CHAPTER 1

INTRODUCTION

1.1 Background

Wireless Sensor Network (WSN) research and technology advances continuously for different applications. The difficulty of WSN hardware and software design has increased. The design needs to fulfill its continuously adaptation to very diverse application requirements and operating conditions[7]. The dimension of a sensor node can vary depending on the actual implementation. As the development progress, WSN is one of factor to enabling IoT along automation, control systems and embedded systems. IoT itself is a system of interrelated computing devices such as WSN which is able to transfer data over a network as an extension of the Internet with little or no need for human interaction [8].

In big data era, adequate network and computation infrastructure needed to maximize IoT benefit which is low-latency and fast response time for IoT applications. Cloud computing has been one of main IoT supporting factor because big storage and high computation capacity. Cloud supported IoT system facing few challenges like high response time with cloud server because located far from the end-users. Massive data transmission generated by IoT to cloud are inefficient because high bandwidth cost and high data redundancy. Instead, data transmission to cloud are more efficient if applications, storage, and processing moved closer to data generated by IoT devices [9].

Fog computing is an option to moved cloud service closer to the Iot devices and end-users. Fog is a layer between cloud layer and IoT layer which is supporting low-latency, location awareness, and geography distribution for IoT devices. Following the concept of cloud computing, fog computing provides computation, storage, and network service for end-users[10]. The main idea is to process a request that needed real-time service and low-latency. Furthermore, request that need permanent storage or advance analysis will be send to cloud.

Clustering is a task of dividing the populations or data points into a number of groups based on certain similarities. A great network congestion and data collisions will be experienced by sensor network if each and every sensor start to communicate and engage in data transmission in the network. To address these issues node clustering is important for sensor network applications where a large number of sensors are deployed for sensing purpose [11]. Centroid models clustering algorithm such as K-means clustering algorithm is the simplest unsupervised learning algorithm [12]. One of K-means clustering algorithm advantage is clusters have similar density which is will be benefit for load balancing.

In previous research about IoT offloading, research [13] introduce general framework for IoT-fog-cloud applications. The research [13] proposed a collaboration between minimizing delay and offloading policy to reduce IoT application service delay. Proposed policy in the research consider interaction between IoT-to-cloud and fog-to-cloud, and using fog-to-fog communication to reduce service delay by load sharing. The research does not give limitation to a particular architecture and number, type, IoT node topology, fog node, cloud server not restricted.

In contrast to the previous research [13], this research limiting the proposed framework on clustered wireless network topology to analyze how offloading framework behaved. K-means clustering algorithm is used for clustering the IoT nodes and to determine the total fog nodes. The contribution of the research is threshold-based offloading framework.

1.2 Problem Identification and Objective

Hundreds even thousands IoT devices can be connected to one access point for a certain range. The requirement characteristics of IoT services is a low latency for communication purpose. Cloud computing is seen as the central computational backbone of the IoT but have a downside like high latency which is unsuitable for IoT services. Fog offloading aims to reduce cloud involvement to process a task. Improving utilization of fog node leads to faster task execution but comes with few challenges. One of the challenges is to provide a framework to address decentralized processing of request.

Therefore, to address the challenge, the main work of this research is to develop an adaptive threshold-based offloading framework. The framework provides a few functions such as request handling and offloading mechanism.

1.3 Related Research and Contribution

The problem of task offloading in fog has recently gained attention from researchers. A problem to task offloading in the fog (fog offloading), is the task or workload assignment problem in the fog, which is assigning tasks or workloads to either fog nodes or cloud servers, while minimizing delay, cost, or energy. In these problems, the set of tasks are given and the problem is modeled as a static optimization problem that determines the assignment of the tasks or workloads. Namely, Gu et al. [14] studied base station association, task distribution, and virtual machine placement in fog-computing-supported medical cyber-physical systems, while minimizing the overall cost and satisfying QoS requirements. Another effort is the work in [?] that addresses power-delay trade-off in cloud-fog computing by workload allocation. Another work in [15] focuses on theoretical modeling of fog computing architectures, specifically, service delay, power consumption, and cost.

The work in [16] addressed the design of a policy for assigning tasks that are generated by mobile subscribers to edge clouds, to achieve a power-delay trade-off. The problem is first formulated as a static optimization problem. The authors then propose a distributed policy for task assignment that could be implemented efficiently on the edge clouds. However, IoT-to-cloud or fog-to-cloud communication and computation offloading capability are not considered. Moreover, in the distributed task assignment policy, edge clouds broadcast their status continuously to mobile subscribers, which may limit scalability in IoT networks with a large number of edge devices and end users.

The work in [13] address the problem of fog offloading for reducing IoT service delay, which is a fundamentally different problem from the problems discussed above. The proposed offloading policy is general as opposed to studies as it does not place any restrictions on the number or topology of the fog nodes. In this research, we study the problem of fog offloading for reducing IoT service delay, similar to the work [13], [17] and [18] but within clustered IoT networks and consider IoT-Fog-Cloud cooperation for task offloading.

This research proposes a threshold-based offloading framework for clustered IoT nodes consist of average waiting time threshold and queue length threshold. Waiting time threshold is applied to fog nodes queue and queue length threshold is applied to offloaded requests queue. Both threshold are used to create offloading decision. K-means clustering algorithm is utilized for clustering the IoT nodes. To determine the optimum cluster number for IoT nodes, Calinski-Harabasz cluster validation is used.

1.4 Scope of Works

Analysis and discussion in this research will apply few certain limitation:

- Proposed framework will be considering fog-to-cloud, and fog-to-fog interactions.
- Proposed framework will be considering queue length and waiting time as threshold.
- Optimum threshold values are obtained using brute force method.
- IoT nodes used in this research are static nodes.
- Requests are generated using five different rate of request intergeneration time.
- Network security not discussed in this research.

1.5 Hypothesis

Proposed offloading framework in this research will keep end-to-end one-way delay below 150 milliseconds for cloud supported IoT application as recommended on ITU-T Rec. G.114.

1.6 Research Methodology

Methodology in this research are as follow:

- Problem Identification
  Problem identification in this research will be conducted by literature study from papers or journal about related previous research.
- System Model Design
  Network architecture consist of sensor layer, fog layer, cloud layer.
- Framework Design
  Designing the framework used in this research.
- System integration
  System model and framework design will be integrated.
- Simulation Process
  Simulation will be conducted on designed network architecture and framework to test few different scenarios.
- Simulation Result Analysis
  Simulation result will be reviewed and analyze to verify if the research objective achieved.
- Conclusion
  Conclusion withdrawal will be taken based on simulation and result analysis. Final conclusion must meet the main purpose of the research.

1.7 Writing Systematic

This research consist of 5 chapter as follow:

1. CHAPTER 1 Introduction
   This chapter discussed problem background, problem identification, research objective, scope of works, hypothesis, research methodology, and writing schematic.
2. CHAPTER 2 Literature Study
   This chapter discussed theory and concept about the research based on existing literature.

3. CHAPTER 3 Research Methodology
   This chapter discussed design process, framework, simulation design, simulation scenario, simulation parameter, and performance metric measurement method.
4. CHAPTER 4 Presentation, Analysis and Interpretation of Data
   This chapter discussed performance metric analysis based on simulation result using different scenario.
5. CHAPTER 5 Conclusion and Future Works
   This chapter discussed the research conclusion and future works idea based on analysis result.