On the Performance of Fog-Cloud Computing for Real-time Surveillance Applications

Saleh Abdalla, Mohamed S. Hassan, Taha Landolsi, Ra’afat Abu-Rukba
American University of Sharjah
Sharjah, United Arab Emirates
{b00069937, mshassan, tlandolsi, raburukba}@aus.edu

***Abstract*—With the remarkable growth of IoT networks, there is a need for viable infrastructures that enable the successful deployment of IoT-based applications. Recently, cloud computing Postured itself as a dynamic promising structure to transfer and process the large amount data generated by the IoT nodes through the Internet. However, the applications are of stringent requirements, like video applications, the cloud fails to meet these requirements due to the latency problems experienced during heavy Internet traffic. Therefore, a more promising infrastructure is fog computing, which brings the cloud closer to the user through intermediary devices known as fog nodes. Therefore, the purpose of this paper is to study the viability of fog computing as a medium for IoT-based real-time surveillance applications. Specifically, we investigate and compare the performance of a classical cloud computing paradigm and a hybrid fog-cloud architecture. Our simulations shows that the hybrid model outperforms the classical model in terms of the number of requests meeting their deadlines and in terms of the forward trip time.**

***Index Terms*—Fog computing, cloud computing, IoT.**

I. INTRODUCTION

The recent increase in the number of interconnected devices and sensors has led to the rapid adoption of the Internet of things (IoT) paradigm in the design of future smart cities. IoT has been defined as a “self-configurable global infrastructure that enables dynamic interconnectivity of physical and virtual things” [1]. To meet the computational requirements of real-time and delay-sensitive applications run by geographically extended topologies of IoT devices, the fog computing paradigm was introduced. While a classic cloud computing model, shown in Fig. 1, comprises a layer of IoT nodes and another layer of cloud computing servers, the fog computing model, shown in Fig. 2, adds an intermediate layer–the fog one–between the IoT and the cloud layers. This design paradigm brings the computational power closer to the IoT nodes, thereby reducing the networking delays that might hinder the performance of time-sensitive applications like video streaming applications [2]–[4]. While fog computing extends the concept of cloud computing, its objectives are relatively the opposite to those of cloud computing. For instance, fog computing offers a decentralized computing structure to improve the overall network efficiency and performance by bringing some of the basic analytic services to the edge of the network. This is done by bringing the computing resources closer to the data source(s) to reduce the network delays the data has to go through before it gets processed [5].

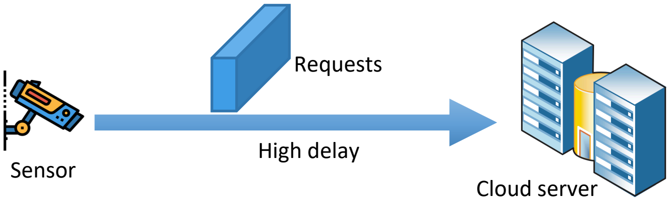


Fig. 1: Block diagram of the classic cloud computing model.

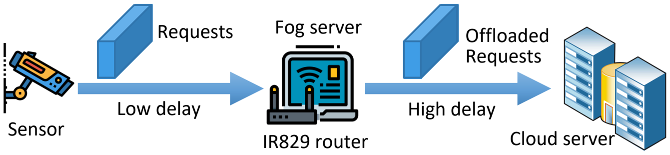


Fig. 2: Block diagram of the system based on the hybrid fog-cloud computing model.

Introducing an intermediate fog layer between the cloud and the IoT devices is justified by the fact that IoT devices and sensor networks are often distant from the cloud and are usually connected via the Internet, whose core routers are likely to introduce large delays. On the other hand, some IoT devices, running time-sensitive applications such as real-time surveillance, do require a great deal of processing as well as a speedy response. The network architecture of the fog layer can be seen as a miniature version of any typical computer network, thus consisting of a control plane and a data plane with varying topologies and designs. Therefore, while processing of time-sensitive requests on the cloud is not an efficient solution for certain IoT applications, the fog layer, from a networking perspective, is considered an inter-network with smaller and more predictable delays [6]. Obviously, this is a major advantage for IoT devices running time-sensitive applications. The fog-computing architecture contains end devices at its bottom layer such as sensors and actuators integrated with applications to enhance their functionality. These bottom layer devices utilize the network layer, which is the next layer to communicate through another layer with edge devices, such as gateways to finally reach the cloud and use its offered services [7]. Clearly, fog computing allows for the integration of edge and cloud resources with the objective of supporting intelligent processing of huge data volumes as generated by IoT-based applications in a decentralized manner.

Fog computing, however, is still evolving and thus yet to reach a wide acceptance and deployment status. It still needs to address several lingering challenges. For instance, it is not clear if a computational request is better off served by a nearby, but less powerful, fog node or whether it would be more advantageous for that request to be served by a more capable, but far away, cloud node. Tradeoff scenarios of this kind are currently the subject of ongoing research.

In this work, we propose to highlight the cases in which it would be beneficial, from a system performance perspective, to add an intermediate fog layer that will address IoT computational requests with stringent latency requirements. The study will take the rate of request completion as the main figure of merit. To this end, we adopt two simulation models: The first model represents the classical cloud computing paradigm with the IoT layer and the cloud computing layer. The second model adds the intermediate fog layer between the IoT and cloud layers.

The rest of the paper is organized as follows. Section II presents the details of the two models that are studied in this paper. The simulation parameters and assumptions are discussed in Section III, and the results are presented in Section IV. Finally, Section V summarizes and concludes the paper.

II. SYSTEM MODEL

In this section, we present the block diagrams for the system models adopted in this paper. The first model is as classical cloud computing paradigm as depicted in Fig. 1. The second model consider the hybrid fog-cloud architecture, with the intermediate fog layer between the IoT and cloud layers as illustrated in Fig. 2.

A. Classical cloud computing model

The overall description of the classical cloud computing model is shown in Fig. 1. The requests generated by an IoT edge node, such as a sensor or a surveillance camera, might require a quick response to a computational request. For instance, in the case of surveillance cameras, it could be a video frame that needs rapid image processing to identify potential threats. The requests need to go through a series of packet switching nodes such as IP routers to a remote cloud server that will process the requests. The responses to these requests will endure more or less the same delays in their way back to the IoT layer. A request for suspect identification is deemed successfully satisfied if its response comes back before a specified deadline. This deadline sets the overall time budget for all the delays encountered in the network side and the computing server side.

B. Hybrid fog-cloud computing model

A high-level system block diagram of the hybrid fog-cloud computing model is shown in Fig. 2. The difference from the previous model is in the introduction of an intermediate fog layer. The fog is characterized by the presence of servers with much lower computing power. However, IoT requests and responses will endure smaller network latencies when compared to the ones in the cloud model.

To overcome this dilemma, we deploy an straightforward scheduling algorithm by which the requests can either be offloaded to the cloud layer or “locally” processed by the fog layer. In our simulation, the outcome of the request scheduling function depends on the network latency and the estimated processing time by the cloud or the fog server.

III. SIMULATION SETUP

In this study, to gauge we assume that the communication channels in the fog and cloud networks are nearly ideal. Specifically, we assume that data packets carrying the IoT requests and responses do not suffer any losses in addition to an assumption that the channels do not introduce any bit errors. This is essentially can be achieved by assuming that the networks are not in or near a congestion state, and that the signal to noise ratio (SNR), over the communication channels, is large enough such that the bit error rate (BER) is negligible. Therefore, In the proposed model no retransmission requests are expected to take place, especially when a reliable protocol, such as TCP, is used at the transport layer protocol in the protocol stack.

Moreover, this study assumes that the IoT edge devices are surveillance cameras monitoring a geographical area of interest to a law enforcement entity. Their goal is to accurately identify known suspects among a crowd as fast as possible. In such a case, an IoT request is a video frame collected by the surveillance camera. The frame needs to undergo some image processing, and a face recognition algorithm needs to be deployed. Then, the results need to be compared using a back-end database where the adequate response needs to be generated. Additionally, it is also assumed that each video frame that comprises the faces of N persons will generate N requests. Moreover, it is assumed that the requests from all the edge surveillance cameras are generated randomly with the same average inter-generation times.

The simulation parameters in this study are as follows: in the cloud model, a truncated normal distribution with a mean $\mu_c = 3.25$ seconds and a standard deviation $\sigma_c = 0.75$ seconds is assumed for the transmission latency. An exponential distribution with an average of $\bar{\tau}_s = 1/4$ seconds is assumed for the service times of the generated requests. Finally, an exponential distribution with an average of $\bar{\tau}_g = 4$ s is assumed for the inter-generation times.

The distribution of the service times for requests processed in the fog is assumed to have the same shape as in the cloud case but with a larger value for the mean $\mu_f = 4$ seconds. In other words, the fog server would take, on average, 0.75 seconds longer than the cloud server to process any of the generated requests.

IV. SIMULATION RESULTS

This section discusses the simulation results for both the classical cloud computing model and the proposed hybrid cloud-fog model shown in Figs. 1 and 2, respectively. One important parameter of the simulation is the forward trip time (FTT), which consists of the sum of the transmission time and service times. To characterize the distribution of FTT, its cumulative distribution function (CDF) is computed numerically. The figure of merit, being the number of satisfied requests, is computed for different simulation scenarios.

A. Cloud computing model

Fig. 3 shows the CDF of the FTT of requests for different numbers of edge IoT cameras. It is clear that the CDF shifts rightward as the number of edge devices increases, which also means that a greater number of edge cameras results in longer FTT. The figure intuitively shows that as the number of cameras, i.e., end devices, increases the probability of not meeting the required delay also increases. This can be clearly seen from the clear shift in the CDF for the case of 4 edge cameras. It also shows that any further increase in the number of edge cameras will result in a greater shift in the CDF.

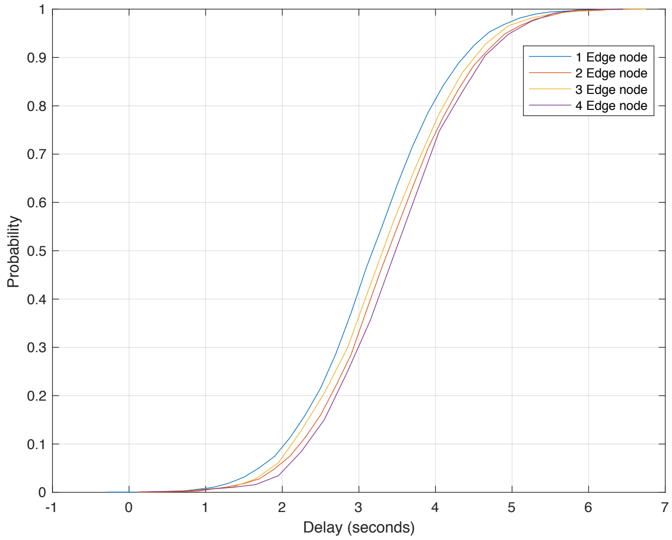


Fig. 3: CDF of FTT used in the cloud computing model.

Fig. 4 shows the number of satisfied requests, which is the number of requests for which its delay requirements are met, versus time for different number of edge nodes (cameras). The rate at which IoT camera requests are satisfied by the cloud servers in requests/seconds is summarized in Table I. Similarly, as the number of end devices increases, the number of requests increases accordingly. However, it is expected that there will be a diminishing return in the rate of request satisfaction when the edge nodes increase beyond a threshold value.

TABLE I: Rate of requests successfully satisfied in the classical cloud model.

No. of nodes	**Request rate (Req/s)**
1	0.1881
2	0.2529
3	0.3075
4	0.4380

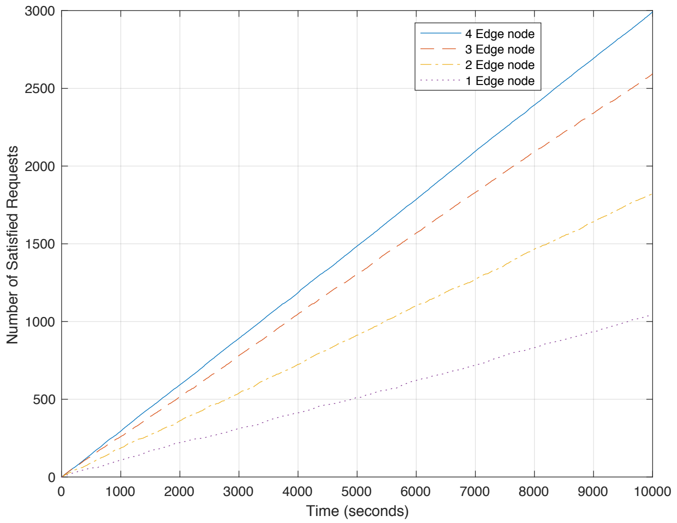


Fig. 4: Request satisfaction rate for the cloud computing model.

B. Fog-cloud computing model

A similar study is conducted but for the proposed fog-cloud model, which is a hybrid fog-cloud architecture, with an intermediate fog layer between the IoT and cloud layers as shown in Fig. 2. Similar to Fig. 4, Fig. 5 shows the CDF of the FTT for requests for different numbers of edge IoT devices (cameras). This figure clearly shows the advantage of the proposed fog-cloud model over the classic cloud structure. Specifically, the proposed fog-cloud model is able to satisfy a more stringent delay requirement even for the case of increased number of IoT edge devices.

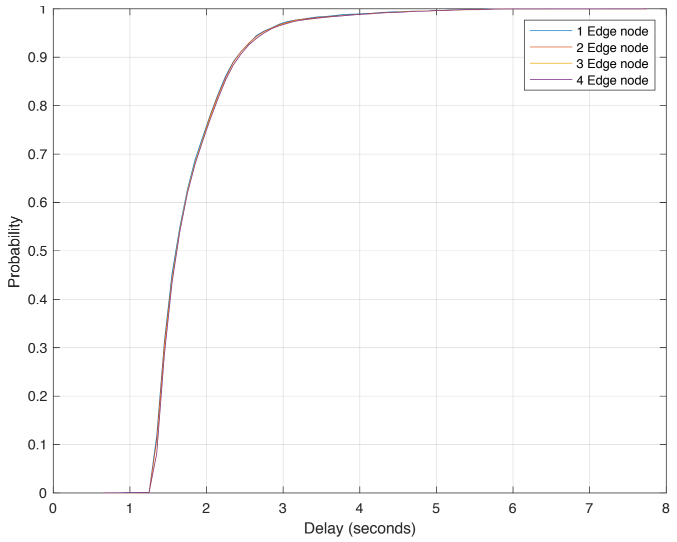


Fig. 5: CDF of the FTT used in the fog-cloud computing model.

Several interesting remarks can be noticed in Fig. 5. First, the shift between the CDF for 1 edge node and 2 edge nodes is around 10 ms. This could be considered a significant figure

TABLE II: Rate of requests successfully satisfied in the hybrid fog-cloud model.

No. of nodes	Req/s (Fog)	Req/s (Cloud)	Total
1	0.2848	0.0155	0.3003
2	0.4237	0.0299	0.4536
3	0.6448	0.0641	0.7089
4	0.8976	0.1445	1.0421

from the perspective of time-sensitive real-time applications. Also, the CDF is steeper for low delay values, which implies a skewed to the right PDF or simply it shows that lower delay values have high probability of occurrence. It also shows that the proposed fog-cloud model can meet the deadlines of the different requests in a manner that is to some extent insensitive to the number of edge devices. This clearly is a desirable feature that enables future extensions. Furthermore, Fig. 6a and Fig. 6b show the number of satisfied request versus time for the cloud layer and the fog layer, respectively. Additionally, Table II records the gradient for each line.

Clearly, the rate of satisfied requests is increasing with the increase in the number of edge nodes, albeit not in a linear fashion as was the case in the previous model. This indicates that the system is yet to reach a saturation point. More importantly, contrasting the rate values of Table I and Table II we can clearly see that the deployment of the fog computing model will increase the rate of satisfied requests, as well as decrease the overall latency. Another interesting point to mention is that as the number of edge nodes increases, the number of requests that are routed to the cloud increases, which can be seen in the difference in slopes between Fig. 6a and Fig. 6b.

V. CONCLUSIONS

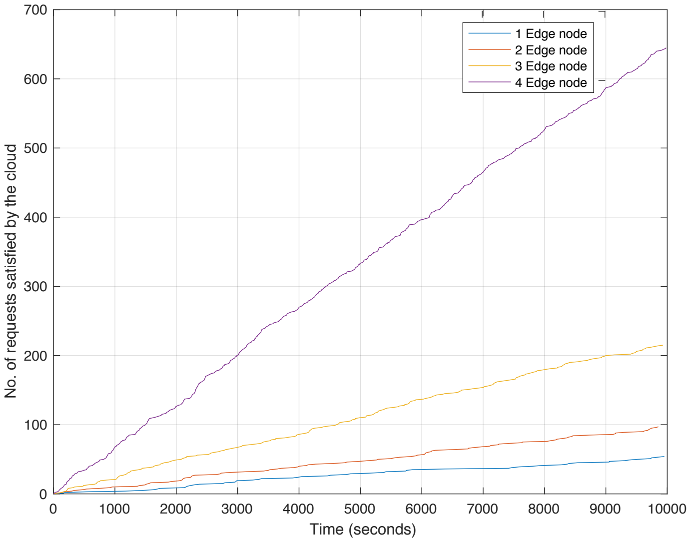
In this paper, a hybrid fog-cloud computing approach was proposed for delay-sensitive surveillance applications that also require intensive data processing. The study modeled the identification of suspect individuals in the captured surveillance videos as a computational request sent by the edge camera to either the cloud or fog servers. The simulation results indicated that the deployment of fog computing layer is beneficial, since it increases the request satisfaction rate and reduces the overall latency.

REFERENCES

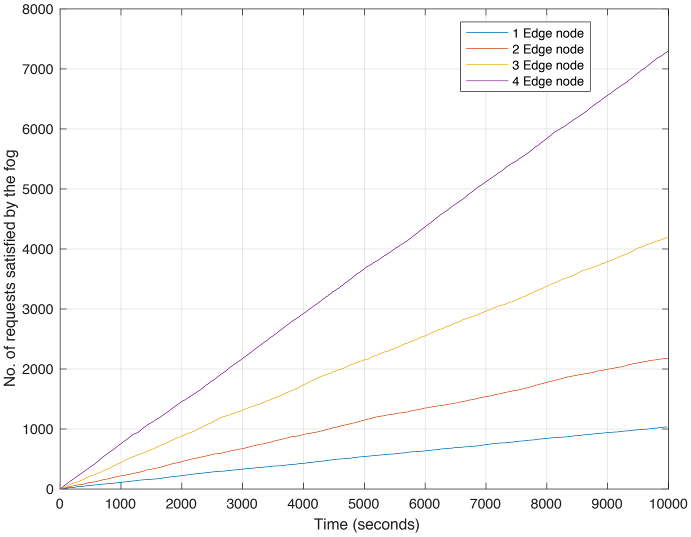
[1] R. Kranenburg and S. Dodson, "The Internet of Things: A Critique of Ambient Technology and the All-Seeing Network of RFID," *Institute of Network Cultures*, 2008.

[2] M. S. Hassan, D. Hussein, and T. Shanableh, "A priority and path diversity-based approach for media streaming over multi-hop and unreliable networks," *International Journal of Wireless and Mobile Computing*, vol. 19, no. 1, pp. 18–32, 2020. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJWMC.2020.109248

[3] M. Hassan and M. Krunz, "Video streaming over wireless packet networks: An occupancy-based rate adaptation perspective," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 8, pp. 1017–1027, 2007.



(a) Request rate for "cloud" layer in the hybrid fog-cloud computing model.



(b) Request rate for "fog" layer in the hybrid fog-cloud computing model.

[4] M. Hassan, L. Atzori, and M. Krunz, "Video transport over wireless channels: A cycle-based approach for rate control," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 916–923. [Online]. Available: https://doi.org/10.1145/1027527.1027736

[5] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing architecture, evaluation, and future research directions," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 46–52, May 2018.

[6] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47 980–48 009, 2018.

[7] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug 2016.