec. Such scenario can be found in a university campus where students, staff, auto-machines in labs send (or request) data to (or from) data centres in the cloud. The Retransmission Timeout (RTO) is set to 200 msec. The router buffer size is 100 packets. The simulation time is 120 sec. The evaluation experiments were conducted using network simulator 2 on CentOS Linux operating system. Based on the numerical results gained from experiments, the performance of the proposed technique is evaluated in terms of congestion window pattern, queue size, delay and throughput which all reflect the fairness between the connections when utilizing the proposed technique and TD mechanism.

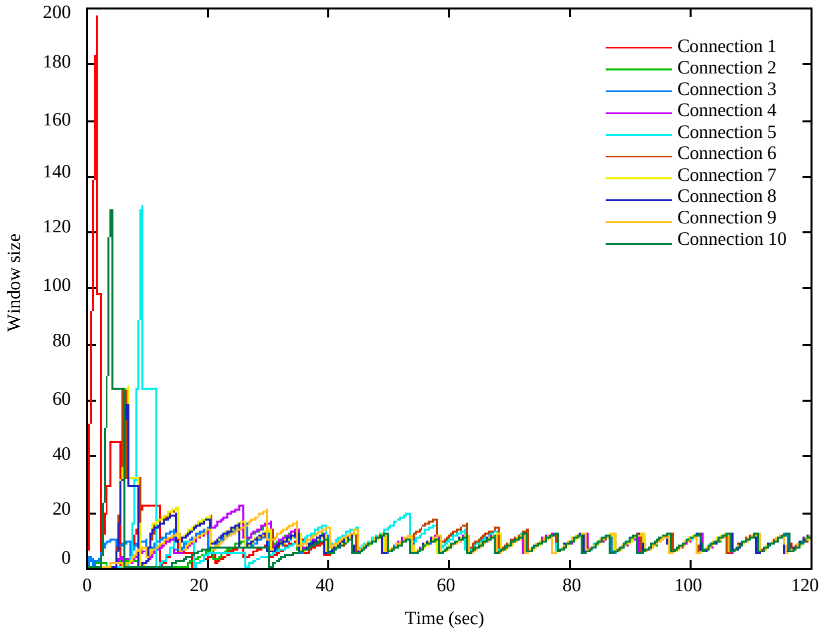

Fig. 4: Behavior of data traffic of TD

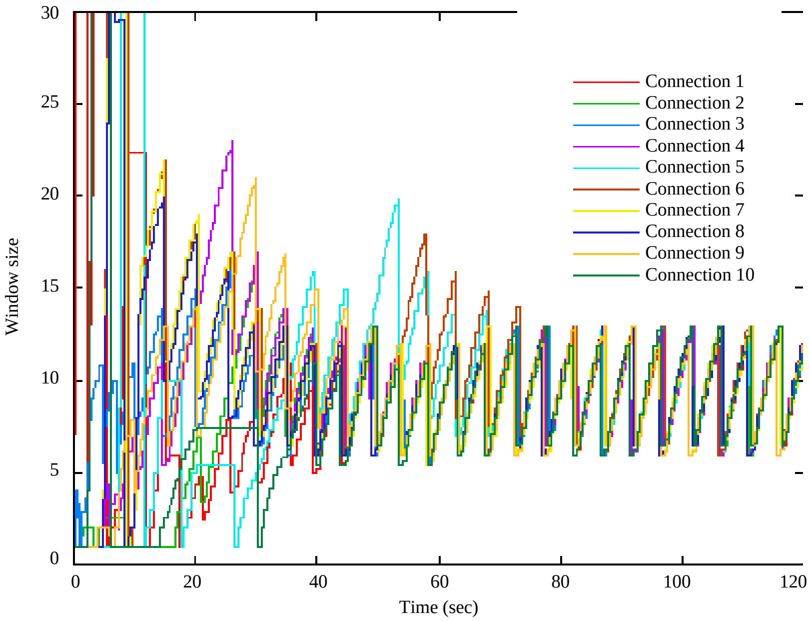

Fig. 5: Behavior of data traffic of TD (zoomed for best visualization)

RESULTS AND DISCUSSION

Congestion window synchronization: For more clarity of the congestion window patterns, Fig. 4 shows the congestion window of 10 random sources which represents the behaviour of corresponding data traffic with the use of TD. It is obvious that synchronization level for the window sizes is high. This implies that data packets are being lost simultaneously, resulting in high fluctuations in the queue length that are comparable to the observed congestion windows. Figure 5 shows the behaviour of data traffic with the use of TD mechanism where the results were zoomed out for best visualization.

In contrast when the proposed technique is utilized, the results suggest that no synchronization is exhibited

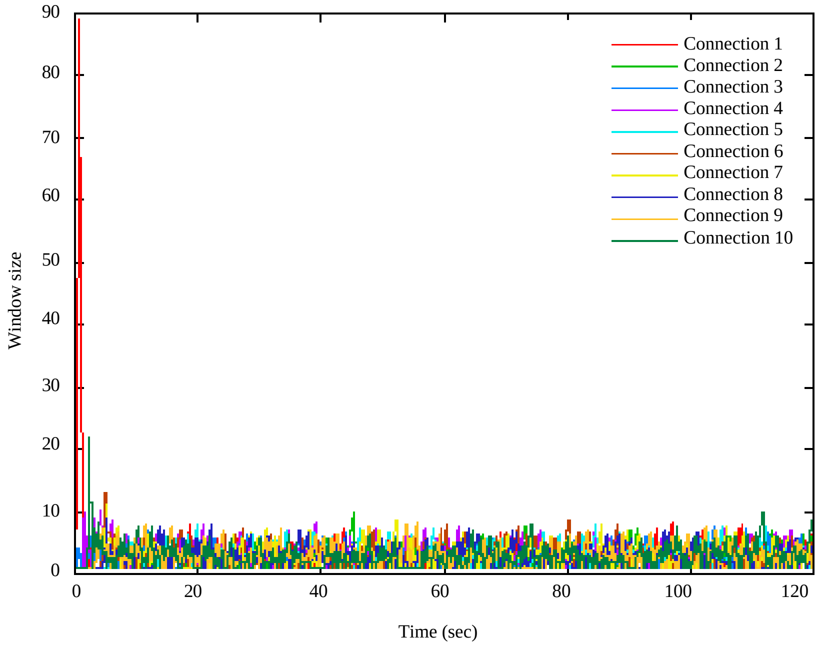

Fig. 6: Behavior of data traffic of the proposed technique

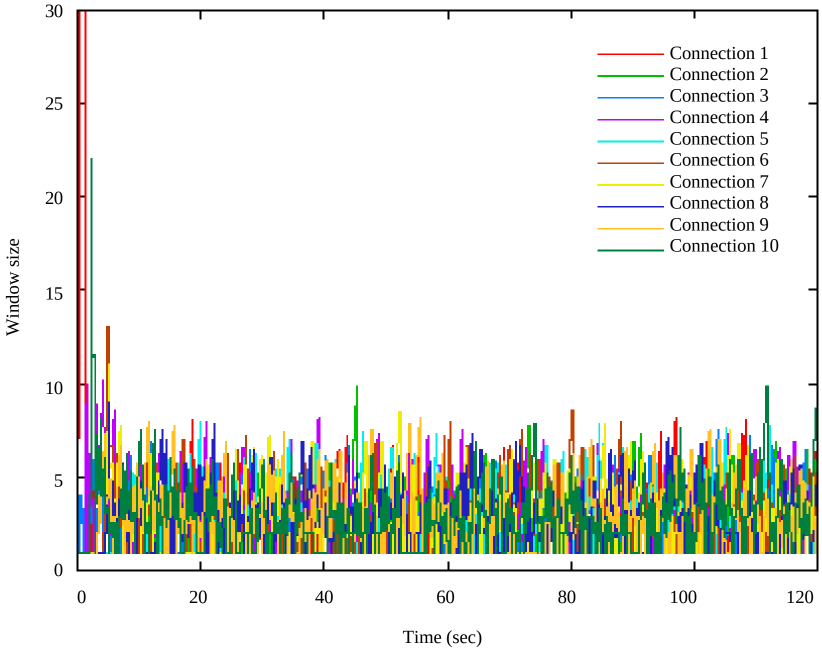

Fig. 7: Behavior of data traffic of the proposed technique (zoomed for better visualization)

between congestion windows. This is confirmed through the graphs presented in Fig. 6 where the congestion windows of 10 random sources are shown. Figure 7 depicts the data traffic behaviour when the proposed technique is used where the results have been zoomed out for better visualization. Compared to pattern of the congestion windows presented with TD that have large oscillations a smaller variation was observed in window sizes when employing the proposed technique. For further clarity on the comparable results, the corresponding congestion windows of the first three connections for both of the proposed technique and TD mechanism were

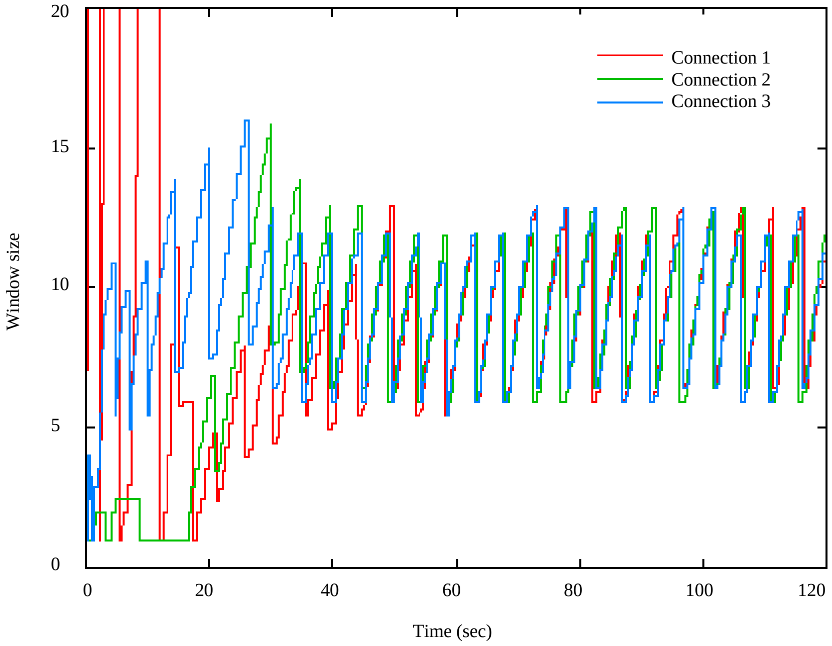

Fig. 8: Behavior of data traffic of first three connections of TD

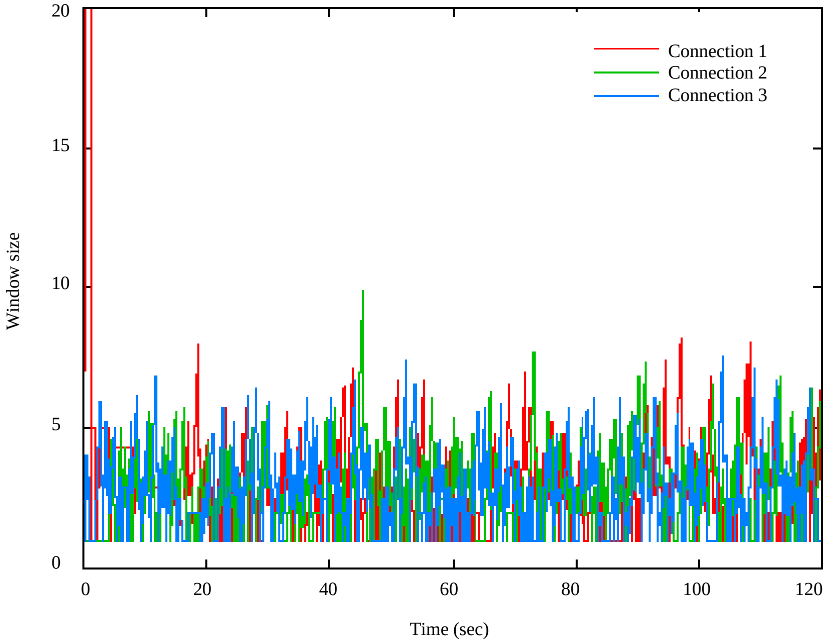

Fig. 9: Behavior of data traffic of first three connections of the proposed technique

plotted in Fig. 8 and 9, respectively. Regarding the synchronization and variation in sizes of the congestion windows, the evaluation results verify that the proposed technique overs much enhancement over TD. It is capable of recovering from several packets drops efficiently and maintaining the lower window fluctuations that are not severe.

Queue length and average delay: This study represents delay observed when using the proposed technique compared to that of TD mechanism. It is inferred accordingly from the observed changes in size of the router queue. Figure 10 presents the actual queue length when the proposed technique is used while Fig. 11 shows

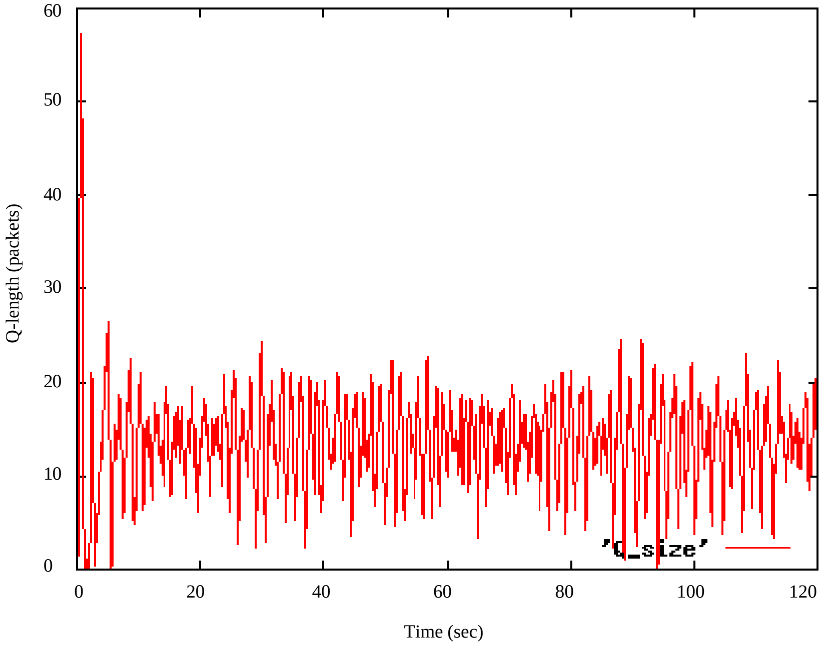

Fig. 10: Actual queue length of the proposed technique

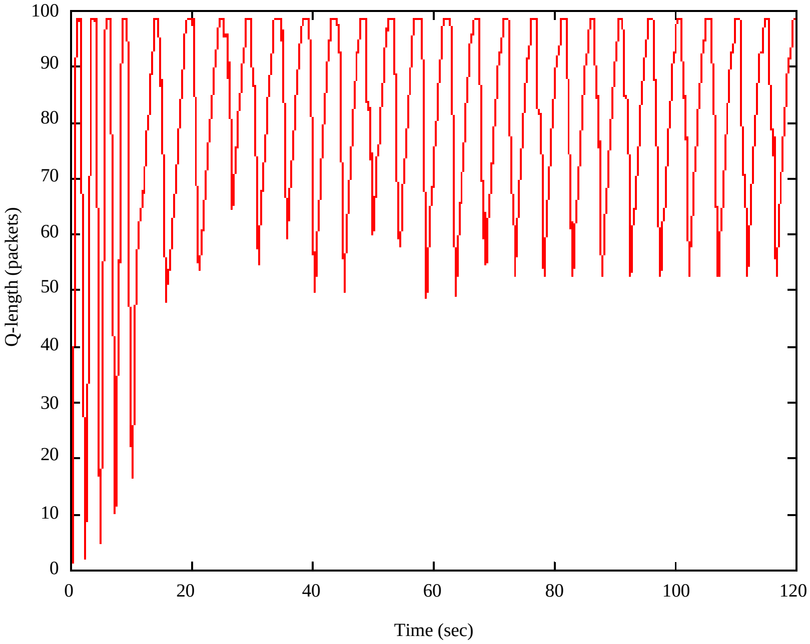

Fig. 11: Actual queue length of TD

actual queue, length introduced by TD mechanism. From the results shown in Fig. 11, it is clear that a small fluctuation in the queue length is presented by the proposed technique compared to that of TD. It presents fairly much lower value in terms of the average queue length in comparison with TD as the average queue length of TD is around 85 packets whereas it is about 15 packets for the proposed technique. Accordingly, the average queuing delay experienced by the connections over the proposed technique is much lower than the average queuing delay contributed by TD. The average queuing delay presented by the proposed technique can be calculated as follows:

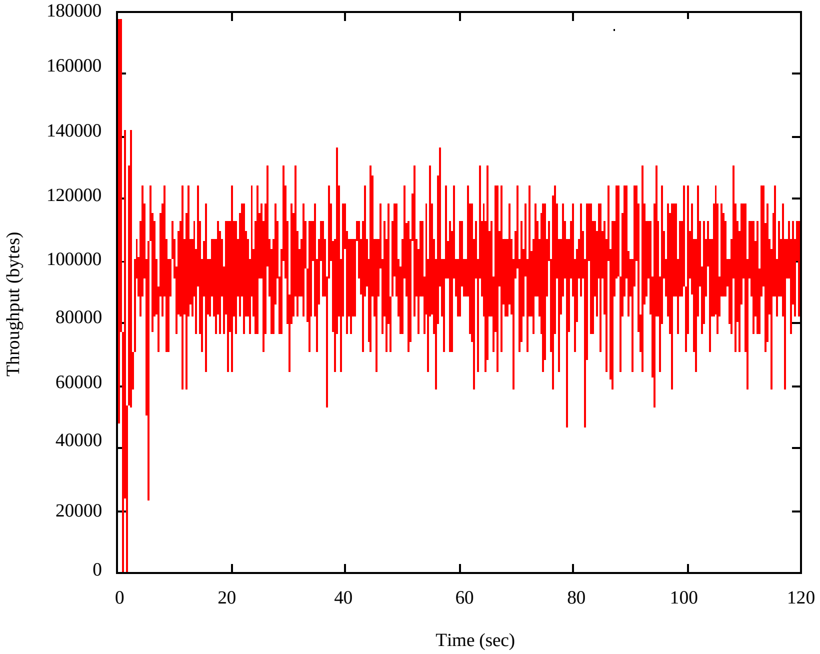

Fig. 12: Throughput of the proposed technique

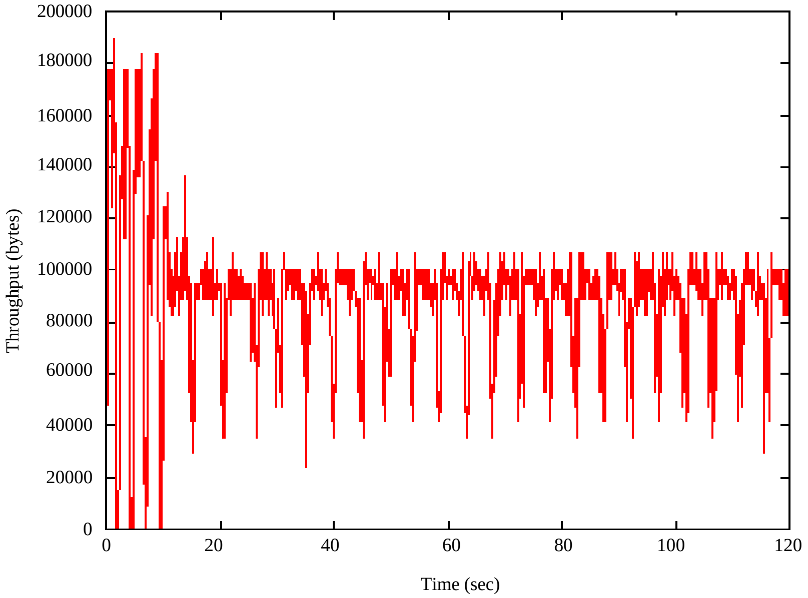

Fig. 13: Throughput of TD

$$\text{Average queuing delay} = \frac{16*592*8}{100*10^6} = 0.75776 \text{ msec}$$

On the other hand, the average queuing delay introduced by TD mechanism is as follows:

$$\text{Average queuing delay} = \frac{85*592*8}{100*10^6} = 4.0256 \text{ msec}$$

While maintain small queue length and presenting lower changeability compared to TD mechanism, the proposed technique allows growth when the connections start transmission which verifies that bursty traffic is not being penalized with the use of the proposed technique.

Throughput: Figure 12 and 13 show the achieved throughput for the proposed technique and TD mechanism, respectively. The results confirm that the proposed technique offers reasonable fairness and achieves better throughput compared to TD mechanism. Also, the proposed technique provides stability in the network presented by its process of packet admission and dropping. The throughput results presented in Fig. 12 reflect the fluctuation in the congestion windows and queue length of the proposed technique.

Conversely, from Fig. 13, it is can be seen that the effect of the global synchronization is very clear on the resulted throughput of TD mechanism which in turns affects the network system utilization inferred by the subsequent identical periods of throughput drops to low level.

CONCLUSION

In this study revealed the importance of cloud computing environment as a part of the contemporary internet and how cloud data centres are classified according to the services presented to the cloud end user. It was shown that, even though there are still many critical issues in regard to networking of data centres in the wireless cloud environment where 15% of the cost is spend on, meeting the requirements of several harmonized traffic types in such environment is a serious challenge. The many-to-one communication model used by most of the cloud applications impose big burden on TCP which is the core transport layer protocol used in cloud data centre. On top of that, TCP could not tolerate highly burst traffic it has failed to offer low delay and high throughput for the cloud applications due to the diversity of the characteristics and requirements of such applications. The traditional TCP congestion control mechanisms make TCP suffers from several performance deficiencies such as TCP incast, TCP outcast, queue build-up, buffer pressure and pseudo-congestion effect in cloud data centre networks. Among these problems, TCP Outcast was the focus in this study where an innovative technique has been developed to control bursty traffic in order to tackle the outcast problem in cloud computing environment that relies on Tail-Drop (TD) mechanism for queue management and congestion control. The proposed technique has been evaluated and compared to with TD mechanism. The results of the performance evaluation showed that the proposed technique offers sensible fairness among connections and outperforms TD in terms of providing lower queuing delay, less queue length oscillation and better data throughput.

RECOMMENDATIONS

For future research, the proposed technique will be modified and examined with the use of Explicit Congestion Notification (ECN) where packets are marked instead of being dropped which can improve the network performance further.

REFERENCES

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

Alizadeh, M., A. Greenberg, D.A. Maltz, J. Padhye and P. Patel *et al*., 2010. Data Center TCP (DCTCP). ACM. SIGCOMM Comput. Commun. Rev., 40: 63-74.

Cheng, L., C.L. Wang and F.C. Lau, 2013. PVTCP: Towards practical and effective congestion control in virtualized datacenters. Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP'13), October 7-10, 2013, IEEE, Goettingen, Germany, ISBN:978-1-4799-1270-4, pp: 1-10.

Das, T. and K.M. Sivalingam, 2013. TCP improvements for data center networks. Proceedings of the 5th International Conference on Communication Systems and Networks (COMSNETS'13), January 7-10, 2013, IEEE, Bangalore, India, ISBN: 978-1-4673-5330-4, pp: 1-10.

Forouzan, B.A., 2002. TCP/IP Protocol Suite. McGraw-Hill Education, New York, USA.,.

Greenberg, A., J. Hamilton, D.A. Maltz and P. Patel, 2008. The cost of a cloud: Research problems in data center networks. ACM. SIGCOMM. Comput. Commun. Rev., 39: 68-73.

Hammid, A.T. and M.H. Sulaiman, 2017. Optimal long-term hydro generation scheduling of Small Hydropower Plant (SHP) using metaheuristic algorithm in Himreen lake dam. Proceedings of the MATEC Web Conference on UTP-UMP Symposium on Energy Systems (SES'17) Vol. 131, October 25, 2017, EDP Sciences, Les Ulis, France, pp: 04017-04021.

Hammid, A.T., 2013. Applications of tuning control actions for the efficient load/frequency control in steam turbine. Intl. J. Curr. Eng. Technol., 3: 1895-1898.

Hammid, A.T., 2016. Direct on line starter motor and reverse system in Allen-Bradley PLC. Diyala J. Pure Sci., 12: 132-148.

Hammid, A.T., A.K. Bhardwaj and S. Prakash, 2013. Design remote power control I/O data acquisition system and control on home automation. Intl. J. Electron. Commun. Comput. Eng., 4: 528-535.

Hammid, A.T., M. Hojabri, M.H. Sulaiman, A.N. Abdalla and A.A. Kadhim, 2016. Load frequency control for hydropower plants using PID controller. J. Telecommun. Electron. Comput. Eng., 8: 47-51.

Hammid, A.T., M.H.B. Sulaiman and A.A. Kadhim, 2017b. Optimum power production of Small Hydropower Plant (SHP) using Firefly Algorithm (FA) in Himreen Lake Dam (HLD), Eastern Iraq. Res. J. Appl. Sci., 12: 455-466.

Hammid, A.T., M.H.B. Sulaiman and A.N. Abdalla, 2017a. Prediction of small hydropower plant power production in Himreen Lake Dam (HLD) using artificial neural network. Alexandria Eng. J., 2017: 1-11.

Hassan, M. and R. Jain, 2004. High Performance TCP/IP Networking: Concepts, Issues and Solutions. Prentice Hall, Upper Saddle River, New Jersey, USA., ISBN:9780131272576, Pages: 383.

Hwang, J., J. Yoo and N. Choi, 2012. IA-TCP: A rate based incast-avoidance algorithm for TCP in data center networks. Proceedings of the 2012 IEEE International Conference on Communications (ICC'12), June 10-15, 2012, IEEE, Ottawa, Ontario, Canada, ISBN:978-1-4577-2052-9, pp: 1292-1296.

Lee, W.H., C.C. Chuang, A.C. Pang, C.C. Shih and C.W. Huang *et al*., 2014. Congestion detection and control in data center networks. Electr. Eng. J., 21: 167-176.

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.

Morandin, G.M., 2010. Transmission Control Protocol (TCP). Google Patents US7656800B2, Cisco Technology, Inc., San Jose, California USA. https://patents. google.com/patent/US7656800B2/en

Prakash, P., A. Dixit, Y.C. Hu and R. Kompella, 2012. The TCP outcast problem: Exposing unfairness in data center networks. Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, April 25-27, 2012, USENIX Association, San Jose, California, pp: 30-30.

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

Vamanan, B., J. Hasan and T.N. Vijaykumar, 2012. Deadline-aware Datacenter TCP (d2tcp). ACM. SIGCOMM. Comput. Commun. Rev., 42: 115-126.

Vasudevan, V., A. Phanishayee, H. Shah, E. Krevat and D.G. Andersen *et al*., 2009. Safe and effective fine-grained TCP retransmissions for datacenter communication. ACM. SIGCOMM. Comput. Commun. Rev., 39: 303-314.

Wang, H., D. Ding, T. Cao and R. Sun, 2015. Time delayed feedback control of internet congestion control model with compound TCP under RED. Proceedings of the 2015 International Conference on Logistics Engineering, Management and Computer Science (LEMCS'15), July 29-31, 2015, Atlantis Press, Shenyang, China, pp: 35-40.

Wu, H., Z. Feng, C. Guo and Y. Zhang, 2013. ICTCP: Incast congestion control for TCP in data-center networks. IEEE/ACM. Trans. Netw., 21: 345-358.

Zhang, J., F. Ren and C. Lin, 2013b. Survey on transport control in data center networks. IEEE. Netw., 27: 22-26.

Zhang, J., F. Ren, L. Tang and C. Lin, 2013a. Taming TCP incast throughput collapse in data center networks. Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP'13), October 7-10, 2013, IEEE, Goettingen, Germany, ISBN:978-1-4799-1270-4, pp: 1-10.

Zhang, J., J. Wen, J. Wang and W. Zhao, 2013c. TCP-FITDC: An adaptive approach to TCP incast avoidance for data center applications. Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC'13), January 28-31, 2013, IEEE, San Diego, California, USA., ISBN:978-1-4673-5287-1, pp: 1048-1052.