FOG COMPUTING

- The rapid developments of cloud computing have brought a centralized solution to application developers and content providers.

- Limitations: { (1) High Latency (2) Delay } , due to long distance between end users and the devices.

- With the motivation of placing the services as close as possible to end users, researchers have proposed a new cloud system called fog computing.

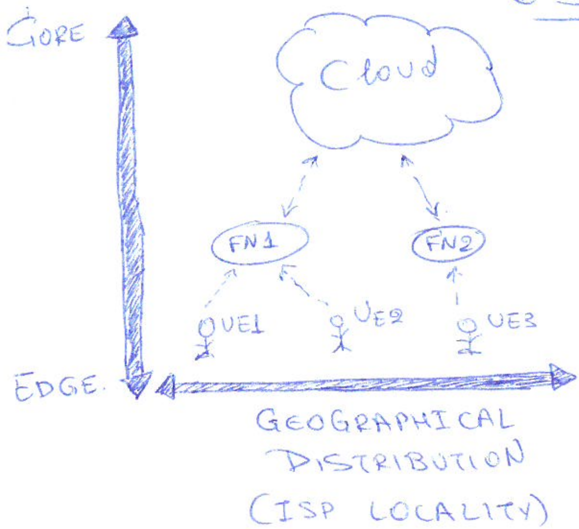
Fog Nodes (FN): [• End-user devices • Access Points • Edge routers - switches] ⇒ Edge of the Network

- Functionalities
  (1) Converged Computing
  (2) Processing
  (3) Management
  (4) Networking
  (5) Storage
  (6) Physical and Cyber Security.

- Service Providers (SPs) can rapidly deploy certain applications and services to improve the Quality of Service (QoS) toward end users.

⇒

System Model



Note

Each fog node can have different utilization levels.

- We assume that in our network we have $|N|$ fog nodes, e.g., $N = \{1, \ldots, n, \ldots, |N|\}$ different fog nodes.

- We assume that we have $|U|$ users (UEs), e.g., $U = \{1, \ldots, u, \ldots, |U|\}$ which are uniformly distributed in the geographical area.

- We consider that each fog node $k \in N$, has a computational capability F_k and a maximum threshold of bits in order to be operational $\bar{B}_k$: So $\Rightarrow$ $\left[\begin{array}{l} \cdot F_k : \frac{\text{CPU-cycles}}{\text{sec}} \\ \\ \cdot \bar{B}_k : \text{bits} \end{array}\right]$

- Each user has a flow of computational tasks, that wants to offload in a fog node.

- We will study the system in a time-slotted manner, where each timeslot is denoted with t.

- In every timeslot t each user $u \in U$, wants to offload a task t_u to the fog node.

$\Rightarrow$

- Each task's t_u characteristics are:

$$t_u = \{I_{t_u}, \phi_{t_u}, C_{t_u}\}$$ where:

- → I_{t_u} : Application's bits [Bits]
- → ϕ_{t_u} : Application's Intensive Parameter $\left[\frac{\text{CPU-cycles}}{\text{bits}}\right]$
- → C_{t_u} : Application's CPU cycles [CPU-cycles]

The communication characteristics between user u and fog node k are:

$$P_{u,k} = \left[\frac{Dist_{u,k}}{Dist_k^{max}}\right]^{\vartheta_1} \qquad \boxed{(Eq. 2)}$$

and

$$g_{u,k} = \left[\frac{1}{Dist_{u,k}}\right]^{\vartheta_2} \qquad \boxed{(Eq. 3)}$$

where: $Dist_k^{max}$ ⇒ Max distance

$Dist_{u,k}$ ⇒ Distance between end-user u and fog node k.

Transmission Rate: $$R_{u,k} = W_k * \log_2\left(1 + \frac{P_{u,k} * g_{u,k}}{G_0 + \sum_{\substack{u' \in V_k \\ u' \neq u}} P_{u',k} * g_{u',k}}\right) \qquad \boxed{(Eq. 4)}$$

where:
(1) W_k ⇒ Bandwidth of fog node k.
(2) G_0 ⇒ Additive White Gaussian Noise
(3) V_k ⇒ Number of users that offload to fog node k.

⇒

⇒ Overhead that a user u experiences by offloading to fog node k.

- Time Overhead

$$O_t^u = \frac{I_{tu}}{R_{u,k}} + \frac{C_{tu}}{f_{u,k}} \quad \boxed{(\text{Eq. } 5)}$$

where: (1) $\frac{I_{tu}}{R_{u,k}}$ ⇒ Transmission Time of the task t_u from user u to fog node k.

(2) $\frac{C_{tu}}{f_{u,k}}$ ⇒ Processing Time in fog node k.

(3) $$f_{u,k} = \underbrace{\frac{\Phi_{tu}}{\sum_{u \in U_k} \Phi_{tu}}}_{\text{Fairness Factor.}} * \underbrace{\left[1 - \frac{\sum_{u \in U_k} I_{tu}}{B_k}\right]}_{\text{Congestion Factor.}} * F_k \quad \boxed{(\text{Eq. } 6)}$$

* So, the $f_{u,k}$ is the computational power that corresponds to every user u that is assosiated with fog node k.

- Energy Overhead

$$O_e^u = \frac{I_{tu} * P_{u,k}}{R_{u,k}} \quad \boxed{(\text{Eq. } 7)}$$

- Total Actual Overhead

Total Overhead = Time Overhead + Energy Overhead

$$= O_e^u + O_t^u$$

$$= \frac{I_{tu} * P_{u,k}}{R_{u,k}} + \frac{I_{tu}}{R_{u,k}} + \frac{C_{tu}}{f_{u,k}} \quad \boxed{(\text{Eq. } 8)}$$

⊛ 1st LAYER : FOG NODE ASSOCIATION DESICION

We will utilize a Reinforcement Learning mechanism:

- The Stochastic Learning Approach (SLA).

- So, each user should determine with which fog node, will he be associated with. So, we determine the following:

- User Centric Cluster Performance (UCCP)

$$UCCP_k = \frac{\sum_{u \in U_k} \left[\frac{O_t^u}{t} + \frac{O_e^u}{e_u} \right]}{|U_k|} \qquad \boxed{(Eq. 9)}$$

where t: Timeslot duration.

e_u: User's u energy availability (e.g., battery).

- Congestion Factor

$$Cong_k = \frac{\sum_{u \in U_k} I_{tu}}{\sum_k \sum_{v \in U_k} I_{tu}^k} \qquad \boxed{(Eq. 10)}$$

- Reputation Factor

$$RF_k = \frac{\frac{\sum_{u \in U_k} f_{u,k}}{|U_k|}}{\sum_k \sum_{u \in U_k} \frac{f_{u,k}}{|U_k|}} \qquad \boxed{(Eq. 11)}$$

⇒

- Actual Reputation

μ_k : It will be modeled as a Bayesian Belief [Formulation in Layer 3].

* SLA Reward Function

$$R_k^u = \frac{\mu_k}{UCCP_k * Cong_k * RF_k} \quad \boxed{(\text{Eq. } 12)}$$

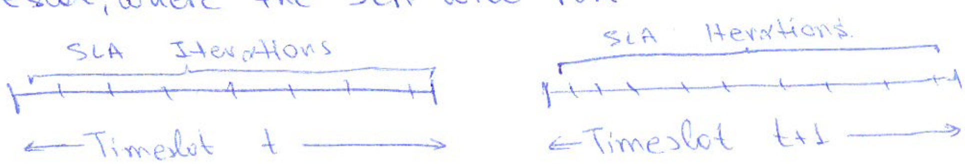
* Normalized SLA Reward Function

$$\hat{R}_k^u = \frac{R_k^u}{\sum_k \sum_{u \in U_k} R_k^u} \quad \boxed{(\text{Eq. } 13)}$$

→ Decision Making with SLA

(1) For every user u there are k strategies, i.e., the user every time can choose one of the available k fog nodes to be associated with.

(2) All the aforementioned equations will be used inside a timeslot, where the SLA will run



(3) So, in every timeslot t the user will receive his/her reward and afterwards he/she will renew the action probabilities.

$$\begin{cases} Pr_{uk}(t+1) = Pr_{uk}(t) + b * \hat{R}_k^u * (1 - Pr_{uk}(t)), & k(t+1) = k(t) \\ Pr_{uk'}(t+1) = Pr_{uk'}(t) - b * \hat{R}_k^u * Pr_{uk}(t), & k(t+1) \neq k(t) \end{cases} \quad \boxed{(\text{Eq. } 14)}$$

* 2ND LAYER: ROBUST BAYESIAN TRUTH SERUM.

At the end of every SLA ~~[illegible]~~ timeslot the system will require from the users their opinion about their experience from fog node k:

Question: "Do you believe that you can find a more efficient fog node / Are you not satisfied?"

Possible Answers: YES or NO {High Signal (1) or Low Signal (0)}

⇒ NOTE: Users that choose another fog node in the next iteration will probably answer YES ⇒ But, if they want to mislead the other users (in order for example to experience lower congestion in the new node) they will answer NO.
So, we need to acquire truthfull information about the efficiency of every node k in order to "build" the corresponding reputation μ_k.

⇒ So, at the end of the timeslot (after the SLA converged) every user is asked from the network operator for 2 reports:

(1) Information report: Let $X_k^u \in \{0,1\}$ be user's u reported signal. (regarding fog node k)

(2) Prediction report: let $y_k^u \in [0,1]$ be user's u report about the frequency of high signals in the population.

Idea:
Incentivize users to answer truthfully ⇒ Bayesian updating argument.

• So, for each user u that is associated with fog node k:

→ Select a reference agent: $J_u = (u+1) \bmod |U_k|$. (Eq. 15)

→ Select a peer agent: $P_u = (u+2) \bmod |U_k|$. (Eq. 16)

and calculate:

$$y_u' = \begin{cases} y_{J_u} + \delta, & \text{if } X_k^u = 1 \\ y_{J_u} - \delta, & \text{if } X_k^u = 0 \end{cases} \quad \text{(Eq. 17)}$$

where $\delta = \min(y_{J_u}, 1 - y_{J_u})$ (Eq. 18)

- Respondent's u RBTS score (How truthful he is?)

$$U_u = \underbrace{R_q(y'_u, x^u_k)}_{\text{information score}} + \underbrace{R_q(y_u, x^u_k)}_{\text{prediction score.}} \quad \boxed{(Eq. 19)}$$

where $R_q(\cdot,\cdot)$ is the Strictly Proper Scoring Rule:

$$R_q(y, \omega=1) = 2y - y^2$$
$$R_q(y, \omega=0) = 1 - y^2 \quad \boxed{(Eq. 20)}$$

Why to be truth-tellers?

Because the actual reputation of how "good" or "bad" is a fog node k (μ_k) is built through the RBTS scores of the users ⇒ So, if a reputation is bad (but the fog node k is good) then the SLA reward will lead the users away from the good fog node (even the liers, who want to lie about the node).

⇒ So after a timeslot (after the SLA converges), and after the U_u RBTS computations (Eq. 19), we need to decide if the fog node has a "satisfying" behavior, or the most users (that are truthful) want to change their choice.

- For each answer (YES|NO), calculate the average RBTS score $\overline{U}_{k,ans}$ of all individuals in the specific fog node k endorsing answer "ans": $\overline{U}_{k,ans} = \dfrac{\sum \text{Scores of "ans"}}{|\text{users that answered "ans"}|}$ $\boxed{(Eq. 21)}$

- Find $ans^* = \arg\max\{\overline{U}_{k,ans}\}$ $\boxed{(Eq. 22)}$

- ans^*
  - → YES: The users believe that they can assosiate with a better fog node.
  - → No: The users are satisfied.

* 3RD LAYER - REPUTATION AS A BAYESIAN BELIEF

⇒ The reputation μ_k of a fog node that is updated in every timeslot can be modeled as a bayesian belief.

⇒ If a fog node k has large μ_k means that it has a good reputation (the customers are "satisfied") and for this reason it is more attractive.

⇒ This is depicted in the 1st Layer SLA Reward.

Initializations

- a_H^k → Probability that fog node k satisfies the users.
- a_L^k → Probability that fog node k does not satisfy the users.
- μ_k^0 → Prior distribution belief of how good is fog node k. It is common to all users.
- S_k → Times of ans* = NO.
- F_k → Times of ans* = YES.

⇒ Every fog node k is linked with a belief distribution μ_k.

⇒ After the end of a timeslot, we update the S_k or F_k.

Example

```
If (ans* = YES) then
        Fk = Fk + 1
else
        Sk = Sk + 1
end.
```

$\boxed{\text{(Eq. 23)}}$

⇒ So, after the end of every timeslot, the μ_k is updated for fog node k according to the bayesian belief:

$$\mu_k = \frac{\mu_0^k * a_H^{S_k} * (1-a_H)^{F_k}}{\mu_0^k * a_H^{S_k} * (1-a_H)^{F_k} + (1-\mu_0^k)\, a_L^{S_k} * (1-a_L)^{F_k}} \qquad \boxed{(\text{Eq. } 24)}$$

Diagram