Colored Ant for Flow Management in Fog Computing

Kaouthar Sethom,
INNOV'COM Lab, SUPCOM,
University of Carthage, Tunisia

firstname.surname@enicarthage.rnu.tn

Abstract: **Nowadays, users' demands are increasing in terms of data volume, interactivity, and multimedia quality. This put higher challenges on providers. Cloud operators are trying to improve network performance to fulfill application requirements and satisfy customers. However, this objective is still increasing in difficulty. Software-Defined Network (SDN) offers a new vision of the networking paradigm with more management flexibility. It simplifies networking by providing global visibility through a central controller. In this paper, we propose a new QoS aware flow management based on the SDN technique and Ant optimization algorithm in the scenario of Fog computing. The per-flow decision is taken more efficiently in the centralized controller based on specific metrics and flows classification. Performance experiments based on our implementation shows promising results.**

Keywords: Ant colony, Cloud, Quality of Service, SDN.

1. INTRODUCTION

Today 's mobile users are asking for more data access from any device, any time, from anywhere. This means that the future of the cloud must support the idea of the ―Internet of Thing (IoT); That's where Fog Computing comes in. Fog aims to distribute data and move it closer to the end-user to eliminate latency and numerous hops and support mobile computing and data streaming. Fog computing represents the edge cloud.

We believe that the success of Fog computing infrastructures will depend on how effectively these infrastructures will be able to instantiate and dynamically maintain computing platforms, resources and cloud services that must meet varying QoS costumer applications. Users expect applications to be highly available with minimum service disruption. Mobility The applications' QoS requirements need to be ensured even when there are changes in the network like link down, and switch failures. A good routing protocol will help the Fog provider in meeting this purpose. The idea in this paper, is to propose a new routing algorithm using an SDN-based Fog Framework.

To enhance the system performance, we propose a new per-Flow management-based Ant optimization algorithm. Our algorithm will run on the top of Fog nodes (FNs) to provide the better path for each flow depending on some QoS and applications requirements (Figure 1).

The Fog Nodes closest to the network edge ingest the data from Internet of Things (IoT) devices. FNs can be deployed anywhere with a network connection: on a factory floor, in a vehicle, or on body area network. Any device with computing, storage and network connectivity can be an FN at the network edge. Fog node will be a new station that minimizes latency between cloud and mobile users, saves the backbone bandwidth and saves the energy consumption of core networks. The out-put of our algorithm will be a colored network graph where each colored path is specific to a class of traffic on the Fog architecture.

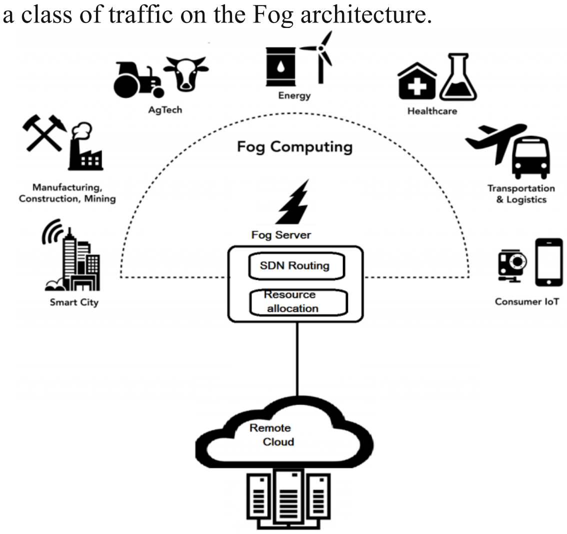

Figure 1: Fog Computing Architecture

2. MOTIVATION SCENARIO: SDN OVER FOG COMPUTING

Fog computing extends cloud computing at the edge of the network [1-5], it provides low latency, and improves quality-of-services (QoS) for streaming and real time applications. In this paper, we propose to combine the SDN virtualization technique and Ant colonies algorithms to

978-1-7281-6535-6/20/$31.00 ©2020 IEEE

enable flow allocation with respect to different QoS applications in the context of Fog computing.

The traditional way to configure the network is done on a device-by-device basis. This approach is time consuming and static, as the configurations don't change as network conditions change. Today's networks management needs more intelligence and user's aware protocols. The static traditional approach is obsolete. SDN opts for network programmability that leads to an increase in the potential performance of the network. Thus, we decide to use SDN to implement more intelligent and application-aware flow allocation algorithm.

Typically, the SDN controller (e.g NOX [6], ONIX [7], or Beacon [8]) observes and controls the entire network state from a central vantage point. The control of the forwarding plane is done through a vendor-agnostic south-bound interface, such as OpenFlow [9], which defines the low-level forwarding behavior of each forwarding element (switch, router, access point, or base station). For example, OpenFlow defines a rule for each flow; if a packet matches a rule, the corresponding actions are performed e.g. drop, forward or modify (Figure 1). We intend to exploit this feature to manage flow forwarding.

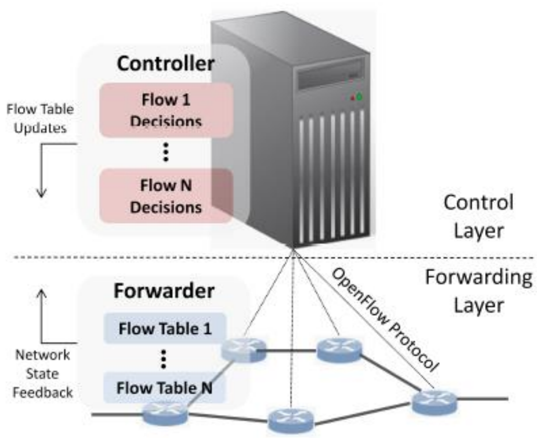

Figure 2: OpenFlow architecture

In our proposal the SDN controller will monitor classes of traffic to offers better QoS. This is possible thanks to the SDN northbound API. It makes the control information available to the applications that will be able to dynamically express their needs and changing the underlying network. This offers the ability to perform tasks such as forwarding packets over the least expensive path or changing the QoS settings based on the available bandwidth...

A key networking feature of our solution is its ability to discover multiple paths from the origin of the flow to its destination and to have measurements on multiple links qualities (delay, bandwidth, node's energy). This capability increases both the performance and scalability of the solution.
OpenFlow current implementations offer only two QoS features: queue and meter table [11-16]. However, none of them takes the multi-class applications and QoS metrics. Most of them deal with specific application type such as video streaming. Moreover, the previous solutions are not designed for the case of Fog computing scenario where devices are IoT nodes with limited resources and dynamic topology.

3. Proposed Flow Management Technique

Since finding the optimal route is a multi-objective problem with high computational complexity, it can be classified as a non-deterministic polynomial (NP) problem [17-21]. Ant colony optimization (ACO) algorithm can be appropriate heuristic to decrease this complexity.
Considering the diversity of networking application requirements (cf. table 1), and defining different well studied combination of QoS metrics, the SDN controller may discover several paths; each one is adequate to specific traffic class. This may be helpful to solve some network communication troubles like congestion, bottleneck, etc. This is the aim of our Algorithm which is a colored multi-flow algorithm (Figure 3).

Our Colored Ant algorithm is executed inside the centralized SDN controller. The global network view offered to the SDN controller helps as to efficiently execute our algorithm. The proposed method discovers all possible paths by forwarding ants and marks them with different backward ants, which have different colored pheromones depending on path's QoS parameters (Table 1).
We classify traffic into four classes (Table 1). We view these 4 classes as a starting point, with the actual choice of application classes being a topic for research. The first class corresponds to Real time application such as VoIP, the second one is streaming, the third class consists of interactive applications such as gaming, and the fourth one is best effort applications (mail, file transfer, etc.)

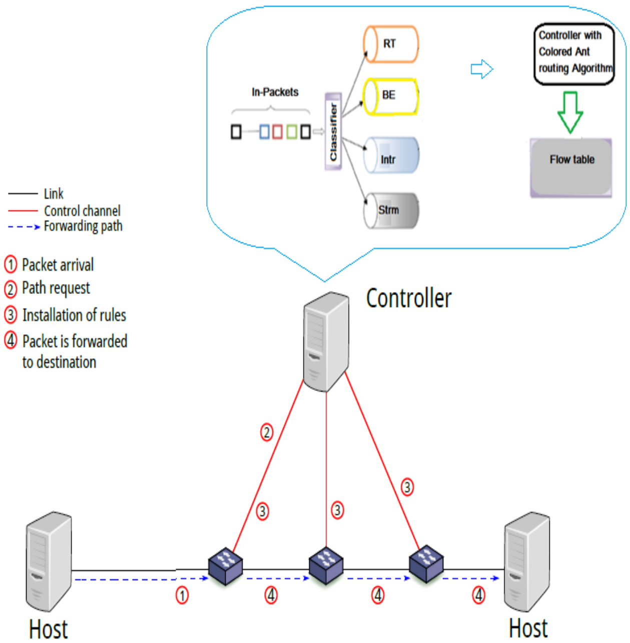

Figure 3: Packets treatments in Colored- Ant OpenFlow

The per-flow QoS metrics are:

- Node residual energy: Fog node can have limited battery because they belong to the IoT devices family. The node residual energy is important to decide about the node ability to survey, when using a Base Station this parameter is of course set to infinite.
- link bandwidth
- and end-to-end delay

As shown in the table below, each class requires specific QoS and is represented by a particular color. For example, traffic class 1 will be represented by the red color and requires low delay.

Our algorithm is run on the SDN controller on the top of the global network view (Figure 3). To find a path to a destination (d), a source node (s) broadcasts a request packet called Forward Ant (FA) to all its neighbors N(s). If a neighbor has a path to destination, it sends to the source node a reply packet called Backward Ant (BA) otherwise it updates the FA and forwards the packet to its neighbors till reaching the destination node. When the destination node receives the FA, it sends à BA to the source node containing the path grade (color) decided when pheromone value is calculated (cf. Eq.6). While travelling from source to destination, FA collects information about the path to determine the pheromone value which is useful to grade the path.

Table 1: Traffic Classification.

Class	1	2	3	4
Application	Real Time (VoIP)	Streamin g	Interactiv e gaming	Best effort
NodeEnerg y	☒	☒	☒	☒
Bandwidth		☒	☒	
Delay	☒		☒	
Color	R (Red)	B(Black)	B(Blue)	Y (Yellow)

Pheromone value is calculated based on the path residual energy (cf. Eq.1), end-to-end delay (cf. Eq.2) and capacity (cf. Eqs.3 and 4). The BA packet follows the FA reverse path, and every node in that path updates its pheromone table when receiving the BA. The path residual energy is the least node residual energy along the source "*s*" to destination "*d*"path "*P*".

$$E_{sd} = min_{i \in P}(E_i) \qquad (1)$$

The path end-to-end delay is:

$$D_{sd} = \sum_{i,j \in P} D_{ij} \qquad (2)$$

D_{ij} is the delay from node "i" to its neighbor "j", "i" and "j" belong to the path "P".

The ideal capacity C_{ij} of a channel with bandwidth B_{ij} is given by Shannon's Equation:

$$C_{ij} = B_{ij} \log(1 + SINR_{ij}) \qquad (3)$$

B_{ij} is the effectively available bandwidth on the "i-j" link. $SINR_{i,j}$ is the Signal to Interference plus Noise Ratio on the same link.

Total Path capacity:

$$C_{sd} = min_{i \in P}(C_{ij}) \qquad (4)$$

The pheromone value is calculated for the four traffic classes and the path will correspond to the color for which we obtained the best pheromone value.

$$ph_{sd}^{c} = \alpha_c . C_{sd} + {\beta_c}/{D_{sd}} + \gamma_c E_{sd} \qquad (5)$$

α_c, β_c and γ_c are constant, adequately fixed for each path color. If a node residual energy E_i is less than a specific threshold E_{th}, this node *i* should not be part of any path and is discarded from the forwarding process.

978-1-7281-6535-6/20/$31.00 ©2020 IEEE

4. IMPLEMENTATION AND PERFORMANCE RESULTS

We use OpenFlow to implement our routing protocol. Depending on the class of traffic the out-put path will differ: Once an OpenFlow switch receives a packet for which it does not have a matching entry (i.e. it has never seen that flow before), it sends this packet to the controller. It is now the controller's decision to handle this packet and to instruct the OpenFlow switches (by adding a flow entry) on what to do with similar packets coming in the future. The SDN controller runs the Co-Ant algorithm based on the value in the TOS field.
The topology testbed is described in Figure 5. Our evaluation platform is based on Open vSwitch to create software switches in the Linux kernel with support for OpenFlow. We use a dedicated machine for the controller and the benchmarker Cbench tool [22]; and all switches are connected to it. We use Floodlight [23] to implement our OpenFlow Controller. Four typical flow types (Interactive, video streaming, VoIP, and FTP for Best effort) were selected to evaluate the performance of proposed Colored-ant routing mechanism on OpenFlow.

Some performance metrics are used for evaluating our protocol: Throughput, PSNR for video flows (The generation of the PSNR metric was done by the PSNR Evalvid tool [24]), The packet delivery ratio and the average end-to-end packet delay.

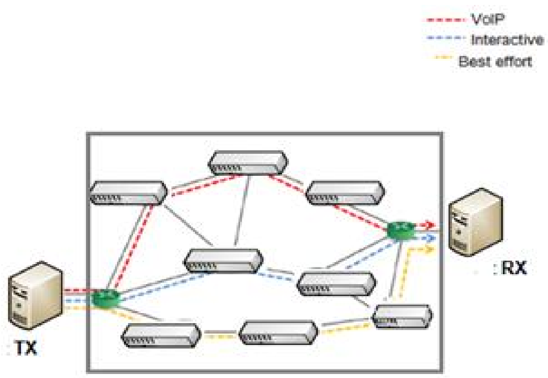

Figure 5: Testbed Architecture

The performance results are shown in following figures. In Figure 6, we can clearly see that the proposed Co-Ant algorithm offers different throughput depending on the type of application. In fact, the metric used to calculate best path is different depending on the applications classes thus the resulted route is distinct for each application type or color (see Figure 5). The same results can be concluded from Figures 7, 8 and 9 where Co-Ant outperforms traditional forwarding. Because as explained in the previous section our algorithm tries to adapt to applications requirement for path selection; each flow type will be routed differently. The algorithm tries to find the best path in terms of shortest delay for real time traffic, best bandwidth for Streaming and with good delay vs. throughput compromise for interactive applications. In figure 7, we have shown that the PSNR in our algorithm is better than classical methods because of this QoS-aware characteristic.

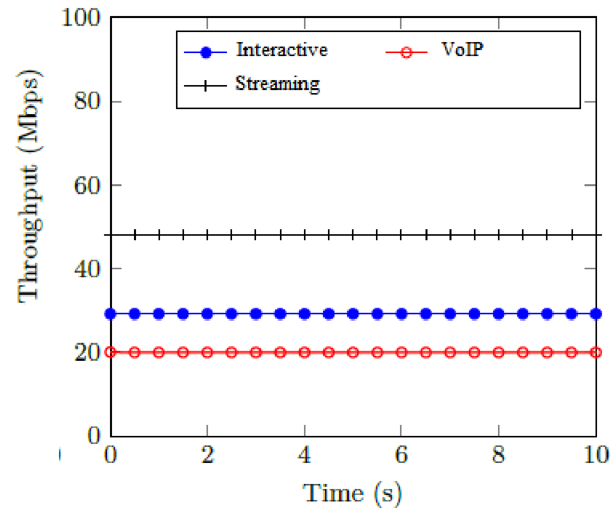

Figure 6: Throughput under different traffic types

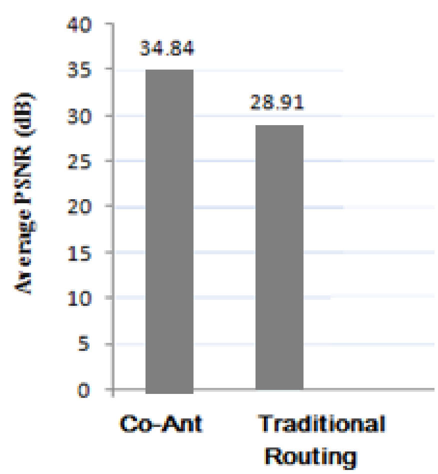

Figure 7: PSNR for Co-Ant versus Traditional routing

5. CONCLUSION

Quality of Service (QoS) support for Fog networks is an exigent task due to its dynamic topology and limited resource. We have proposed in this paper a new flow management framework over SDN Fog networks. Our algorithm starts with a traffic classification inside the SDN controller, in order to differentiate network data flows, and to treat each flow with its proper QoS metrics. Then, we use ant algorithms for path selection. We propose a concept with colored ant pheromones, where a color corresponds to a particular class of traffic. Thus, the network will treat the packets of an application according to the specific application requirements delay and bandwidth.

Implementation performance result shows the efficiency of our method.

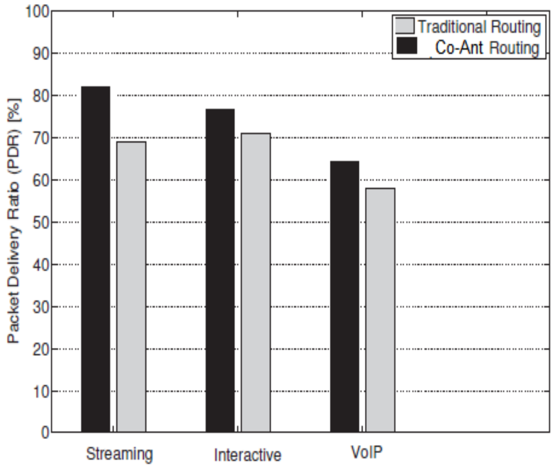

Figure 8: PDR for different applications types

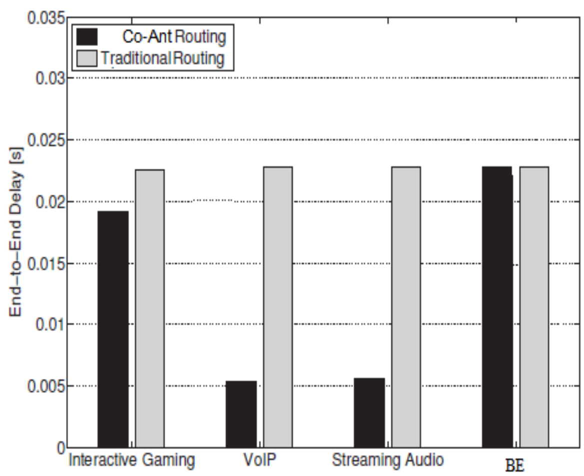

Figure 9: End-to-end delay for different applications classes

REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in Proceedings of ACM MCC, 2012, pp. 13–16.

[2] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H. I. Yang, "The Case for Cyber Foraging," in Proceedings of ACM SIGOPS, 2002, pp. 87–92.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[4] E. T. S. Institute, "Mobile-edge computing-introductory technical white paper," 2014.

[5] G. M. Relations, "Greyhound Launches "BLUE", an Exclusive Wi-Fi Enabled Onboard Entertainment System," https://www.greyhound.com/, Accessed on Dec. 2016.

[6] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, and N. McKeown.Nox: Towards an operating system for networks. In ACM SIGCOMM CCR: Editorial note, July 2008

[7] T. Koponen, M. Casado, N. Gude, J. Stribling, P. L., M. Zhu, R. Ramanathan, Y. Iwata, H. Inouye, T. Hama, and S. Shenker. Onix: A distributed control platform for large-scale production networks. In Operating Systems Design and Implementation.USENIX, 2010.

[8] Beacon: a Java-based OpenFlow Control Platform. http://www.beaconcontroller.net/

[9] Nick McKeown; et al. "OpenFlow: Enabling innovation in campus networks" ACM Communications Review. Retrieved 2009-11-02.

[10] F. Wamser, D. Hock, M. Seufert, B. Staehle, R. Pries, and P. Tran-Gia, "Using Buffered Playtime for QoE-Oriented Resource Management of YouTube Video Streaming," Transactions on Emerging Telecommunications Technologies, vol. 24, Apr. 2013.

[11] M. Jarschel, F. Wamser, T. H¨ohn, T. Zinner, and P. Tran-Gia, "SDN based Application-Aware Networking on the Example of YouTube Video Streaming," in 2nd European Workshop on Software Defined Networks (EWSDN 2013), (Berlin, Germany), Oct. 2013.

[12] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al., "B4: Experience with a globally-deployed software defined wan," in Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pp. 3–14, ACM, 2013.

[13] S. Das, Y. Yiakoumis, G. Parulkar, N. McKeown, P. Singh, D. Getachew, and P. D. Desai, "Application-aware aggregation and traffic engineering in a converged packet-circuit network," in Optical Fiber Communication Conference and Exposition (OFC/NFOEC), pp. 1–3, IEEE, 2011.

[14] K. Jeong, J. Kim, and Y.-T. Kim, "Qos-aware network operating system for software defined networking with generalized openflows," in Network Operations and Management Symposium (NOMS), 2012 IEEE, pp. 1167–1174, IEEE, 2012.

[15] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An optimization framework for qos-enabled adaptive video streaming over openflow networks," Multimedia, IEEE Transactions on, vol. 15, pp. 710–715, 2013.

[16] M. Jarschel and R. Pries, "An OpenFlow-Based Energy-Efficient Data Center Approach," ACM SIGCOMM, 2012.

[17] Rout A, Sethi S, Mishra D. "Optimized ant based routing protocol for MANET". ICCCS'11; 2011. p. 21–25.

[18] Gupta A, Sadawarti H, Verma A. "MANET routing protocols based on ant colony optimization". J Model Optim. 2012; 2(1):42–49.

[19] Sabari A, Duraiswamy K. "Multiple constraints for ant based multicast routing in mobile Ad Hoc networks", Journal Computer Scencei. 2009; 5(12):1020–7.

[20] Yoshikawa M, Otani K. "Ant colony optimization routing algorithm with tabu search". International Multi Conference Engineers Computer Science; 2010 Mar. p. 17–20.

[21] Dominguez-Medina C, Cruz-Cortes N. "Energy-efficient and location-aware ant colony based routing algorithms for wireless sensor networks". GECCO; 2011. p. 117–24.

[22] Cbench. http://docs.projectoodlight.org/display/oodlightcontroller/Cbench

[23] FLOODLIGHT. Open sdn controller. http://floodlight.openflowhub.org/

[24] Evalvidtool ; http://www.tkn.tu-berlin.de/research/evalvid/