

# Empirical Investigation of TCP Incast Congestion in Wireless Cloud Computing Networks

<sup>1</sup>Hamid Sadeq Mahdi Alsultani, <sup>2</sup>Qusay Kanaan and <sup>1</sup>Inteasar Yaseen Khudhair

<sup>1</sup>Department of Computer Science, College of Basic Education, University of Diyala, Diyala, Iraq

<sup>2</sup>Department of Computer Science, Al Yarmouk University College, Diyala, Iraq

## Article history

Received: 20-12-2017

Revised: 20-02-2018

Accepted: 3-03-2018

Corresponding Author:

Hamid Sadeq Mahdi Alsultani

Department of Computer

Science, College of Basic

Education, University of

Diyala, Diyala, Iraq

Email:

hamid.sadeq.mehdi@basicedu.uodi  
yala.edu.iq

**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 Transmission Control Protocol/Internet Protocol (TCP/IP) for reliable transfer of data traffic between the cloud end-users and servers and vice-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. Transmission Control Protocol incast problem happens in cluster based storage workloads where a lot of end-users communicate simultaneously to a server in the cloud through a bottleneck router, creating buffers overflows which lead to high packet loss. This paper presents an empirical study on TCP incast in current wireless cloud networks and how it is caused. It evaluates TCP-Vegas and TCP-Sack to examine their behaviors and suitability for short-lived connections in terms of queue occupancy level, packet drops, throughput, link utilization and bandwidth unfairness between the TCP connections. It was found that both protocols suffer from high packet loss and link underutilization with comparable throughput.

**Keywords:** Internet Congestion Control and Avoidance, Queue Management, TCP Incast, Wireless Cloud Networks, Cloud Computing

## Introduction

Cloud computing has become very popular in recent years. The propagation of networked devices and offered services over Internet has given the rise to huge amount of data that need to be processed and stored. According to studies by an independent consulting firm, the need for cloud computing will reach about 22 million users by 2020, with annual growth projected to grow % 26 percent annually (Yassin *et al.*, 2017). It is impossible to be processed or stored by one computer or even a small group of computers (Puthal *et al.*, 2015). Thus, cloud computing has emerged and become widely popular as a global technology for cost effective resource availability and management to achieve significant performance improvements in analyzing such large-scale data. Well-known organizations such as Amazon, Google, Microsoft, HP and IBM have been deploying on-demand “Clouds” for all required software (services) around the world (Staudinger *et al.*, 2014; Botta *et al.*, 2016) and host

applications that produce numerous data such as scientific computing, social networks, e-commerce, web search, retail and distributed files systems, to name a few (Hammadi and Mhamdi, 2014).

In cloud computing, Quality of Service (QoS) for network communication should be guaranteed. Communication is an essential component of the cloud infrastructure. Thus, it is necessary for cloud systems to provide reliable communication along with processing ability and sufficient bandwidth. In the direction of this goal, Communication as a Service (CaaS) layer is responsible of meeting such requirements, through for example dynamic provisioning of bandwidth and network monitoring. Transmission Control Protocol (TCP) is underlying transport protocol that is commonly used in cloud networks due to its reliability of data delivery (Kulkarni *et al.*, 2013; Garai *et al.*, 2015).

Even though TCP has succeeded in satisfying the requirements of most of Internet-based communications, however, its performance has been degraded in the cloud environment where high burst tolerance, low latency and

high throughput are required. Furthermore, the distributed nature of wireless cloud computing environment and the requirement for guaranteed Quality of Service imposes additional challenges and complexities (Ren and Schaar, 2014). TCP end-to-end connections are affected by the movement of virtual machines in the cloud and the frequent topology changes of the wireless network due to mobility. Also, TCP communications via low quality wireless connections are unable to offer high throughput; especially with large communication delay (Li *et al.*, 2015).

Most of the traffic in wireless cloud networks uses many-to-one communication pattern such as distributed storage systems, data-intensive scalable computing systems and partition/aggregate workflows (Tahiliani *et al.*, 2015). This type of communication with such huge amount of data resulted in TCP incast problem that is widely exist in cloud environments.

An empirical study on TCP incast in current wireless cloud networks and its causes are presented. Different TCP versions are evaluated, namely TCP-Vegas and TCP-Sack, which are usually used in wireless cloud computing environment, to examine their behaviours with short-lived connections that are commonly exist in web search, social network content composition and advertisement selection etc. The way that is used to evaluate these two TCP versions has done by simulating short-lived TCP connections (mice). In these TCP connections, data are sent by thousands of clients simultaneously for processing at data centre in a cloud with many servers-connected to a router for implementing drop-tail mechanism. The rest of the paper is organized as follows: congestion in wireless cloud computing environments and TCP incast problem is discussed. Then, the performance evaluation of TCP-Vegas and TCP-Sack

in wireless cloud networks in terms of queue occupancy level, packet drops, throughput, link utilization and bandwidth unfairness between the TCP connections; and how the TCP incast problem appears with the use of these two protocols is presented. Finally, the conclusion of the research work is presented and some suggestions are provided for future work.

### Congestion and TCP Incast Problem

Performance issues of cloud services are related to some QoS metrics such as queue length fluctuation, throughput and bandwidth utilization; where network congestion is the main reason. Congestion has very serious effect on the quality of services expected by cloud end-users (Bae *et al.*, 2014; Elakkiya *et al.*, 2016; Mohammed *et al.*, 2017).

Congestion phenomena occur when the traffic load on the wireless cloud is higher than the available bandwidth, which may cause buffer overflow, large delays and high packet loss (Bai *et al.*, 2014). Also, as cloud services in wireless cloud computing are accessed throughout Internet via wireless medium from distant locations, Round Trip Time (RTT) of data transmission is usually long (Isobe *et al.*, 2014). While accessing cloud services through Internet increases, packet loss will be increased as well. It is due to queue overflows of routers (Fig. 1) at the bottleneck links (insufficient bandwidth) or the confliction with other data traffic.

Moreover, bit error can cause packet loss over wireless-network (Tian *et al.*, 2005; Benkikeri *et al.*, 2014). As end users commonly use TCP to communicate with the wireless cloud over wireless networks with long RTT and potentially high packet loss ratios, this would definitely declines access quality of the cloud service and decrease the throughput.

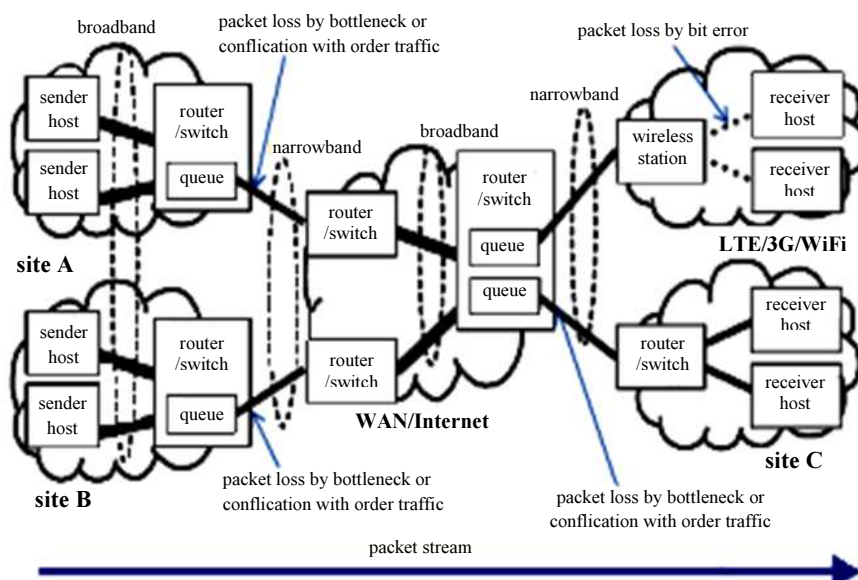


Fig. 1: Packet loss resulted from overflowed bottleneck-routers and bit error in wireless cloud networks (Isobe *et al.*, 2014)

TCP incast problem happens in many-to-one communication paradigm such as cluster based storage workloads where a lot of end-users (senders) communicate to a server (or few servers) in the cloud (data centers) through a bottleneck router. As these users send their traffic simultaneously, creating buffers overflows which lead to high packet loss (Hsu *et al.*, 2014; Kadhim *et al.*, 2017). TCP incast problem happens in either forwarding path (towards the servers in the cloud) or reverse path (towards the cloud end-users) or in both.

As data centers in the cloud are distributed systems, many-to-one communication such as partition/aggregate pattern is followed inside the centers, where received used requests are aggregated by aggregator and then partitioned and send to several workers (Fig. 2) for processing (Chen *et al.*, 2015; Kadhim *et al.*, 2017).

Once requests are received, workers concurrently response with a big amount of data that pass through a bottleneck link towards the aggregator.

The instantaneous and concurrent response from workers is due to the tight time bounds imposed by real-time requirements for the applications at the end-users. Thus, the data arriving at the aggregator port overflow the buffers and lead to high packet loss. This requires workers to perceive the loss, re-send the lost data (which is only done after the Retransmission Timeout (RTO)) and slowly increase the transmission rate. Thus, the back off of the workers contributes to significant fluctuation in the throughput.

As web mice traffic (short-lived connections) such as search and updates require short response time and they are delay sensitive, the frequent timeouts caused by TCP incast and the retransmission delay of the lost packets considerably degrade the performance of such connections.

According to (Adesanmi and Mhamdi, 2015), congestion can be avoided if the traffic load to a gateway is kept as closer as possible to the outgoing link capacity. Furthermore, the gateway queue size should be maintained to smaller level in order to ensure the buffer availability for accommodating the temporary increase in traffic load, which would otherwise results in buffer over- flow and in turn high packet loss (Premalatha and Natarajan, 2011; Maripalli and Abirami, 2014).

Congestion control is the responsibility of end-user devices and the gateways of the wireless clouds and Internet (Gholami and Akbari, 2016). Gateways are equipped with mechanisms to delay or drop data packets when necessary. They are responsible for queue length control, packet arrival control, congestion detection and notification. End-user devices are responsible of tuning their transmission rates to enable gateways to manage and control the congestion at the cloud (Jaworski, 2013; Dhingra *et al.*, 2015; Baker and Fairhurst, 2015).

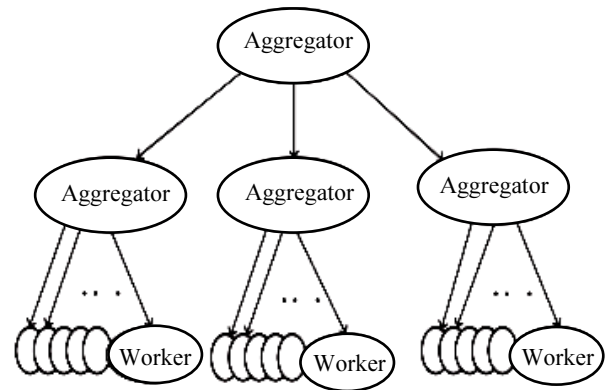


Fig. 2: Many-to-one communication in data centers

### ~~Performance Evaluation of TCP Vegas and TCP Sack~~

~~Aiming at investigating the performance and behaviour of TCP Vegas and TCP Sack protocols 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 centre in a cloud with several servers connected to a gateway (router) that implement drop tail mechanism. The purpose of the scenario is to explore and study the TCP incast problem where 2000 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 2 Gbps link with delay of 25μsec. 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 is 1 Kbytes. The arrival of new TCP sessions flow Poisson process. The arrival time between new sessions is exponentially distributed and selected to be 40msec, 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. Thus:~~

~~$$\text{Required Bandwidth} = \text{sessions} * \text{end users} * \text{transmission rate (bits / second)} \quad (1)$$~~

~~gives the required bandwidth for such number of users. It is as follows:~~

~~25 sessions \* 2000 users = 50,000 sessions per second;  
50,000 sessions \* 10 Kbytes (transmission rate) = 500,000 Kbytes \* 1024 Bytes \* 8 Bits = 4,096,000,000 bits per second = 4,096,000,000/1000,000,000 = 4096 Gbps, which is obviously higher than the available bandwidth to the cloud (2 Gbps). The Retransmission Timeout (RTO) is set to 200 m sec. The router buffer size is 128 packets the simulation time is 100 sec.~~

The evaluation experiments were conducted using network simulator 2 on CentOS 5.2 Linux operating system. Based on the numerical results gained from experiments, the performance of TCP Vegas and TCP Sack was evaluated in terms of queue occupancy level, packet drops, throughput, link utilization, bandwidth unfairness between the TCP connections.

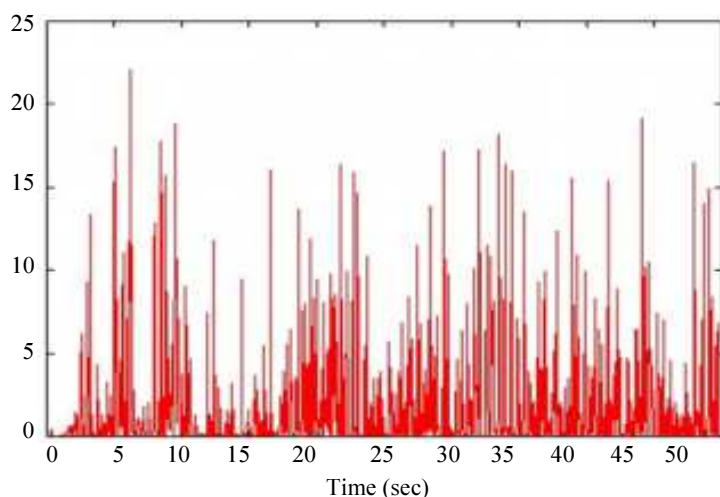
### Queue Occupancy Level

The simulation results presented in this section represents the queue occupancy level at the router for TCP Vegas and TCP Sack for the first 50 sec of the simulation time before they halt. Figure 3 shows the router queue occupancy level per connection when TCP Vegas is used. From the figure, it is noticeable that there is a high fluctuation in the queue occupancy level which is corresponding to the frequent timeouts caused by severe packet drops by the router when overflow events take

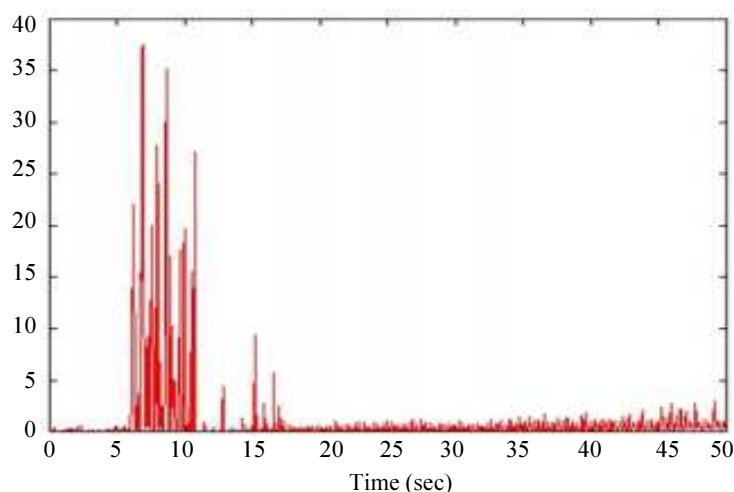
place. This would make the senders to basically adjust their transmission rates in order to avoid congestion.

The results confirm the occurrence of such event that leads to high packet loss. The late sensing of the packet loss and retransmission of the lost packets result justify fluctuation in the queue occupancy level. The high fluctuation shown in the queue occupancy level means that there is a considerable queuing delay which would affect the quality of service required by the application of that connection.

Figure 4 shows the router queue occupancy level per connection when TCP Sack is used. Compared to the result when TCP Vegas, the queue occupancy level of a TCP Sack connection shows smoother fluctuation, except for the for the time interval between 5 to 10 sec of the simulation time, where the queue level is high. This means that the connection is seriously affected by the router queue management mechanism used and the behavior of TCP Sack in response to packet drop.



**Fig. 3:** Queue occupancy level per connection using TCP-Vegas



**Fig. 4:** Queue occupancy level per connection using TCP-Sack

However, the queuing delay experienced by a TCP-Sack connection is less compared to that of TCP-Vegas.

### Packet Drops

Packet drops results in noticeable packet loss which significantly affects cloud end-users applications. Figure 5 shows the number of packets that were dropped by the queue management mechanism in the router.

In case of TCP-Vegas, the packet lost observed throughout the experiments is 14000 packets, compared to that of TCP-Sack, which is about 22000 packets (Fig. 6).

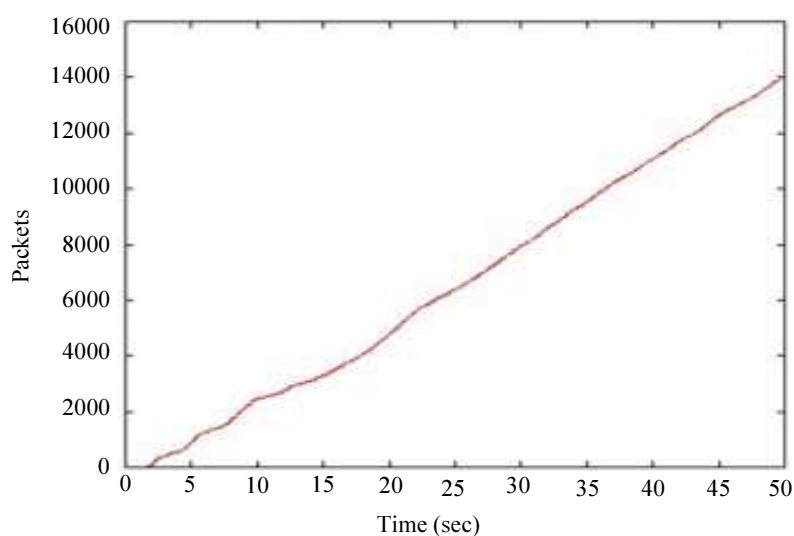
The results shown in Fig. 6 confirms observed queue occupancy level depicted in Fig. 4, as there is less number of data packets entering the buffer compared with TCP-Vegas (shown in Fig. 3).

### Throughput

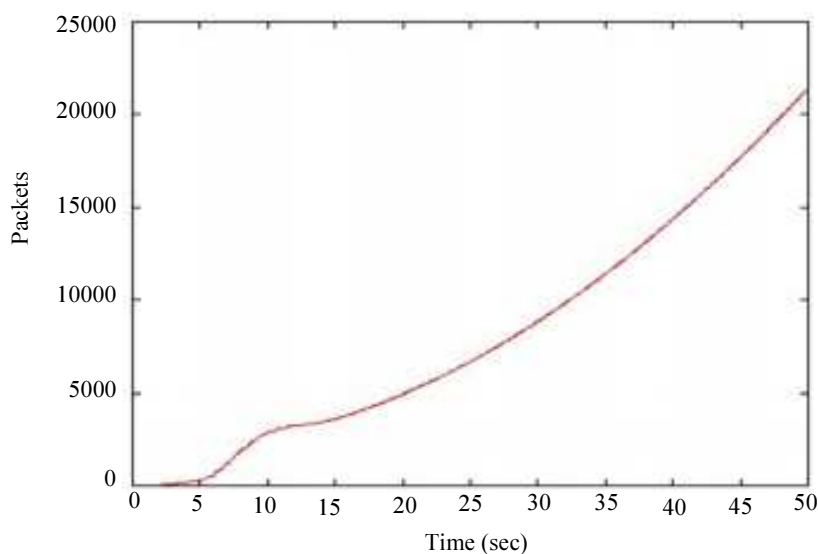
Figures 7 and 8 show the total throughput gained for first 100 connections before they back off for TCP-Vegas and TCP-Sack. The fluctuation in throughput shown in Fig. 7 is due to the frequent back off of the TCP-Vegas senders as a reaction to packet drops.

The behaviour of TCP-Vegas where the sender dose not wait for a timeout and that when the actual gained throughput is less than the expected throughput, the sender would decrease the congestion window, all resulted in the observed throughput.

On the other hand, TCP-Sack sender-response to packet loss, where it fast retransmits only the missing data, contributes to the resulted throughput.

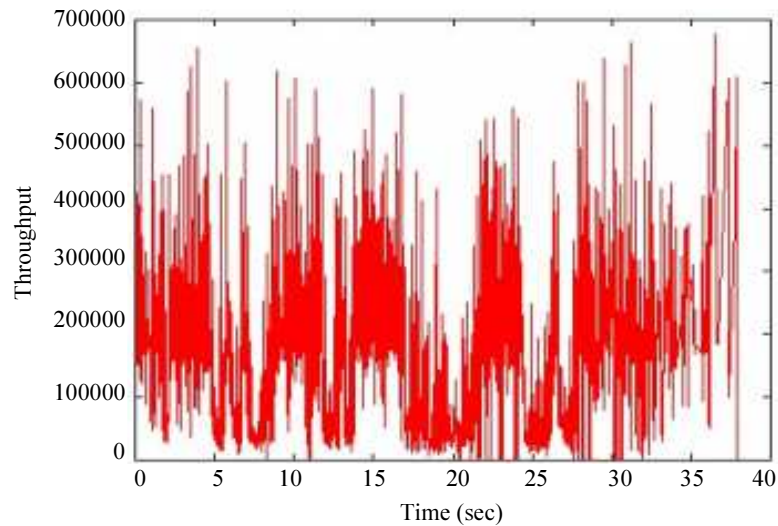


**Fig. 5:** Packet dropped at the router for the first 50 sec using TCP-Vegas

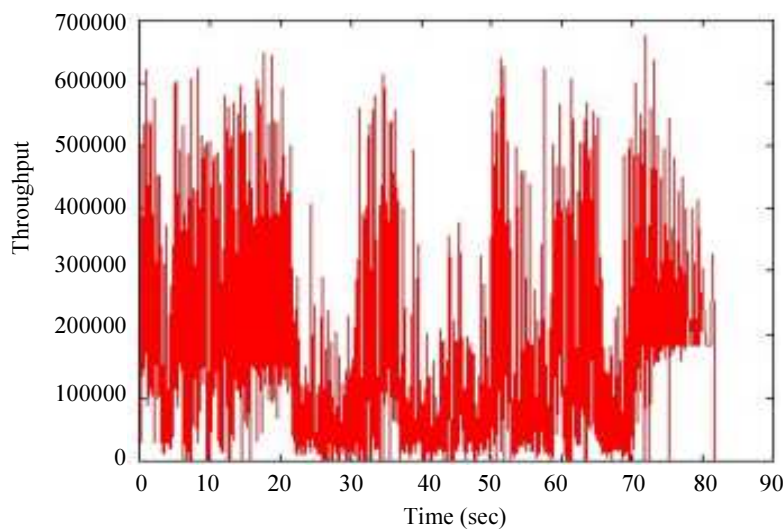


**Fig. 6:** Packet dropped at the router for the first 50 sec using TCP-Sack

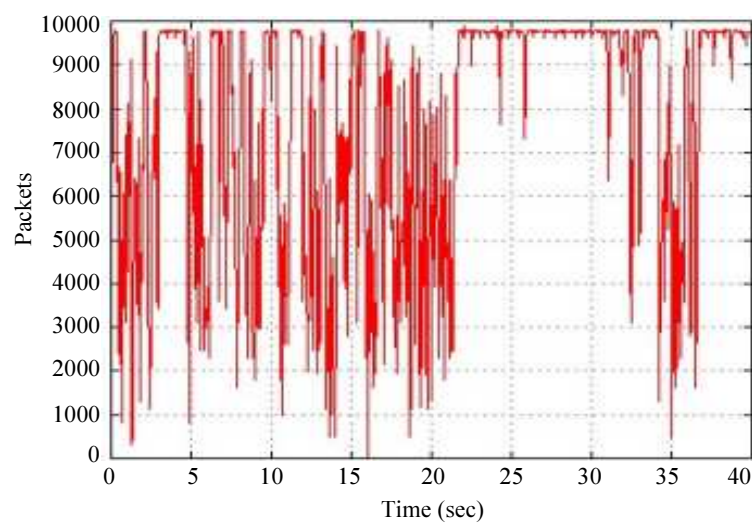




**Fig. 7:** Total throughput gained for first 100 connections before they back-off using TCP-Vegas



**Fig. 8:** Total throughput gained for first 100 connections before they back-off using TCP-Sack



**Fig. 9:** Link utilization per connection for 40 sec before it halts using TCP-Vegas

### Link Utilization

Through the evaluation experiment, it was found that both TCP-Vegas and TCP-Sack present considerable underutilization of the link capacity because of the queue management mechanism used in the router.

The data resulted from the evaluation experiment showed that drop-tail causes serious global synchronization problem where users reduce their transmission rate simultaneously causing irregular busy-idle phases (buffer overflow followed by almost empty buffer) over the link to the servers in the cloud. This problem reduces the utilization of the link and causes degradation in network performance (due to high delay, high packet loss and low throughput).

Also, it was found that this problem not only arises on the bottleneck link from cloud end-users to the servers, but also on the bottleneck from servers to the end-users, as both sides use drop-tail in the router.

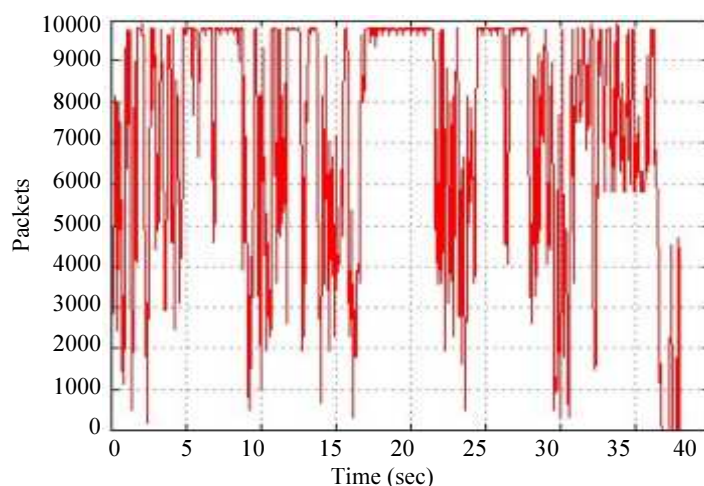
The resulted data from the experiment confirmed that drop tail affects the performance of short-lived-connections considerably, because of frequent packets timeouts that cause high delay and, accordingly, slow response times.

Figures 9 and 10 illustrate the link utilization per connection for 40 sec for both TCP-Vegas TCP-Sack connections before they halt.

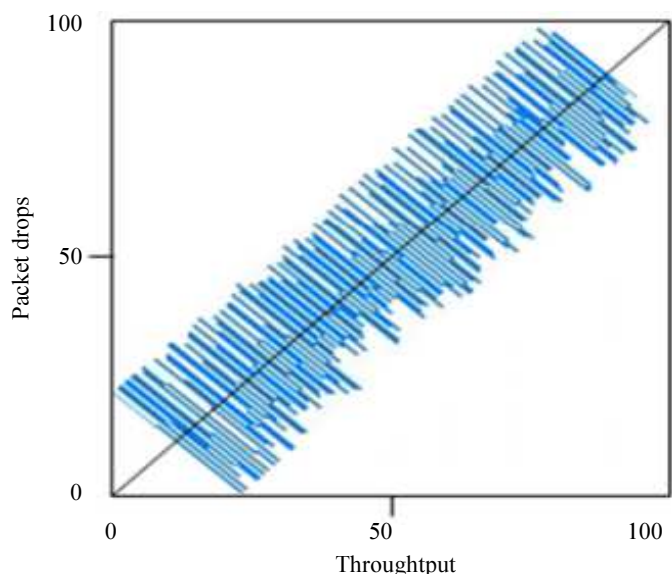
### Bandwidth Unfairness

The experimental results of the bandwidth unfairness between the TCP connections for TCP-Vegas and TCP-Sack experiments, with respect to the perfect fairness line (diagonal line), are shown in Fig. 11 and 12, respectively.

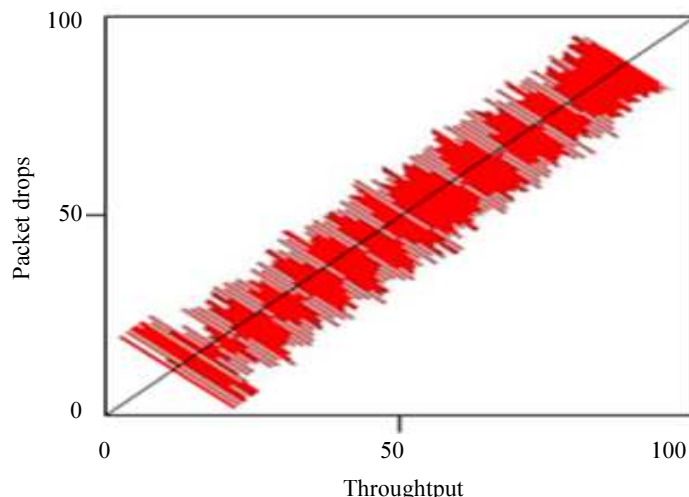
The line characterizes the points where the percentage of throughput and packet drops for a particular TCP connection are identical. Any connection whose point falls on the line is treated fairly.



**Fig. 10:** Link utilization per connection for 40 sec before it halts using TCP-Sack



**Fig. 11:** Bandwidth unfairness between connections depending on throughput and packet drops using TCP-Vegas



**Fig. 12:** Bandwidth unfairness between connections depending on throughput and packet drops using TCP-Sack

## Conclusion

The aim of this paper is to discuss congestion in cloud computing environments and TCP incast problem. The performance evaluation of two different TCP versions, called TCP-Vegas and TCP-Sack in wireless cloud networks is evaluated in terms of queue occupancy level, packet drops, throughput and link utilization. It was shown how the TCP incast problem appears with the use of these two versions. With regards to queue occupancy level; it is clear that high fluctuations mean that there is an overflow in the router queue occupancy level which causes severe packet drops. The lost packets must be retransmitted, so a queuing delay is considered; which means that the senders need to adjust the retransmission rate. TCP-Vegas has more response to packet loss than TCP-Sack. As a result of the achieved response, it is noticeable that packet drops in TCP-Vegas is less than TCP-Sack.

With regards to throughput; TCP-Vegas has less throughput than TCP-Sack, because the senders must decrease the congestion window to avoid packet drops, which leads to decrease the throughput altogether. According to the high delay, the high packet drops and the low throughput, the link utilization will reduce in both TCP-Vegas and TCP-Sack. This reduction in link utilization affects the network quality, especially short-lived-connections.

With regards to bandwidth unfairness between the TCP connections; in both TCP-Vegas and TCP-Sack, the connection is treated fairly if the percentage of throughput and packet drops is the same. The throughput is directly proportional with packet drops.

It was found that TCP incast problem still exists which causes high packet loss, which requires end-users to perceive the loss, re-send the lost data and slowly increase the transmission rate. Hence, the back-off of TCP senders makes considerable fluctuation in the throughput per connection.

However, TCP-Vegas is better than TCP-Sack in terms of queue occupancy level, packet drops, throughput, link utilization and bandwidth unfairness.

In future work, TCP-Westwood and Data Center TCP (DCTCP) versions of TCP are going to be investigated with RED queue management mechanism to examine their suitability for wireless cloud environments and provide suggestion to improve the performance of such environment.

## Acknowledgement

This research was not funded by any institution.

## Author's Contributions

**Hamid Sadeq Mahdi Alsultani:** Participated in the research plan, the study organization, the manuscript writing and implementing all experiments.

**Qusay Kanaan:** Participated in analysing the experimental results.

**Intesar Yaseen Khudhair:** Participated in Performing the revision of the manuscript.

## Ethics

The corresponding author confirms that there are no ethical issues or conflict of interest.

## References

- Adesanmi, A. and L. Mhamdi, 2015. Controlling TCP incast congestion in data centre networks. Proceedings of the IEEE International Conference on Communication Workshop, Jun. 8-12, IEEE Xplore Press, London, pp: 27-32.  
DOI: 10.1109/ICCW.2015.7247446



- Bae, S.Y., S.K. Lee, J.G. Koh and K.W. Park, 2014. QoS routing method considering congestion in WMSNs. *Int. J. Multimedia Ubiquitous Eng.*, 9: 309-316. DOI: 10.14257/ijmue.2014.9.4.32
- Bai, W., K. Chen, H. Wu, W. Lan and Y. Zhao, 2014. PAC: Taming TCP incast congestion using proactive ACK control. *Proceedings of the IEEE International Conference on Network Protocols*, Oct. 21-22, IEEE Xplore Press, Raleigh, pp: 385-396. DOI: 10.1109/ICNP.2014.62
- Baker, F. and G. Fairhurst, 2015. IETF recommendations regarding active queue management. *Tech. Report*. DOI: 10.17487/RFC7567
- Benkikeri, S.N., V. Budyal and M.M. Kodabagi, 2014. Techniques for loss recovery in wireless networks based on acknowledgments. *Int. J. Electrical, Electronics Data Commun.*, 2: 67-71.
- Botta, A., W. Donato, V. Persico and A. Pescapé, 2016. Integration of cloud computing and internet of things: A survey. *Future Generation Comput. Syst.*, 56: 684-700. DOI: 10.1016/j.future.2015.09.021
- Chen, W., F. Ren, J. Xie, C. Lin and K. Yin *et al.*, 2015. Comprehensive understanding of TCP incast problem. *Proceedings of the IEEE Conference on Computer Communications*, Apr. 26-May 1, IEEE Xplore Press, Kowloon, pp: 1688-1696. DOI: 10.1109/INFOCOM.2015.7218549
- Dhingra, M., M. Sharma and V. Gupta, 2015. Congestion control within working ability and capacity limit in cloud computing environment. *Int. J. Adv. Eng. Sci.*, 5: 12-15.
- Elakkiya, A., B.S. Krishnan and M. Ramaswamy, 2016. Congestion control improved QoS routing scheme for wireless sensor network. *Global J. Eng. Sci. Res. Manage.*, 3: 104-111. DOI: 10.5281/zenodo.56717
- Garai, M., S. Rekhis and N. Boudriga, 2015. Communication as a service for cloud VANETs. In *IEEE Symposium on Computers and Communication*, Jul. 6-9, IEEE Xplore Press, Larnaca, pp: 371-377. DOI: 10.1109/ISCC.2015.7405543
- Gholami, M. and B. Akbari, 2016. Congestion control using open flow in software defined data center networks. *Proceedings of the International Conference on Innovations in Clouds, Internet and Networks*, (CIN' 16), pp: 1-5.
- Hammadi, A. and L. Mhamdi, 2014. A survey on architectures and energy efficiency in data center networks. *Comput. Commun.*, 40: 1-21. DOI: 10.1016/j.comcom.2013.11.005
- Hsu, C.H., J. Ma and M.S. Obaidat, 2014. Dynamic intelligence towards merging cloud and communication services. *Inform. Syst. Frontiers*, 16: 1-5. DOI: 10.1007/s10796-014-9488-3
- Isobe, T., N. Tanida, Y. Oishi and K. Yoshida, 2014. TCP acceleration technology for cloud computing: Algorithm, performance evaluation in real network. *Proceedings of the International Conference on Advanced Technologies for Communications*, (ATC' 14), pp: 714-719. DOI: 10.1109/ATC.2014.7043480
- Jaworski, P., 2013. Cloud computing based adaptive traffic control and management. *PhD Thesis*, CoventryUniversity.
- 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. *Int. J. Commun. Antenna Propagation*, 7: 357-363. DOI: 10.15866/irecap.v7i5.10104
- Kulkarni, G., M. Jadhav, S. Bhuse, H. Bankar and A. Sushma, 2013. Communication as service cloud. *Int. J. Comput. Network.*, 3: 149-156.
- Li, W., Y. Zhao, S. Lu and D. Chen, 2015. Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing. *IEEE Commun. Magazine*, 53: 89-97. DOI: 10.1109/MCOM.2015.7060487
- Maripalli, S. and G. Abirami, 2014. Congestion control for TCP in data-center networks. *Int. J. Comput. Sci. Inform. Technol.*, 5: 1903-1907.
- Mohammed, H., G. Attiya and S. El-Dolil, 2017. Active queue management for congestion control: performance evaluation, new approach and comparative study. *Int. J. Comput. Network Technol.*, 5: 37-49.
- Premalatha, N. and A.M. Natarajan, 2011. Congestion control in wireless ad hoc networks by enhancement of transmission control protocol. *J. Comput. Sci.*, 7: 1824-1830. DOI: 10.3844/jcssp.2011.1824.1830
- Puthal, D., B. Sahoo, S. Mishra and S. Swain, 2015. Cloud computing features, issues and challenges: A big picture. *Proceedings of the International Conference on Computational Intelligence and Networks*, Jan. 12-13, IEEE Xplore Press, Bhubaneshwar, pp: 116-123. DOI: 10.1109/CINE.2015.31
- Ren, S. and M. Schaar, 2014. Dynamic scheduling and pricing in wireless cloud computing. *IEEE Trans. Mobile Comput.*, 13: 2283-2292. DOI: 10.1109/TMC.2013.57
- Staudinger, M., D. Nedbal, W. Wetzlinger, G. Wagner and M. Erskine, 2014. Impacts on the organizational adoption of cloud computing: A reconceptualization of influencing factors. *Procedia Technol.*, 16: 85-93. DOI: 10.1016/j.protcy.2014.10.071

- Tahiliani, R., M. Tahiliani and K. Sekaran, 2015. TCP Congestion Control in Data Center Networks. In: Handbook on Data Centers, Khan S. and A. Zomaya, (Eds.), Springer, New York, ISBN-10: 1493920928, pp: 485-505. DOI: 10.1007/978-1-4939-2092-1\_15
- Tian, Y., K. Xu and N. Ansari, 2005. TCP in wireless environments: Problems and solutions. IEEE Commun. Magazine, 43: 27-32.  
DOI: 10.1109/MCOM.2005.1404595
- Yassin, I., M. Zan, N. Tahir, H. Abidin and Z. Rizman, 2017. Integration of CompTIA cloud+ into universiti teknologi mara's computer engineering special topics syllabus. Int. J. Adv. Sci. Eng. Inform. Technol., 7: 22-27, 2017. DOI:10.18517/ijaseit.7.1.941