Fog Computing as a Resource-Aware Enhancement for Vicinal Mobile Mesh Social Networking

Chii Chang[1], Mohan Liyanage[2], Sander Soo[3] and Satish Narayana Srirama[4]

Mobile and Cloud Computing Laboratory, Institute of Computer Science, University of Tartu, Estonia
[1]chii.chang@acm.org, [2]liyanage@ut.ee, [3]sander.soo@ut.ee, [4]srirama@ut.ee

***Abstract*—Mobile Mesh Social Network (MMSN) represents an environment where the mobile device users are capable of performing various virtual social network activities such as sharing information, forming social groups, text messaging when they encounter each other in the physical vicinity within the wireless network range. Moreover, the characteristics of MMSN such as the Internetless activities and Wireless Mesh Network (WMN)-based connectivity provides various potentials including but not limited to business opportunities, scalable crowdsourcing or crowdsensing deployment, edge computing and so on. Although there exist a fair number of software platforms that help developers to implement MMSN, they still cannot fully overcome the limitation derived from the hardware resource constraint nature of the participative mobile devices. In order to enhance the MMSN in terms of cost efficiency, we introduce Fog Social Network (FSN) model, which utilises the computing and networking resources in users' close vicinity to improve the overall efficiency of MMSN. Further, the proposed FSN framework consists of an adaptive resource-aware cost-performance index (CPI) scheme, which performs dynamic approach selection autonomously at runtime to choose the most efficient route for the delivery of the messages for MMSN activities. With this intention, we have implemented and validated a proof-of-concept prototype.**

***Index Terms*—Fog computing, edge computing, mobile computing, mesh social network, social network application, cost-performance index.**

I. Introduction

For past few years, social applications have rapidly grown to one of the most popular mobile applications [1]–[3] in which accessing the classic centralised distant data centre-based social network services (CSNS; e.g. Facebook[1], Twitter[2], Foursquare[3] etc.) became a common daily activity of many mobile device users. Meanwhile, a new breed of Mobile Social Network (MSN) application is getting attention by the society. In contrast, such an MSN application does not require the full-time Internet connection with the distant data centre (DDC) for performing social activities (e.g., finding friends, forming or joining social groups, chatting, sharing files etc.). Specifically, the Internetless MSN applications utilise the inbuilt wireless communication mechanisms of mobile devices (e.g., Wi-Fi and Bluetooth) to enable the data routing among the participative devices in the Wireless Mesh Network (WMN) [4] topology. In general, the WMN-based MSN enables the spatio-temporal social activities among the participants, which potentially brings the opportunities to the MSN users to participate in the real world events. Here, we use the term - Mobile Mesh Social Network (MMSN) to describe such MSN environment.

MMSN, such as Open Garden's FireChat[4] or Serval Project's Serval Chat[5], are useful in the situation when the regular cellular network is unstable, unavailable (e.g. in disaster recovery scenarios) or simply when the MSN users want to save their mobile Internet bandwidth in the area where the free access points are not available. To enumerate, existing open frameworks that can realise MMSN include but not limit to Apple's Multipeer Connectivity[6], AllSeen Alliance's Alljoyn[7], Open Connectivity Foundation (OCF)'s IoTivity[8] and so on.

In MMSN, mobile devices can act as both regular peers and routers. As a router node, the mobile device will try to deliver the data to the destination node for its nearby peers based on the routing information shared among the peers. Ideally, the WMN topology used by MMSN is flexible and scalable since it does not rely on particular peers to be the cluster heads (super peers). However, it may face a number of limitations as illustrated in Figure 1:

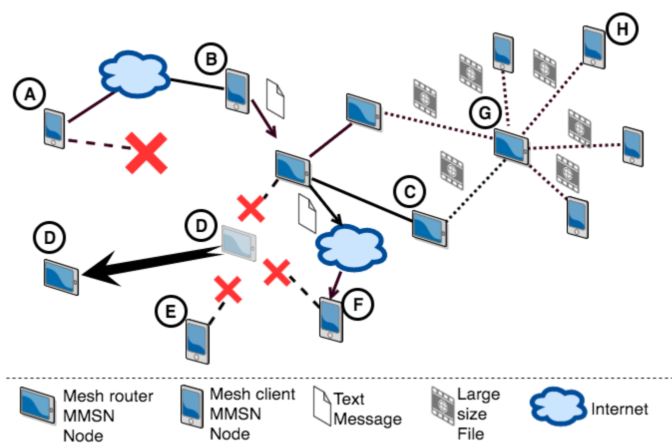

Figure 1: MMSN example.

[1]https://www.facebook.com
[2]https://www.twitter.com
[3]https://foursquare.com/

[4]http://www.opengarden.com/firechat.html
[5]http://www.servalproject.org
[6]https://developer.apple.com/reference/multipeerconnectivity
[7]https://allseenalliance.org/framework
[8]https://www.iotivity.org

- *Unstable connectivity*. For example in Figure 1, suppose Node-B intends to send a message to Node-F. Initially, the only connected node to Node-F was Node-D. Since Node-D has moved out from the network, Node-B's message cannot reach Node-F unless the intermediated node utilises Internet connection (if Node-F is connected to the Internet). Alternatively, the intermediate node has to wait until it encounters another node that can route the message to Node-F.
- *Bottleneck issue*. In MMSN, there is a chance that a node becomes the only router node for a group of participants. For example in Figure 1, Node-G has faced such a situation. Consequently, due to the heavy traffic occurred at Node-G, the transmission at that edge became extremely slow. Further, it may discourage Node-G to maintain its connection and eventually disconnect from the network because it has consumed too much of its own hardware resources and energy (battery).
- *Lack of routing path*. For example, Node-A does not have any connected path. Hence, the only way for Node-A to participate in the network is to utilise Internet. Similarly, Node-E is facing the same issue as Node-D has moved out from the network.

One promising solution to overcome the limitation of MMSN is utilising Fog computing model. Fog Computing (or Fog) [5] represents a Utility Cloud computing environment that is available in the users' vicinity. In general, such computing resources, which are known as Fog Drivers or Fog nodes, are industrial network routers or switches (e.g. Cisco 829 Industrial Integrated Services Routers[9]) that are employed as wireless Internet access points in various areas such as residential or business buildings, factory plants, shopping centres, urban areas and so forth. Fog nodes embed virtualisation technologies (e.g. Virtual Machines) or containerisation technologies (e.g. Docker containers[10]) that allow users to deploy software on them. Compared to the traditional distant Cloud computing model, which requires sending all the data files to the DDC for the processing, Fog can provide much higher agility.

Fog computing model can enhance the efficiency and the Quality of Experience (QoE) of MMSN in terms of reliability and cost efficiency. Hence, in this paper, we introduce Fog Social Network (FSN), which is an enhancement of MMSN. Further, we also have considered that if the entire MMSN activities are fully relying on Fog, it may not be adaptive to different situations due to the dynamic nature of mobile networks. Hence, we introduce an adaptive resource-aware Cost-Performance Index (CPI) scheme, which helps the system to choose the best approach for delivering the messages in MMSN based on the runtime context factors.

This paper is organised as follows. Section II summarises the related works. Section III describes the details of the proposed FSN architecture and framework. It is followed by the prototype implementation and evaluation in Section IV. The paper is concluded in Section V together with the future research plan.

II. Related Works

The proposed framework relates to two research fields—edge computing and opportunistic social network. Therefore, this section summaries the related works and justify the differences between the proposed framework of this paper and the related works.

Cloudlet [6] is an open source edge computing solution[11] that has similar concept as Fog. In contrast, the initial concept of Cloudlet is based on providing Virtual Machine (VM)-enabled service from the wireless network communicable computing machines of local business (e.g. cafeteria) to the users in vicinity. In general, it is expected that the Cloudlet machines are fairly powerful in terms of computing and networking. Hence, they can provide on-demand computational process offloading services (e.g., real-time face recognition [7]).

A number of works utilised Cloudlet to enhance multimedia content delivering or streaming [8], [9]. Further, similar strategies have also been applied in Fog for online game streaming [10], [11]. These works are focusing on streaming the content between the distant server and the edge network mobile devices. On the other hand, the FSN introduced in this paper does not involve distant content streaming for the social activities in the edge network. Specifically, the main focus of FSN is for enhancing the social activities in the vicinal MMSN.

Opportunistic mobile social network (OppMSN) [12] represents the vicinal MSN established by utilising Delay-Tolerant Networks (DTNs) among the participative mobile devices. The social activities among the participants rely on the mobile ad hoc routing mechanisms. Initially, OppMSN is triggered by academic research projects such as [12]. The discontinued Haggle[12] mobile app was a representative OppMSN enabler.

OppMSN-based research projects [13]–[15] usually focus on how to enable the efficient data routing in the DTNs. Expressly, works such as MobiClique [13], Peoplerank [14] were more focused on improving the opportunity of the social encounter among the participants. Further, they did not consider utilising the infrastructures of Cloudlet or Fog to improve the efficiency. Different from their works, the proposed FSN tends to enhance the efficiency of MMSN towards improving the overall QoE.

Mobile Social Network in Proximity (MSNP)-based approaches [16], [17] focus on open standard-based interoperability among participants. Similar to OppMSN, MSNP frameworks are also based on ad hoc topology. Instead of using vicinal resources (e.g. Cloudlet), MSNP utilises distant Cloud to extend its computational resources, which is the

[9]http://www.cisco.com/c/en/us/products/routers/829-industrial-router/index.html

[10]https://www.docker.com/

[11]http://openedgecomputing.org

[12]https://play.google.com/store/apps/details?id=org.haggle.kernel

major difference between MSNP and the proposed FSN in this paper.

Overall, past frameworks have not considered the cost-performance of the system while utilising the concept of Fog Computing, which is one of the core contributions of this paper.

III. System Design

A. Overview of Environment

Figure 2 illustrates a FSN environment. FSN is a multi-layered system that consists of Cloud and Edge network.

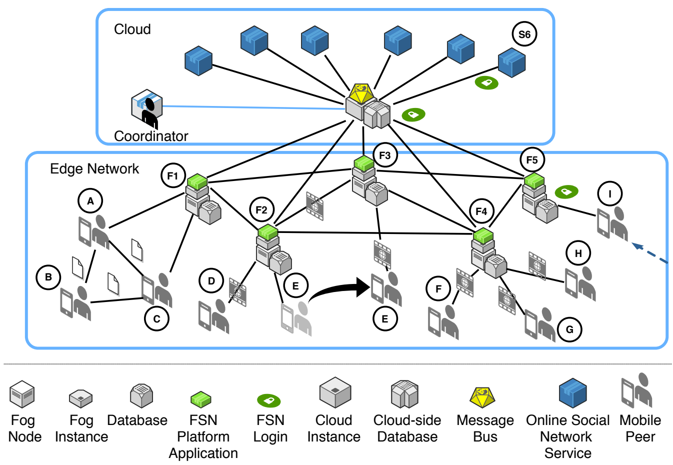

Figure 2: Fog Social Network environment.

1) Cloud: The upper layer Cloud consists of following main elements:

- *Online Social Network Service (OSNS)* providers (e.g. Facebook, Twitter, Foursquare) are the sponsors of FSN. They handle the end users mobile Internet access bandwidth cost for their bootstrapping to FSN. The bootstrapping phase will be described later. This assumption is based on the incentive that SNS provider is willing to offer the free Internet for limited accessibility [18].
- *Message Bus (MB)* is an elastic Cloud service for routing messages between OSNS and Fog nodes. It is managed by the Coordinator. MB only routes data or message exclusively to the corresponding OSNS servers. For example, a Facebook User-A who signed in to FSN can share a content to another Facebook User-B via FSN. Since the content will be sent through a Fog node, Facebook can subscribe for such content via MB in order to synchronise the content to User-A and User-B's Facebook accounts. On the other hand, since the content was from Facebook user, Twitter server will not be able to access or subscribe to it.
- *Coordinator* is a trustworthy party who has been delegated the responsibility to manage the MB for the OSNS providers. The coordinator can be an individual organisation established by the OSNS providers or it can be the Cloud service provider itself. Note that although the data is routed in MB, the Coordinator does not store any user data in its database. The Coordinator only maintains the up-to-date FSN deployment package and the connection states of FSN users.

2) Edge Network: The lower layer Edge Network consists of Fog nodes (e.g. F1, F2, F3, F4 and F5 in Figure 2) and the participative mobile peers (users) of FSN.

The edge network of FSN consists of the following characteristics:

- *Retain social activities without the Internet*. For example, Node-A, Node-B and Node-C can still communicate in the P2P mode.
- *Rapid file sharing regardless of mobility issues*. For example, Node-D intends to share a large file to Node-E. However, Node-E has moved away from the signal rage of Node-D. In this case, Node-D can forward the file to Fog node F2. Afterward, Node-E will receive the file when it connects with another Fog node (e.g. F3). It is achieved based on utilising the MB in Cloud that helps to maintain the peers' connection states. In other words, F2 will try to track which Fog node coverage Node- E is in by utilising the MB, then F2 will forward the file to that node either by direct connection (e.g. they are in close proximity or they have connected to the Internet via the same network provider) or via the MB.
- *Large size file sharing with low latency*. For example, Node-F can share a large size media file to Node-G and Node-H via F4.
- *Secure sign-in*. A newly joined peer (e.g. Node-I) who has OSNS account can establish its connection to FSN with secure sign-in. The details will be described in next section.
- *FSN Platform Application package* is a software that can be deployed on a Fog node. It helps users to connect to the FSN. Fundamentally, it works same as a regular MMSN router node. Additionally, it has connection to the Cloud-side MB to assist FSN users to sign-in to the FSN without using their own mobile Internet usage. In summary, FSN platform application provides the following basic mechanisms:
  - Assist FSN participants to sign-in.
  - Provide subscription for OSNS to their users' content.
  - Assist FSN participants in discovery of their vicinity.
  - Assist FSN participants in interacting with their surrounding devices including FSN user devices and Internet of Things (IoT) devices via standard protocols such as Bluetooth.
  - Temporary storage of data and delivering files via Fog Director cluster to receivers.
- *General Access*. General desktop computer users can also use FSN via Fog node or even directly subscribe to the MB.

3) Sign-in: In MMSN such as FireChat, users need to access Internet in order to sign-in. In FSN, it is handled by the

Fog node. For example in Figure 2, Node-I can send a sign-in request to F5. F5 will then forward the request to MB, who will forward the request to corresponding OSNS server based on the account used by the initial requester. Suppose Node-I is using the account of OSNS provider-S6. Hence, MB forwards the request to S6. If the account of Node-I is valid, S6 will confirm the validity of the request and send the request to the Coordinator to help Node-I establishing connection to FSN via F5.

4) Deployment: If the FSN user moves to an environment where the FSN is not yet been deployed, the user can send a FSN deployment request to the Fog node in vicinity. For example in Figure 3, the request message will be forwarded

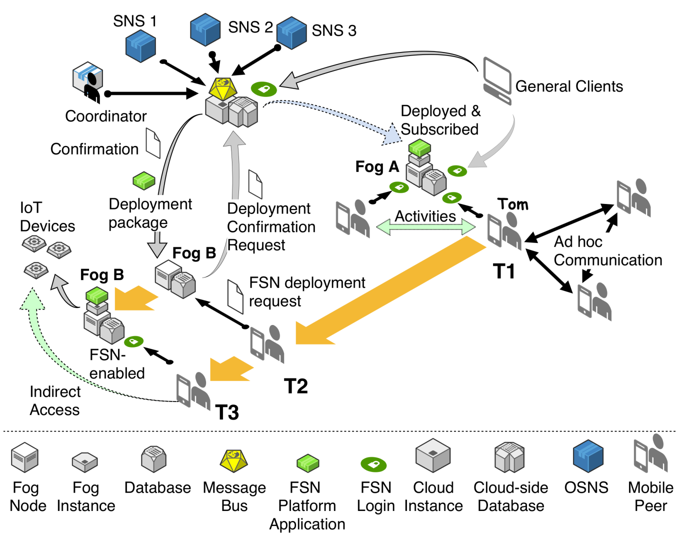

Figure 3: Fog Social Network deployment example.

to the MB Coordinator. The Coordinator will confirm the user account with OSNS providers. If the account is valid, the Coordinator will send the FSN platform deployment package to the Fog node and deploy the FSN. Afterward, the requester will receive an acknowledgment and it will be connected to FSN. On the other hand, if the Fog node has previously deployed FSN platform and the image is still available in its local storage, then it can deploy the FSN platform without needing the Coordinator to send the package to it again.

MMSN applications in FSN can use the Fog nodes not only for transmitting messages, but they can also use Fog nodes for further needs such as retrieving information from environmental public accessible IoT devices. We consider this topic as one of our future research directions.

5) Adaptation: Social activities will be based on both MMSN and FSN. For simple activities such as text messaging or small file sharing between participants, the FSN application will calculate the efficiency. It will use direct P2P when:

- **Reliability**. Direct P2P signal strength is high (connection is good) [19] and the mobility of the participants will not affect the complement of their data transmission. This can be measured based on the accelerometer of their mobile devices and the latency testing based on the data transmission measurement.
- **Agility**. Direct P2P transmission will be faster than utilising the Fog. It needs to be calculated based on comparing the two approaches (the Fog-based and the direct P2P-based).
- **Cost**. Direct P2P transmission does not require more cost than utilising Fog. The cost can be for example, the energy consumption. it involves the protocols and data size they are using. For example, Fog nodes are commonly Wi-Fi-based (based on Cisco's Fog Computing approach). If the data transmission required between two participants can be done rapidly using Bluetooth, then the cost-performance value of the direct P2P-based approach can be higher. We will discuss the details of the cost-performance computing in next section.

6) Constraint: FSN may not provide regular Web browsing to its users, neither the content hosts in their OSNS servers. Users who intend to use the Fog node for additional Internet accessing may need to handle the cost by themselves, unless either Fog provider or OSNS provider is willing to cover the cost.

B. Cost-Performance Index Scheme

As mentioned previously, if the entire system is fully relying on one approach, it may not be adaptive due to the dynamic nature of mobile network environment. For example, in some cases, utilising the classic mobile peer-to-peer (MP2P) communication can be more efficient than utilising Fog when the participants are sending small size messages. On the other hand, Fog can be more efficient than MP2P when the participants intend to share large size media files. Furthermore, in the case where the participant is situated in a high density area where Wi-Fi and Bluetooth signal interaction is occurring frequently, it can make the short-range-based communication extremely slow, the Cloud-based approach may become a better option. With this in mind, the proposed FSN framework also consists an adaptive resource-aware Cost-Performance Index (CPI) scheme, which can autonomously decide which is the best route to deliver the messages for MMSN nodes based on the runtime context factors. This section provides the details of the CPI scheme.

Let $O = \{o_i : 0 \leq i \leq \mathbb{N}\}$ be a set of routing options (e.g. via mobile P2P, via Fog, via Cloud etc.).

Cost of an option $o_x \in O$ (denoted by $cost_x$) is computed by:

$$cost_x = \sum_{j=0}^{|C|} \left(\frac{v_j^x}{\sum_{i=0}^{|O|} v_j^i} \times \left(1 - \sqrt{1 - (w_j)^2}\right) \right) \quad (1)$$

where:

- $C = \{c_j : 0 \leq j \leq \mathbb{N}\}$ is a set of cost elements considered by the sender.
- v_j^x denotes the value of the cost element c_j of option o_x.
- v_j^i denotes the value of the cost element c_j of option o_i.

– w_j is the weight of the cost element c_j.

Figure 4 illustrates the generated value from $1-\sqrt{1-(w_j)^2}$ where $\gamma = 1 - \sqrt{1-(w_j)^2}$, which defines the stimulus of weight to the cost.

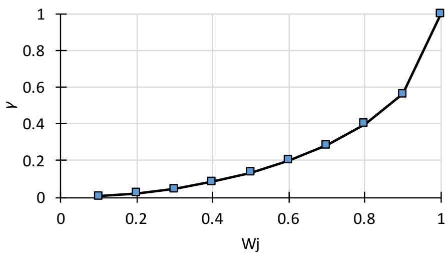

Figure 4: The weight calculation.

The weight w is a dynamic value influenced by the hardware usage. The higher the current hardware usage is, the higher the weight can be.

The weight of a cost element $c_y \in C$ (denoted by w_y) is computed as below:

$$w_y = \frac{(\mathcal{U}_y)^{con}}{\sum_{j=0}^{|C|}(\mathcal{U}_j)^{con}} \tag{2}$$

where:

- $\mathcal{U}$ is the current average resource usage value. It is between 1 to 100 (representing 1% to 100% of the resource usage). For example, "CPU = 67" denotes CPU currently has 67% usage of its available threshold; "battery = 30" denotes battery has been used up to 30% (i.e. 70% left) of its available threshold.
- con is the value that can influence the constraint of the weight and $con \geq 1$. By default, con is 2, in which the variety of the weight increases explicitly high when it is closer to the threshold.

Note that the available threshold does not necessary equal to the total availability of the hardware resource. User can decide for example; the total available threshold of the energy is 70% of the hardware battery. The detail of how user can define the threshold will involve the user interface design, which is out of the scope of this paper.

The performance of each option is measured based on their timespan. Let $\mathscr{T} = \{\tau_i : 0 \leq i \leq \mathbb{N}\}$ be a set of timespan values of the options, where τ_i denotes the timespan of o_i. Let τ_x be the normalised timespan value of option $o_x \in O$, which is computed by:

$$\tau_x = \frac{\sum_{l=0}^{|E^x|}\frac{S}{\mathcal{M}_l - ED_l}}{\sum_{i=0}^{|\mathscr{T}|}\tau_i} \tag{3}$$

where:

- $E^x = \{e_l : 0 \leq l \leq \mathbb{N}\}$ is a set of node pairs for delivering a message from sender to the receiver involved in option o_x.
- S is the size of the data that needs to be sent from the sender to the receiver(s). It is formatted in Megabit (Mb).
- $\mathcal{M}_l$ is the default transmission speed of pair e_l in Megabit per second (Mbps), which depends on the protocol used by the involved nodes.
- ED is the sum of the delays of all the environmental delay factors $\mathfrak{D}$. $\mathfrak{D} = \{D_m : 0 \leq m \leq \mathbb{N}\}$, which are dynamic delay value influenced by the environmental network conditions. For example, in a high density environment, a large number of wireless network transmissions are happening, the ED can be high.

Let τ' be the normalised value of τ. For option $o_x \in O$, its normalised τ value (denoted by τ'_x) is computed by:

$$\tau'_x = \frac{\tau_x}{\sum_{i \in |O|}\tau_i} \tag{4}$$

Afterward, the performance rank of an option $o_x \in O$ (denoted by $perf_x$) is computed by:

$$perf_x = 1 - \tau'_x \tag{5}$$

where the lower timespan the option involves, the higher performance rank it has.

Finally, the CPI score of option $o_x \in O$ (denoted by CPI_x) is computed by:

$$CPI_x = \frac{perf_x}{cost_x} \tag{6}$$

IV. Prototype

The goal of the evaluation is twofold. First, we aim to evaluate the cost efficiency of utilising Fog-based MMSN when compared to the classic MMSN and the Cloud-based social content delivery. Second, we aim to evaluate the proposed adaptive resource-aware CPI scheme with the collected data from real world experiments as the input values of the scheme.

The prototype is based on simulating a FSN environment where the communication of the participative entities is based on MQ Telemetry Transport (MQTT; ISO/IEC PRF 20922), which is a publish/subscribe-based protocol. Publish/subscribe protocol is the core enabler of many social network applications such as Facebook Messenger, Jabber, Skype etc. Although Extensible Messaging and Presence Protocol (XMPP; IETF RFC 3920) may be more popular than MQTT since it has existed for many years, we choose MQTT because it is lightweight and more feasible to operate in the resource constraint environments.

A. Implementation

The main elements in the prototype have been implemented as below:

- MMSN participants—were simulated in LG G4C and LG Spring smartphones, which are operated in Android OS 6.0. For the classic MMSN use case, one of the smartphone was operating Mosquitto[13] MQTT broker/server

[13]https://mosquitto.org/

and client at the same time. The reset smartphones were MQTT clients.

- FSN node—was simulated by a HP laptop computer, which has installed an Apache Tomcat server with MQTT broker/server Web application.
- OSNS server—was simulated by an Amazon cloud instance.
- Networking—The communication between MMSN participants and FSN node was based on Wi-Fi (802.11n). In the case of mobile to cloud communication, which simulated the traditional OSN activities, the communication is performed using Estonian Tele2's 4G LTE mobile Internet connection, which has in average 50 to 70 Mbps for download speed and in average 30 to 40 Mbps for upload speed.

B. *Use Cases*

The evaluation aims to compare the cost and performance among the three models: OSNS, classic MMSN and FSN. In order to compare them, we used the following main use cases.

1) User-1, who has subscribed to the social network topic-A, intends to publish a multimedia content to the topic. Meanwhile, 3 other users have subscribed to the topic.
2) User-1, who has subscribed to three social network topics, is receiving three multimedia content published by the other 3 users to the three topics.

For each use case, we first compare the cost and performance when each of them is performed by OSNS, classic MMSN or FSN model. Afterwards, we applied the data collected from the real world testing to the proposed CPI scheme to validate how well the system can autonomously adjust the decision making among the three options (i.e. publish the content via OSNS, classic MMSN or FSN)

C. *Evaluation Results*

1) Performance Comparison: Performance is based on the timespan measurement of each option. We tested them by using different sizes of data as the published content.

Figure 5 shows the measurement results in comparison of the three options in the Case 1, known as publishing a content in different data size to three topics. The timespan represents the time from publishing the content to the time that all the subscribers have received the content and have acknowledge the MQTT broker node. As the result shows, Fog delivered the best performance, followed by MP2P. Cloud has the lowest performance.

Figure 6 illustrates the Case 2 in which a participant has subscribed to three topics and it has received the three new contents published from the other three participants. In this case, MP2P approach results the better performance than Fog. Cloud is still having the lowest performance.

2) Cost Comparison: In this evaluation, we consider CPU usage and energy consumption as the cost elements. The CPU usage measurement is based on recording the CPU usage while performing the activities. An example of The raw data of the CPU usage is shown in Figure 7.

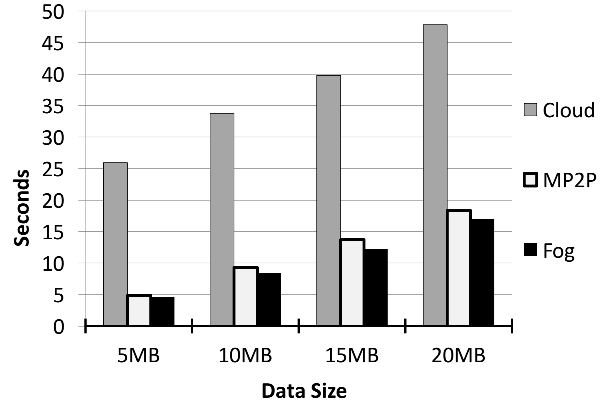

Figure 5: Timespan comparison for Case 1.

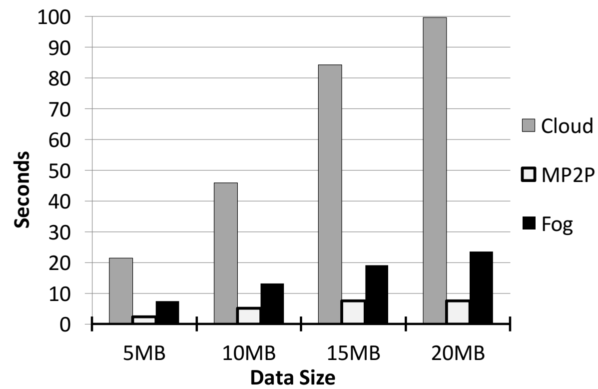

Figure 6: Timespan comparison for Case 2.

The total CPU usage is based on the usage value times the total number of the timespan. For example, while performing a content publishing task, if CPU load is 60% and it remains 60% for 5 timestamps, the total cost of the CPU will be considered as $60 \times 5 = 300$ raw cost value. Figure 8 illustrates the testing result for the CPU usage in Case 1 (publishing content to three topics).

As the result shows, Cloud-based approach consumes the highest CPU usage. Fog-based approach consumes the smallest CPU usage.

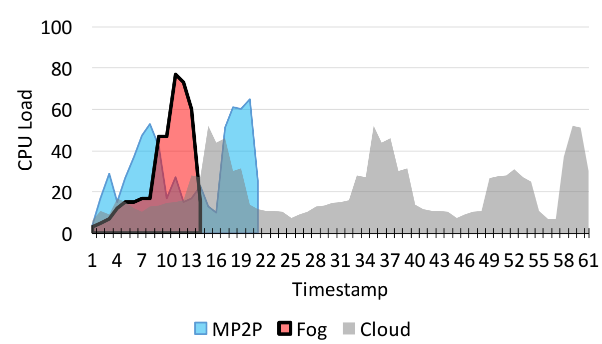

Figure 7: Raw CPU usage log example.

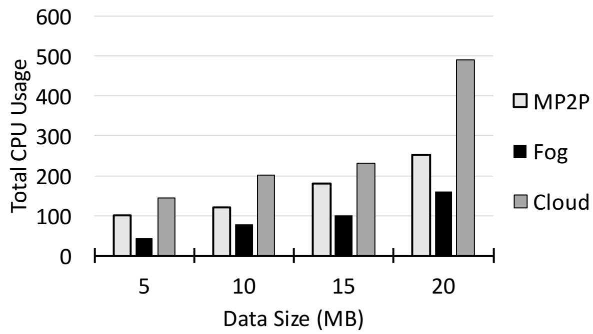

Figure 8: Total CPU consumption comparison for Case 1.

Figure 9 illustrates the CPU consumption measurement and comparison of the three options in Case 2 (receiving content from three subscribed topics). We have received a similar result as the previous testing in which the Cloud-based approach is still the one which consumes the highest CPU usage, followed by MP2P. Fog-based approach is still resulting as the approach that requires the lowest CPU usage.

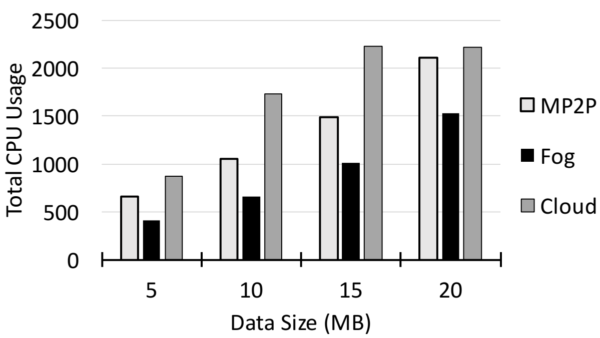

Figure 9: Total CPU consumption comparison for Case 2.

Next, we have tested the energy consumption using the same setting. Figure 10 shows the result of the energy consumption of the Case 1 (publishing content). As the result shows, Fog-based approach consumes the lowest energy. Cloud consumes the highest energy usage.

Figure 11 illustrates the energy consumption testing result for Case 2 (receiving content from three subscribed topics). Different to the previous testing result, in Case 2, MP2P-based approach consumed the lowest energy, followed by Fog. Cloud-based approach still consumed the highest energy.

This result indicates that Fog-based approach is not always the best option in some cases. Therefore, utilising CPI scheme can improve the cost-performance efficiency when we consider about different situations.

3) CPI Scheme: The aim of the evaluation of CPI scheme is to validate that the scheme is capable of ranking the options in order to identify which option is more efficient to perform the process.

This testing is based on the setting of both Case 1 and 2 with 10MB as the data size. We considered CPU and energy consumption as the cost elements of the options. Further,

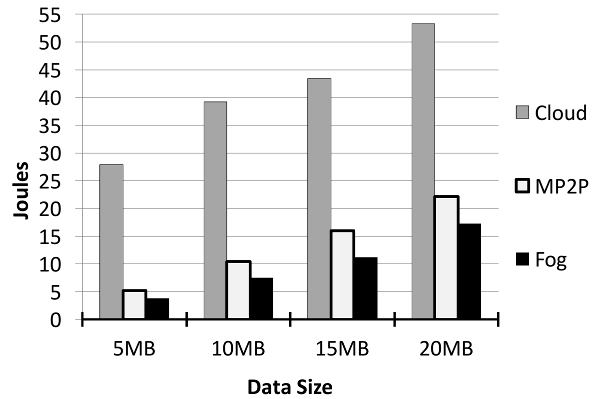

Figure 10: Energy consumption comparison for Case 1.

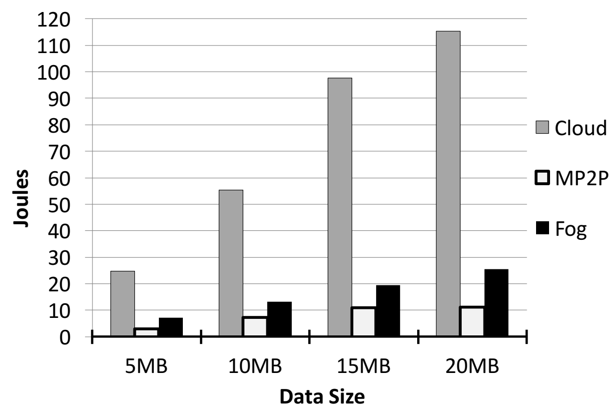

Figure 11: Energy consumption comparison for Case 2.

we consider the timespan as the performance. Based on the proposed scheme, we received the results shown in Figure 12.

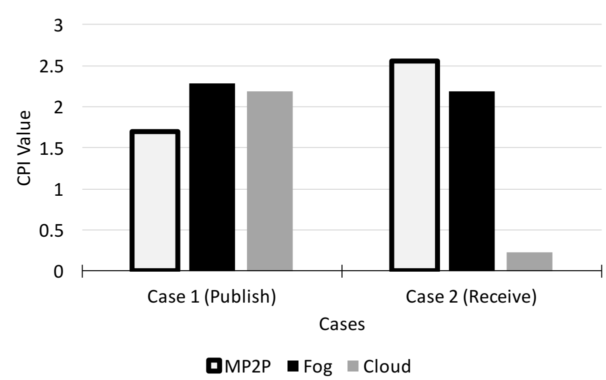

Figure 12: CPI comparison between Case 1 and Case 2 with 10MB content data size.

As the result shows, when the participant intends to publish a content to a topic which has been subscribed by three other participants, Fog-based approach can provide the best CPI value. Hence, the content publisher node has chosen to publish the content via Fog. One the other hand, in the case of a participant who has subscribed to three topics and intends to receive the new content from the topics, MP2P approach provides the best CPI value. Hence, the participant utilised the MP2P-based connection to subscribe for the topics.

Note that, these CPI computation measurements can be influenced by other factors such as the signal strength or RAM usage. We consider to include more factors in our future work. At this stage, the result has provided a proof-of-concept that the proposed CPI scheme is capable of computing the efficiency of different options based on the cost and performance parameters.

V. Conclusion and Future Work

In this paper, we introduced Fog Social Network, a platform to enhance the reliability and cost efficiency of Mobile Mesh Social Network. Considering the performance and context factors of runtime interaction among the MMSN participants, we further introduced the resource-aware cost-performance index (CPI) scheme. The scheme takes runtime environmental context factors to dynamically make decision about how the interaction should be performed (i.e. using mobile P2P, Fog or Cloud). The prototype experimental results have shown that utilising Fog can improve the cost efficiency of MMSN. Further, the adaptive CPI scheme can optimise the decision making in choosing the best approach for the data transmission between MMSN participants.

In the future, we plan to address the following aspects:

- Validate and compare between the containerisation (e.g., Docker containers)-based Fog implementation and Virtual Machine (VM; e.g. VirtualBox)-based Fog implementation of our system to analysis the efficiency of different FSN deployment approaches. Moreover, we also plan to compare the current approach to the Mist computing [20] model in terms of performance and cost efficiency.
- Integrate Fog-centric Business Process Management System (BPMS) [21] with FSN to improve the management of the overall FSN activities. Further, the integrated system can interact with the IoT devices in the surrounding area using Device-to-Device (D2D)-based BPMS [22] in order to improve the context-awareness of the FSN.

Acknowledgment

This research is supported by Estonian Science Foundation grant PUT360 and Study IT in Estonia.

References

[1] S. Khalaf, "Seven years into the mobile revolution: Content is king again," August 2015, [Last access] 14 sep. 2016. [Online]. Available: http://flurrymobile.tumblr.com/post/127638842745/seven-years-into-the-mobile-revolution-content-is

[2] A. Smith, "Mobile mania! australians spend on average more than an hour a day on their smartphones," October 2015, [Last acess] 14 Sep. 2016. [Online]. Available: http://www.nielsen.com/au/en/insights/news/2015/mobile-mania-australians-spend-on-average-more-than-an-hour-a-day-on-their-smartphones.html.

[3] comScore, "comScore Reports January 2016 U.S. Smartphone Subscriber Market Share," March 2016, [Last acess] 14 Sep. 2016. [Online]. Available: http://www.comscore.com/Insights/Rankings/comScore-Reports-January-2016-US-Smartphone-Subscriber-Market-Share.

[4] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications magazine*, vol. 43, no. 9, pp. S23–S30, 2005.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. New York, NY, USA: ACM, 2012, pp. 13–16.

[6] 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, Oct 2009.

[7] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Computers and Communications (ISCC), 2012 IEEE Symposium on*. IEEE, 2012, pp. 000 059–000 066.

[8] Y. Xing, Z. Yang, C. Chen, and Y. Dai, "Beehive: low-cost content subscription service using cloudlets," *Science China Information Sciences*, vol. 56, no. 7, pp. 1–16, 2013.

[9] X. Wang, T. Kwon, Y. Choi, H. Wang, and J. Liu, "Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 72–79, 2013.

[10] W. Cai, V. C. Leung, and L. Hu, "A cloudlet-assisted multiplayer cloud gaming system," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 144–152, 2014.

[11] Y. Lin and H. Shen, "Cloudfog: Leveraging fog to extend cloud gaming for thin-client mmog with high quality of experience," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[12] J. Wu and Y. Wang, *Opportunistic Mobile Social Networks*. CRC Press, 2014.

[13] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: middleware for mobile social networking," in *Proceedings of the 2nd ACM workshop on Online social networks*. ACM, 2009, pp. 49–54.

[14] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi, "Prometheus: User-controlled p2p social data management for socially-aware applications," in *Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware*. Springer-Verlag, 2010, pp. 212–231.

[15] K. Zhu, W. Li, X. Fu, and L. Zhang, "Data routing strategies in opportunistic mobile social networks: Taxonomy and open challenges," *Computer Networks*, vol. 93, pp. 183–198, 2015.

[16] C. Chang, S. N. Srirama, and S. Ling, "Towards an adaptive mediation framework for mobile social network in proximity," *Pervasive and Mobile Computing*, vol. 12, pp. 179–196, 2014.

[17] ——, "Mobile social network in proximity: taxonomy, approaches and open challenges," *International Journal of Pervasive Computing and Communications*, vol. 11, no. 1, pp. 77–101, 2015.

[18] BBC NEWS, "India blocks zuckerberg's free net app," February 2016, [Last access] 15 Sep. 2016. [Online]. Available: http://www.bbc.com/news/technology-35522899

[19] S. A. Golden and S. S. Bateman, "Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1185–1198, 2007.

[20] M. Liyanage, C. Chang, and S. N. Srirama, "mepaas: Mobile-embedded platform as a service for distributing fog computing to edge nodes," in *Proceedings of the 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT-16)*, Guangzhou, China, 2016, p. [Accepted to be published].

[21] A. Kalbag and G. Silverman, "Enriching Business Processes through Internet of Everything," Cisco, Tech. Rep., 2014.

[22] J. Mass, C. Chang, and S. N. Srirama, "Wiseware: A device-to-device-based business process management system for industrial internet of things," in *Proceedings of the 9th IEEE International Conference on Internet of Things*. Chengdu, China: IEEE, 2016, pp. 269–275.