B. Tech. Project Problem Statement

Siddharth Verma, Neeraj Kamal

January 2023

1 Introduction

The healthcare industry has witnessed significant transformations in e-health services where Electronic Health Records (EHRs) are transferred to edge clouds to facilitate healthcare.
Digital healthcare alternatives such as EHRs have gained prominence with increased patients' data volume.
With the aim of reducing healthcare costs and providing improved and reliable services, several healthcare frameworks based on Internet of Healthcare Things (IoHT) have been developed.

2 Challenges in Existing Schemes

Due to the critical and heterogeneous nature of healthcare data, maintaining high quality of service (QoS), i.e., responsiveness and data-specific complex analytics has always been the main challenge in designing such systems.
Traditional EHR-based systems are affected by data loss risks, security and immutability consensus over health records, gapped communication among constituted hospitals, and inefficient clinical data retrieval systems, etc.
Many edge cloud-based system designs have been proposed, but some technical challenges still remain, such as low QoS, data privacy and system security due to centralized healthcare architectures.

3 Problem Formulation

We aim to implement a novel hybrid approach of data offloading and data sharing for healthcare using edge cloud and blockchain.
An efficient five-layered heterogeneous mist, fog, and cloud-based data offloading scheme where IoT health data can be offloaded to nearby edge servers for data processing with privacy awareness.
A data sharing scheme is integrated to enable secure EHRs sharing among healthcare users via blockchain.
A reliable access control mechanism is developed to jointly optimize time latency, energy cost, memory usage and resource allocation under system constraints providing high QoS.

4 System Architecture

We consider a healthcare system architecture comprising of 4 layers:

1. IoT Layer
2. Edge Layer
3. Cloud Layer
4. End-user or Application Layer

The Edge layer consist of 3 layers used for efficient data offloading and pre-processing:

1. Perception Layer
2. Mist Layer
3. Fog Layer

5 IOT LAYER

It consists of many smart hospitals which monitor patients by sensor IoT devices in different locations.

Figure 1: IoT Layer

6 EDGE LAYER

It includes a cluster of edge cloud nodes, each edge node manages a group of nearby IoT devices to provide distributed computing services for healthcare.
It is further subdivided into perception, mist and fog layers.

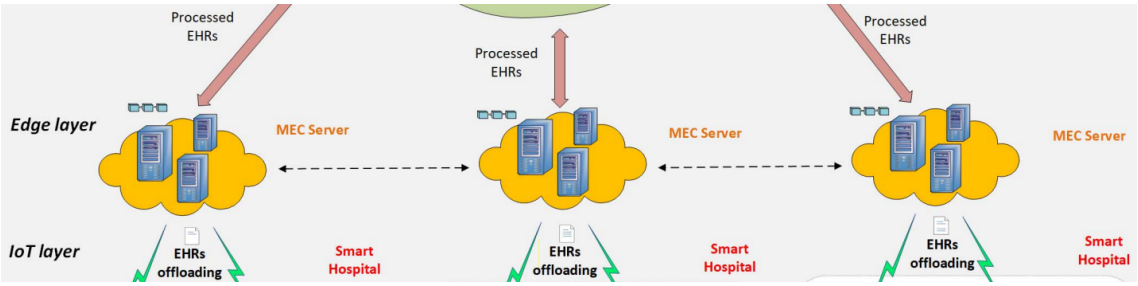

Figure 2: Edge Layer

7 PERCEPTION LAYER

It is used for gathering real-time and non-real-time healthcare data generated from devices.
The data are measured from individuals and their surroundings through small sensors, embedded systems, diagnostic and healthcare devices, medical imaging devices, etc.

Figure 3: Perception Layer

8 MIST LAYER

It performs basic rule-based pre-processing of the sensor data (e.g., data aggregation, fusion, and filtering). and facilitates time-critical data processing.
Since communication consumes approximately 5 times the power of computing, ensuring required transmission instead of on-demand transmission optimizes the power consumption.

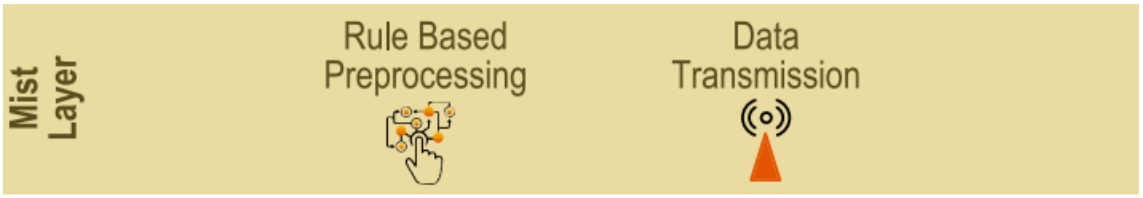

Figure 4: Mist Layer

9 FOG LAYER

It brings the computing resources and application services closer to the edge, and thus reduces the response latency.
It involves local data storage, data filtering, data compression, data fusion, and intermediate data analytics to reduce disposable load on the cloud— improving system performance.

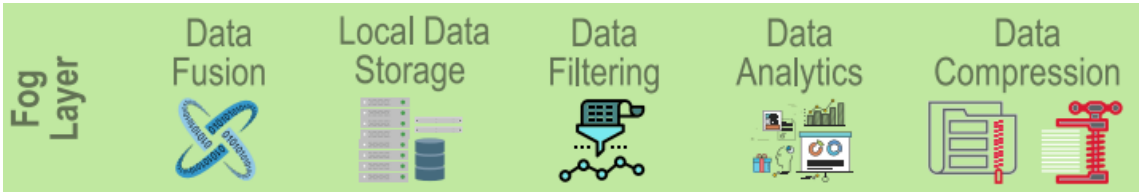

Figure 5: Fog Layer

10 CLOUD LAYER

It stores processed health data from edge nodes in a blockchain and performs data sharing with end users.
The cloud layer performs various advanced data analytics including machine learning, data mining, rule-based processing, and automated reasoning-based algorithms.
Delegating appropriate computing loads to fog layer and using cloud layer for computationally expensive operations improves system performance.

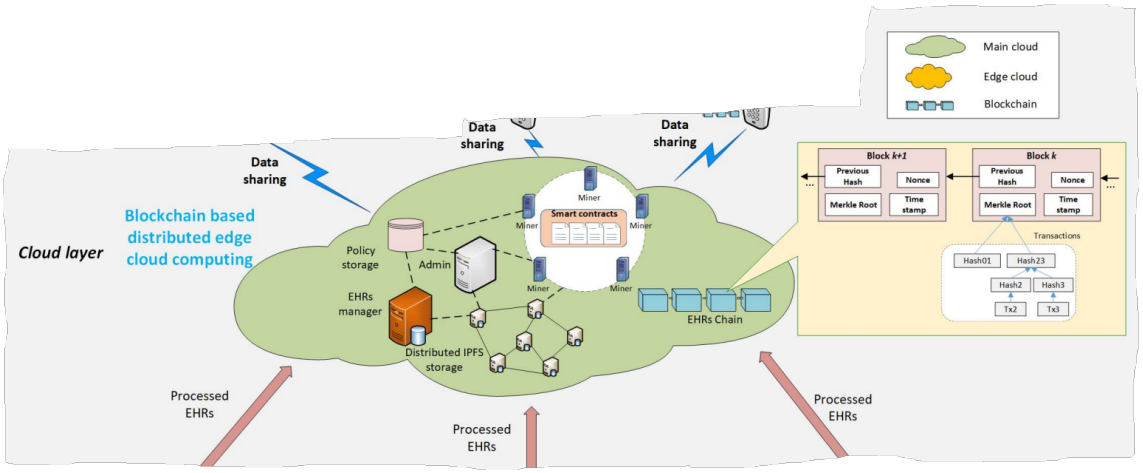

Figure 6: Cloud Layer

11 APPLICATION LAYER

It provides user interface between the IoHT stakeholders like healthcare providers, caregivers and patients and the framework to avail the generated economic and social benefits.

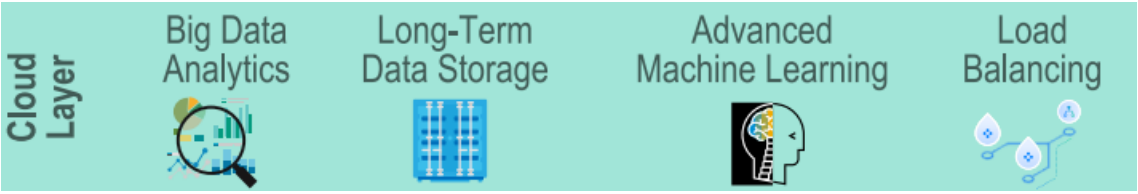

Figure 7: Cloud Layer - alternative representation

For instance, doctors use analysed health data on cloud for disease diagnosis, or patients can track their medical record history.

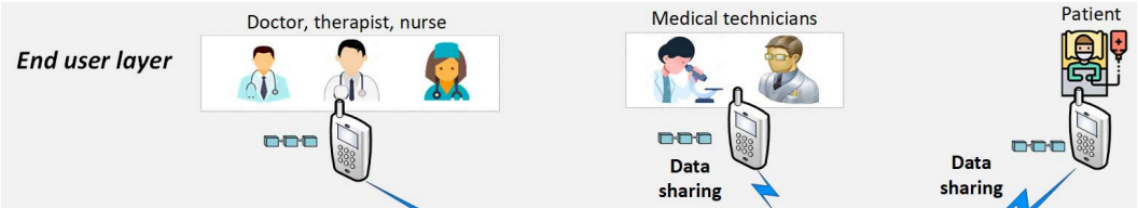

Figure 8: Application Layer

Figure 9: Application Layer - alternative representation

12 HEALTH DATA OFFLOADING MODEL (Between IoT and Edge Layers)

We consider that each IoT device has multiple health data tasks N = 1, 2, ..., N to be executed.
We introduce an offloading decision policy denoted by a binary variable x_n^t belongs to 0, 1, where $x_n^t = 1$ means that task n is offloaded to the edge server, otherwise it is executed locally if $x_n^t = 0$.
There are 2 main modules on a device: task profile and decision maker.

13 TASK PROFILE

This module collects device information such as energy consumption (E), processing time (T) and memory usage (M).
A task profile with a size D_n bits can be formulated as a variable tuple $[D_n, E_n, T_n, M_n]$ which is then stored in a database created on the device for supporting offloading decisions.

14 DECISION MAKER PROFILE

This module receives task profile information collected by the profile module to make offloading decisions. By using profile information, the algorithm (discussed ahead) analyses and makes decisions for executing locally or offloading to the edge server.
The main objective is to determine an optimal computation decision for each task to minimize computing latency, energy consumption and memory usage.

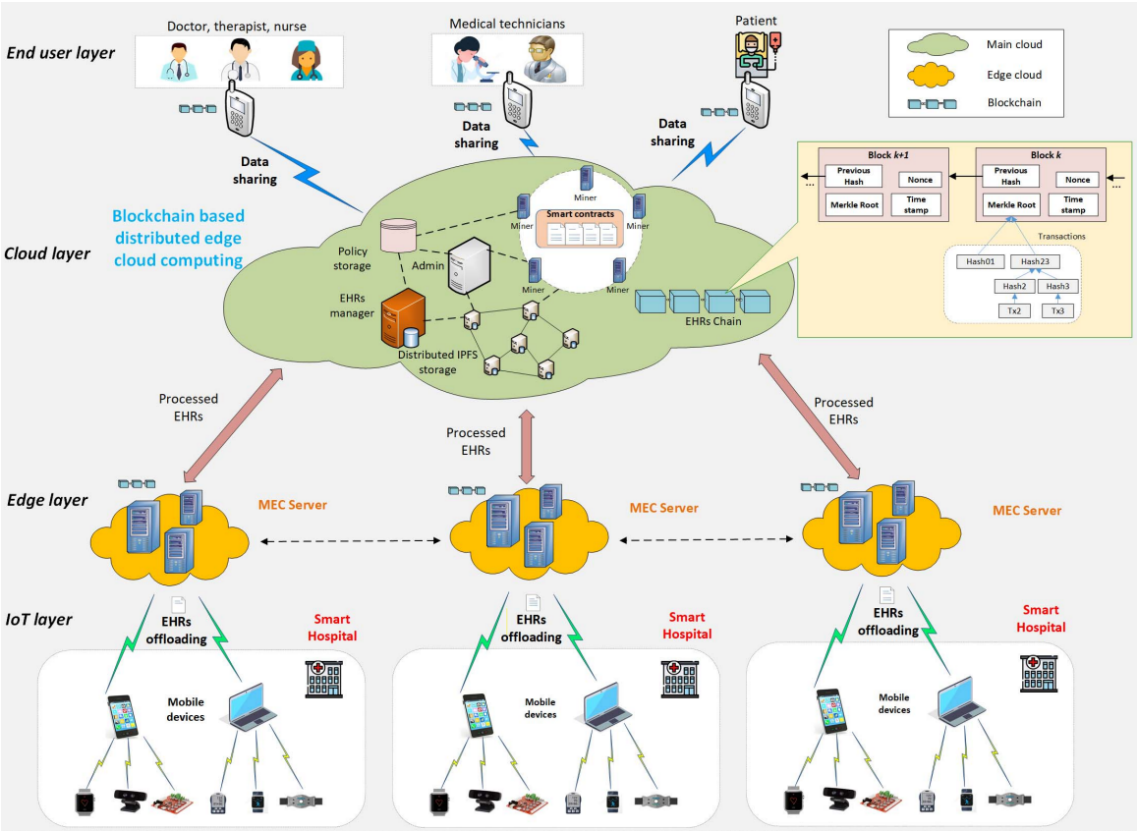

Figure 10: System architecture of major layers

15 DECISION MAKING ALGORITHM

Consider each task N consists of a set of parameters, namely, input size ($input_i$), memory usage ($memo_i$), CPU utilization (CPU_i), and battery consumption ($power_i$), for local execution.
Memory used for security (memo_sec$_i$), battery consumption used for security (power_sec$_i$), and CPU used for security (CPU_sec$_i$) are considered if the method is offloaded for remote execution.

16 DECISION MAKING ALGORITHM'S OBJECTIVE

$$\min_{x \in 0,1} (c_{transfer} * w_{tr} + c_{memory} * w_{mem} + c_{CPU} * w_{CPU} + c_{power} * w_{power}),$$

$$\begin{aligned}
C_{transfer} &= \sum_{i=1}^{n} input_i * x_i \\
C_{memory} &= \sum_{i=1}^{n} memo_i * (1 - x_i) + \sum_{i=1}^{n} memo_sec_i * x_i \\
C_{CPU} &= \sum_{i=1}^{n} CPU_i * (1 - x_i) + \sum_{i=1}^{n} CPU_Sec_i * x_i \\
C_{power} &= \sum_{i=1}^{n} power_i * (1 - x_i) + \sum_{i=1}^{n} power_sec_i * x_i,
\end{aligned} \tag{1}$$

where, $C_{transfer}$, C_{memory}, C_{CPU}, and C_{power} represent cost for transferring the input size, memory used, CPU used, and power consumed for a task respectively.
w_{tr}, w_{memo}, w_{CPU}, and w_{power} are the weights for each these costs, which lead to different objectives.

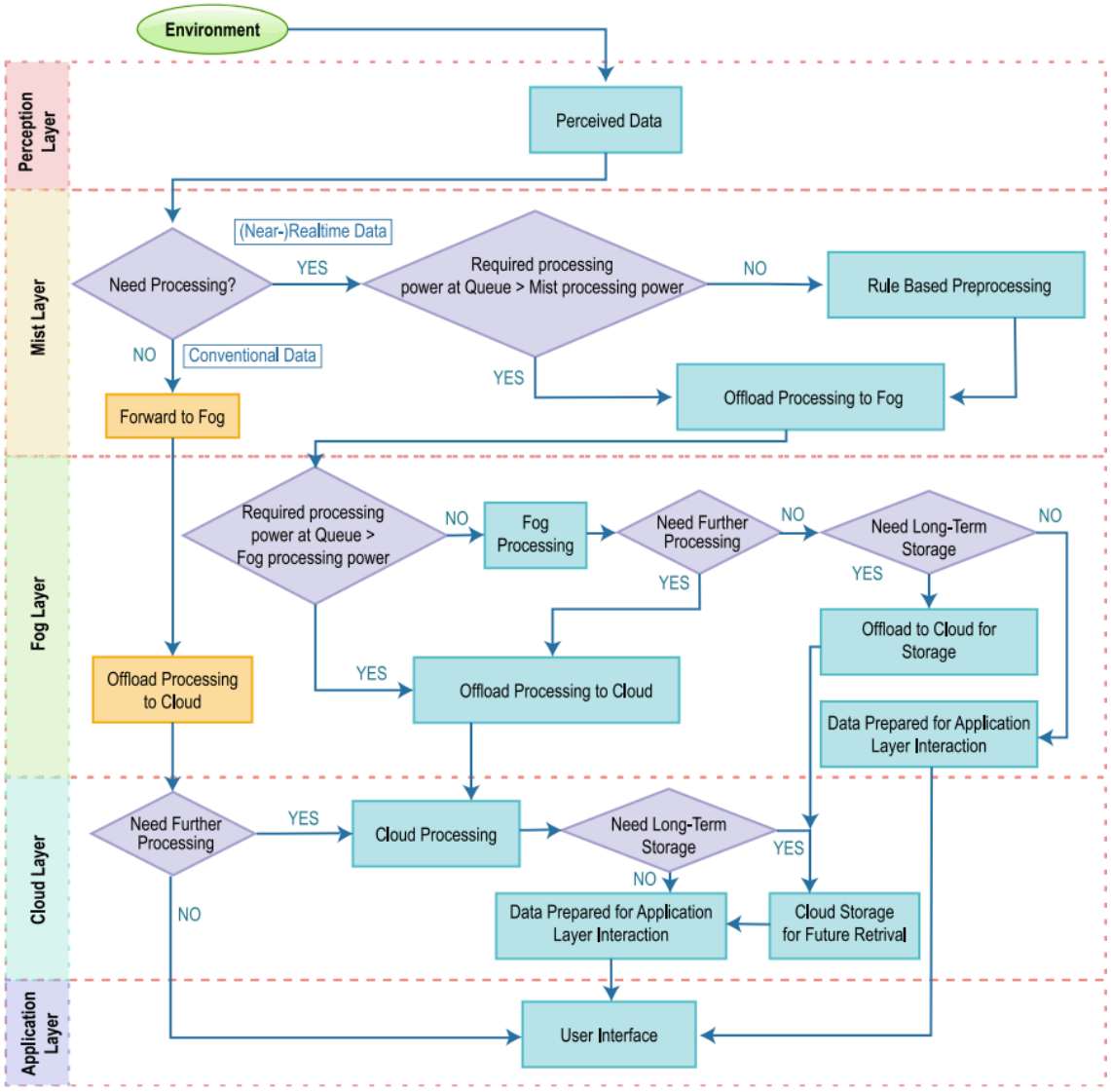

Figure 11: System architecture of sub layers

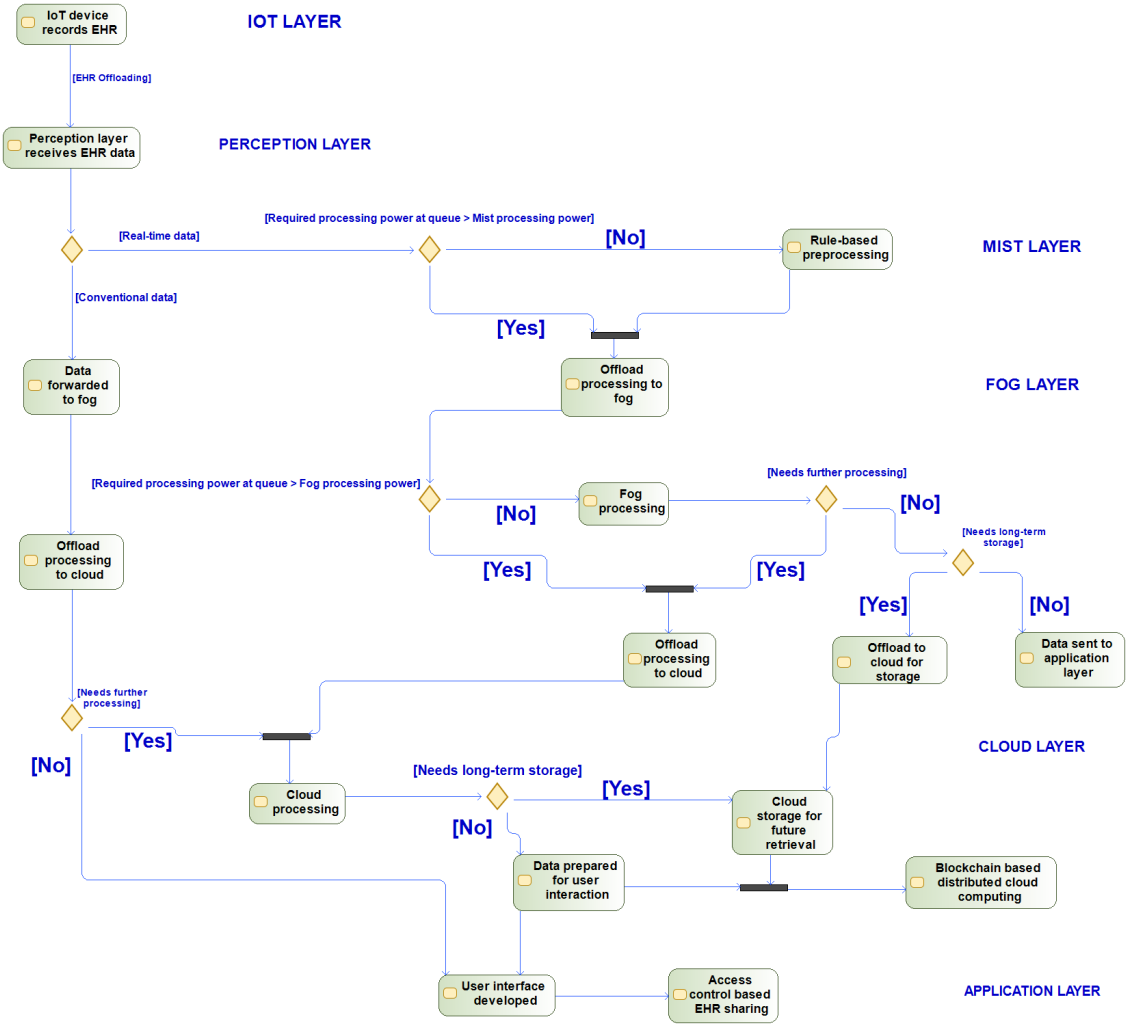

Figure 12: Activity diagram of System Architecture

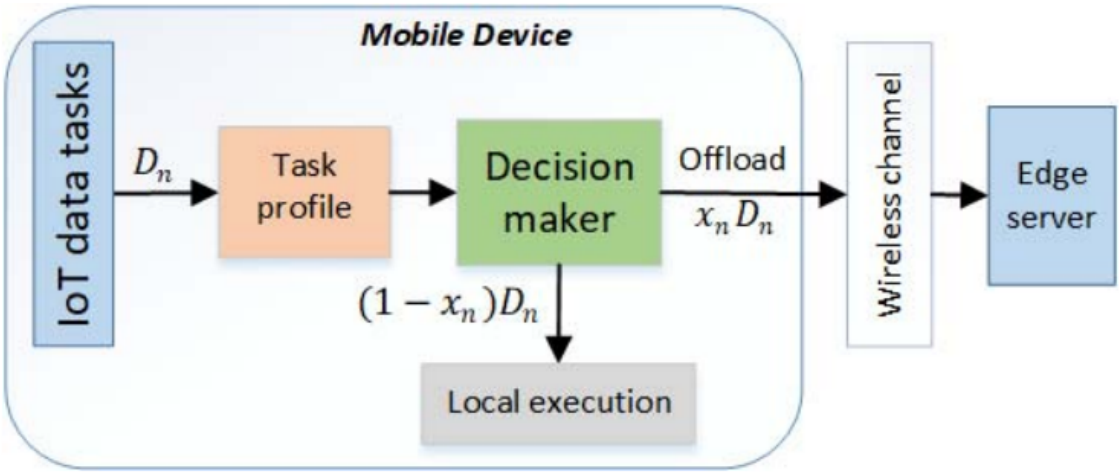

Figure 13: Data offloading scheme

17 HEALTH DATA OFFLOADING FORMULATION

We formulate the health data offloading problem with three main metrics, namely processing time, energy consumption and memory usage under two computation modes.
Local Execution: The local execution time is calculated as: $T_n^{local} = (D_n X_n^l)/f_n^l$, where X_n^l : device CPU utilization for task n (in CPU/bit) f_n^l : mobile CPU utilization for task n (CPU/sec)
Offloading to edge server: The data task needs to be encrypted for security before transmitting to the edge server. Let X_n^{enc},X_n^e, f_n^e, r_n as mobile CPU utilization for encrypting the task n (in CPU/bit), edge CPU utilization (in CPU/bit), edge computational capacity (in CPU/sec), and the device's transmission data rate respectively.
The total offloading time can be expressed as $T_n^{offload}$ =($(D_n X_n^{enc})/f_n^l + (D_n X_n^e)/f_n^e + D_n/r_n$).
We also define E_n^{enc} , E_n^{trans} as encryption energy and energy for transmitting the task n to the edge server respectively.
The total offloading energy is calculated as:
$E_n^{offload} = (E_n^{enc}((D_n X_n^{enc})/f_n^l) + E_n^{trans}(D_n/r_n))$.
The total offloading time, energy cost and memory usage can be expressed as follows:

$$T_n = (1 - x_n)\, T_n^{local} + x_n T_n^{of\ fload}, \tag{1}$$

$$E_n = (1 - x_n)\, E_n^{local} + x_n E_n^{of\ fload}, \tag{2}$$

$$M_n = (1 - x_n)\, M_n^{local} + x_n M_n^{of\ fload}. \tag{3}$$

18 OPTIMIZATION PROBLEM

The optimization problem to jointly optimize time latency, energy cost and memory usage under system constraints is as follows:

$$\begin{aligned} &\min_{\mathbf{X}} \sum_{n=1}^{N} (\alpha_t T_n + \alpha_e E_n + \alpha_m M_n) \\ \text{st.} \quad &(C1): \sum_{n=1}^{N} \left(x_n T_n^{of\ fload}\right) \le \sum_{n=1}^{N} (1 - x_n) T_n^{local}, \\ &(C2): \sum_{n=1}^{N} \left(x_n E_n^{of\ fload}\right) \le \sum_{n=1}^{N} (1 - x_n) E_n^{local} \Bigg), \\ &(C3): \left(\sum_{n=1}^{N} \left(x_n T_n^{of\ fload}\right) + \sum_{n=1}^{N} (1 - x_n) T_n^{local} \right) \le \tau, \\ &(C4): \left(\sum_{n=1}^{N} \left(x_n M_n^{of\ fload}\right) + \sum_{n=1}^{N} (1 - x_n) T_n^{local} \right) \le \zeta, \end{aligned} \tag{4}$$

α_t, α_e, α_m are the cost weights, all set to 1/3.
(C1), (C2) represent that the offloading cost of time delay and energy consumption should be less than the local execution cost when computing all healthcare data tasks on a device. (C3) states that the total task execution time should not exceed a maximum latency value τ. (C4) defines that the memory used for task computation must not exceed the available mobile memory ζ.

19 DATA TRANSMISSION POLICY(Across Edge Layer)

The perception layer generates three possible types of delay sensitive data, i.e., real-time, near-real-time, and offline/batch mode data.
In order to achieve better QoS, reduced latency, and optimized power consumption, separate transmission paths for real-time data and big data are used.
Based on data traffic and resource availability, the computational loads are allocated to an appropriate layer.

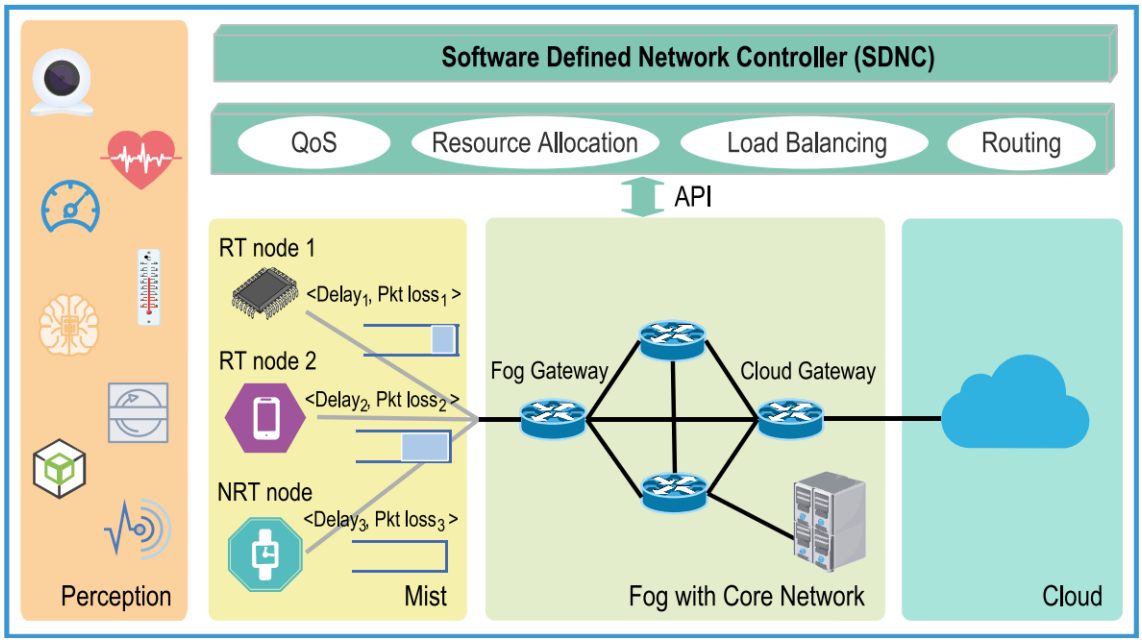

Figure 14: IoHT framework with mist, fog and cloud Layers. The network is configured using SDN. The AP allocates resources based on the latency and packet drop rate demand of each IoHT nodes.

20 IoHT framework with mist, fog and cloud layers

21 REAL-TIME DATA TRANSMISSION SCENARIO-1

If a patient experiences high blood pressure fluctuations along with symptomatic discomforts, it is necessary to process the generated data and forward a decision to the caregiver as soon as possible to prevent a possible stroke.
In this case, the pre-processed data from mist layer are further processed in fog layer and forwarded to the application layer for necessary actions by the stakeholders.

22 REAL-TIME DATA TRANSMISSION SCENARIO-2

The data and the analysis results are usually stored in the cloud for further reference.
Adverse drug reaction (ADR) service can be an example of this scenario.
Medication for a particular disease needs diagnosis as well as patient's previous history as ADR is inherently generic.
Data sensed from patient's terminal are forwarded to mist layer for recognizing the drug.
Fog layer forwards the identified drug to the cloud for testing the drug compatibility to be sent to the application layer.

23 REAL-TIME DATA TRANSMISSION SCENARIO-3

Massive data generated from advanced medical instruments, test results requires data mining, predictive analysis, and other advanced analytics.
Only cloud computing is capable of performing these computationally demanding processing.
Data from conventional sources are directly offloaded to the cloud for processing.
For example, MRI produces thousands of high resolution images per examination which require more computation power and storage, and can be efficiently served only by the cloud.

24 OPTIMAL RESOURCE ALLOCATION AND LOAD BALANCING

Traffic Classification
In order to achieve better QoS, these network traffic are classified as the delay-sensitive (DS), loss-sensitive (LS), and both delay and loss-sensitive (termed as "Mixed") (M) traffic.
This classification is mainly based on transmission data rate (C) and queuing delay (tQ), and is used to prioritize the network traffic.

25 TRAFFIC CLASSIFICATION

P	Traffic Type	Description		Service Type	Example
		C	tQ		
1	$\mathcal{DS}$	H	L	Critical Traffic	RT Patient Monitoring
				Video Traffic	VidStream EM & MC
				Multiconf.	Teleconf.
2	$\mathcal{LS}$	L	H	Images/Video	Medical Imaging
				Test Results	EMR
3	$\mathcal{M}$	M	M	NCMeasure	Regular PhyMeas

Legend: H–High; L–Low; M–Medium; RT– Real-time; VidStream EM & MC– Video streaming of elderly monitoring & motion control; Multiconf– Multimedia conferencing; Teleconf– Teleconferencing; PhyMeas–Patient physiological measurements; NCMeasure– Non-critical healthcare parameter measurement.

Figure 15: Traffic Classification of IoHT Healthcare Data

26 CRITICALITY INDEX FOR TRAFFIC CLASSIFICATION

We classify data based upon the value of the criticality index of the patient, which varies at any time instant.
Hence, any delay in the delivery of the physiological parameters may worsen the patient's condition. The criticality index CI_{wi} is given as:

$$CI_{W_i} = \frac{1}{\xi_{W_i}} \left(\sum_{p=1}^{x} \frac{|\mathcal{D}^p_{min} - S^p_{min}|}{S^p} + \sum_{q=1}^{y} \frac{|\mathcal{D}^q_{max} - S^q_{min}|}{S^q} + \sum_{r=1}^{z} \frac{(O^r_c - S^r_{min})}{S^r} \right)$$

where wi belongs to W1, W2, W3,. . . , WN, the set of sensors in an IoHT device.
Each sensor has a normal value, N, such that N_{min} ¡= N ¡= N_{max}.
The deviation of the sensor's value of any patient beyond N_{min} and N_{max} is considered to be critical.
The value of the physiological sensor less than N_{min} is denoted by difference D_{min} and the physiological sensor value more than N_{max} is denoted by D_{max}.
S_{min} and S_{max} denote the minimum and maximum possible value of a sensor, such that $S_{min} \leq N \leq S_{max}$.
S = S_{max} - S_{min}
Mathematically, $D_{min} = N_{min} - O_c$,and $D_{max} = O_c - N_{max}$ where O_c is the current observed value of the sensor.
Let x, y, and z denote the set of physiological sensors, which attain $O_c < N_{min}$, $O_c > N_{max}$, and N_{min} $O_c \leq N_{max}$ respectively. $x + y + z = \sigma_i$

27 RESOURCE ALLOCATION

To achieve better QoS, the objective is to reduce the time delay (tD) and packet drop rate (Pkt_{drop}) during the transmission process.
As the allocated resource to ith IoHT device is proportional to the requirement of that user, the maximum resource Ti awarded to the ith IoHT node is:

$$\begin{aligned}\Gamma_i &= \max\left[\frac{B_i}{C}, \frac{L_i}{L}\right] \\ &= \max\left[\frac{B_i^d}{C}\frac{B_i}{B_i^d}, \frac{L_i^d}{L}\frac{L_i}{L_i^d}\right] = \max\left[D_i^c U_i^c, D_i^l U_i^l\right]\end{aligned} \tag{1}$$

C : output link capacity of an access point AP, i.e., IoHT device
¡tD_i, Pkt_{drop}¿ : user requirement for best QoS
¡B_i^d, L_i^d¿ : corresponding resource demand
B_i^d : bandwidth demand of ith IoHT device
L_i^d : buffer length demand of ith IoHT device
$D_i^c = B_i^d/C$: ratio of bandwidth demand of the ith node and the maximum capacity
$D_i^l = L_i^d/L$: ratio of buffer length demand of the ith node and the total buffer length of AP
$U_i^c = B_i$ / B_i^d : requirement to demand ratio of bandwidth for ith node
$U_i^l = L_i$ / L_i^d : requirement to demand ratio of buffer length for ith node
The E2E delay tD includes the transmission delay tTx, processing delay tP, and queuing delay tQ which are calculated by:

$$\begin{aligned}tD =& tTx_i + tP + tQ \\ =& \sum_{cl}\sum_{\text{fog}}\sum_{\text{sen}}\left[\frac{N_{\text{pkt}}\text{Pkt}_{\text{size}}}{C} + \left(\frac{\lambda}{2\mu(\mu-\lambda)} + \frac{1}{\mu}\right) + c\lambda\right]\end{aligned} \tag{2}$$

λ and μ are the arrival and service rate, N_{pkt} is the number of packets, c is the constant duration required to complete a job by a processor, sen refers to sensor and cl refers to the cloud.
The packet drop occurs when the average queuing length E[Q_i] is higher than demanded buffer length L_i^d/Pkt_{size}.
The packet drop rate is expressed by:

$$\text{Pkt}_{\text{drop}_i} = \frac{E[Q_i] - L_i^d/\text{Pkt}_{\text{size}}}{E[Q_i]}. \tag{3}$$

The resource allocation optimization problem is formulated as:

$$\begin{aligned}\max\ & \left[\left(U_1^c, U_1^l\right), \left(U_2^c, U_2^l\right), \ldots, \left(U_N^c, U_N^l\right)\right] \\ \text{s.t}\ & \sum_{n=1}^{N} B_n \le C \\ & \sum_{n=1}^{N} L_n \le L.\end{aligned} \tag{4}$$

28 LOAD DISTRIBUTION

In the Fog/access layer, the E2E latency can be reduced by link distribution and link fusion techniques. If the link scheduler selects M links based on the demand of the IoHT users, the link adaptation opti-

mization problem can be formulated as:

$$\begin{aligned} &\max\ f(T, 1/\mathrm{Pkt}_{\mathrm{drop}}) \\ &\text{s.t.}\ \sum_{m=1}^{M} \gamma_m B^d \leq C \\ &\quad \sum_{m=1}^{M} \gamma_m L^d \leq L \end{aligned} \tag{5}$$

where T is throughput, γ_m is the fraction of bandwidth/buffer length allocated by the load balancer and is expressed by:

$$\begin{aligned} \gamma_m &= \text{Fraction of allocation in the } m\text{link} \\ &= \left(\frac{B_m}{\sum_m B_m}\right)\beta + \left(\frac{L_m}{\sum_m L_m}\right)(1-\beta) \end{aligned} \tag{6}$$

$$\beta = \begin{cases} 0, & \text{if Traffic type is } \mathcal{LS} \\ 1, & \text{if Traffic type is } \mathcal{DS} \\ 0.5, & \text{if Traffic type is } \mathcal{M}. \end{cases}$$