Controlling Timeliness with a Costly Oracle

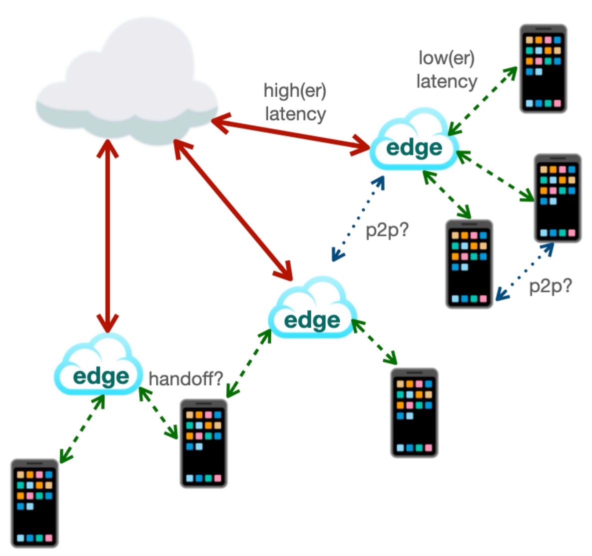
Nitya Sathyavageeswaran, Anand D. Sarwate, Roy D. Yates, Narayan Mandayam

Rutgers University

Motivation

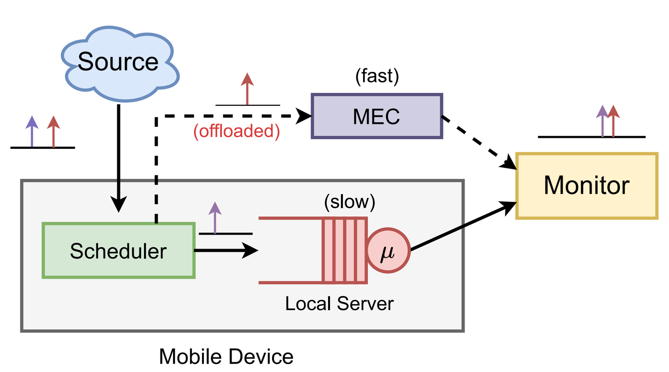
Real-time applications employ complex Machine Learning (ML) models.
For e.g., autonomous vehicles run ML algorithms to identify pedestrians, traffic signs, vehicles, objects, etc.
Applications can be run on mobile (or low- complexity devices).
Low latency is required for these applications.

Offloading some computation (e.g. to a mobile edge cloud) can help such devices.



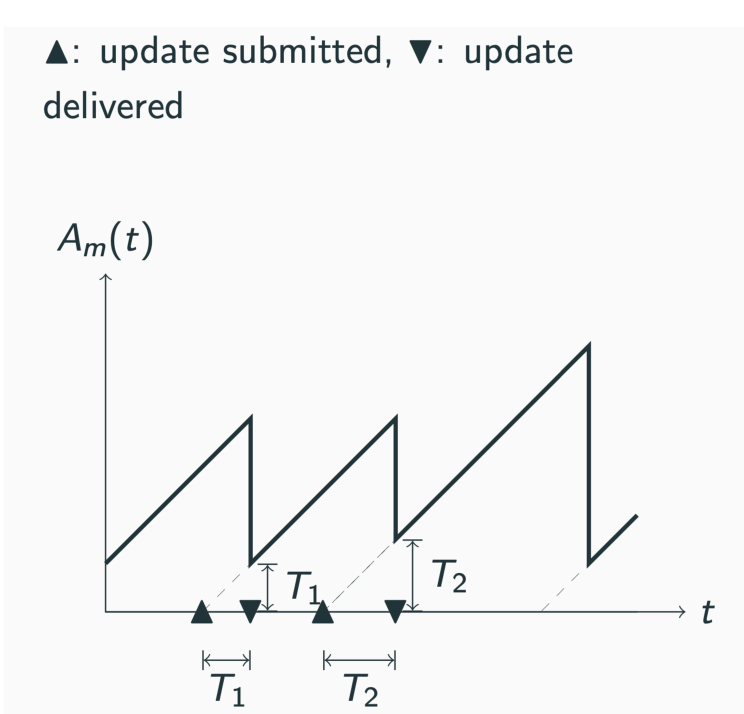
Problem setup

Source sends time- stamped updates to a monitor.
Device can process update at the local server or offload to a mobile edge cloud (MEC).
Processing at MEC yields timely information.
Timeliness at monitor measured by Age of Information (AoI).



Asking for help from the MEC is not free!

Age of Information Evolution

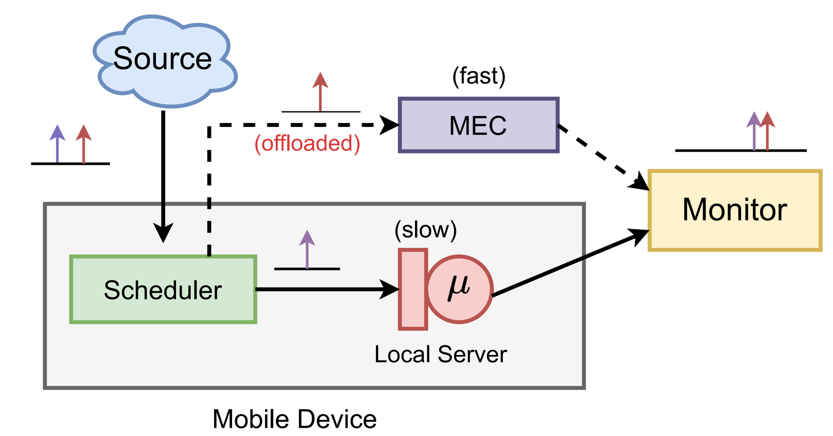


$$E[A_m(t)] = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} A_m(t)$$

$A_m(t)$: Age of most recently delivered update at the monitor.
Age increases linearly in time in the absence of updates at the monitor.

Scheduling Problem

Generate at will source: Source submits a fresh update only after the previous update is serviced, and received at the monitor.
Local server has service distribution $\mathrm{Geom}(\mu)$.
MEC services new update it receives from source in 1 time slot.



We model the scheduler's problem as an MDP.

Cost on edge:
We measure cost p_t by counting the number of times the mobile uses the MEC.

$$p_t = \begin{cases} 1 & \text{MEC services update in that slot} \\ 0 & \text{otherwise.} \end{cases}$$

Setting up an MDP

Model as a Markov Decision Process (MDP) that consists of $(\mathcal{S}, \mathcal{U}, P, C)$:
To track the current age/ age process we need to know two things:

- a_t: age in slot t
- z_t: cumulative service an update has received in slot t

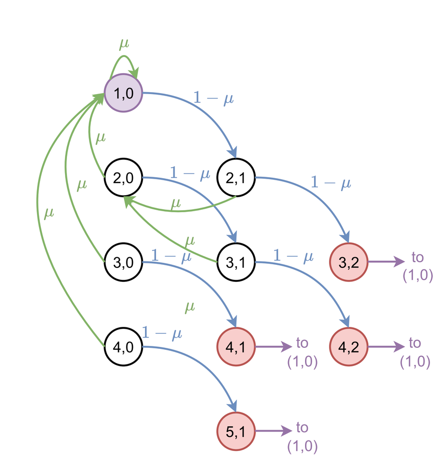
Define state of the system: $s_t = (a_t, z_t)$
P is the transition probability from state $s_t \to s_{t+1}$ defined as follows:

- When $u_t = 1$ (MEC update) , $s_{t+1} = (1,0)$
- When $u_t = 0$ (local server update) w.p. μ, $s_{t+1} = (z_{t+1}, 0)$ and w.p. $1-\mu$, $s_{t+1} = (a_{t+1}, z_{t+1})$.
- Cost in slot t modeled as : $c_t = C(s_t, u_t) = a_t + \lambda u_t$ where $\lambda \in (0, \infty)$.
- Minimize long term average cost : $V_\pi(s) = \lim_{T\to\infty} \frac{1}{T+1} E_\pi\left[\sum_{n=0}^{T} C(s_t, u_t)\right]$.

Heuristic Policies

Age threshold policy: Offload computing to MEC if $a_t > a^*$.
Service time threshold policy: Offload computing to MEC if $z_t > z^*$.

The optimal policy is an age threshold policy that depends on how long an update has been in service.



The age thresholds are monotonically non-decreasing in z.
We get larger age reduction by using the MEC when z gets bigger.

Results

For the z- dependent age threshold policy, for each z there is an age threshold $\bar{a}_z$ given by

$$\bar{a}_z = \min\{a : U(a,z) = 1\}$$

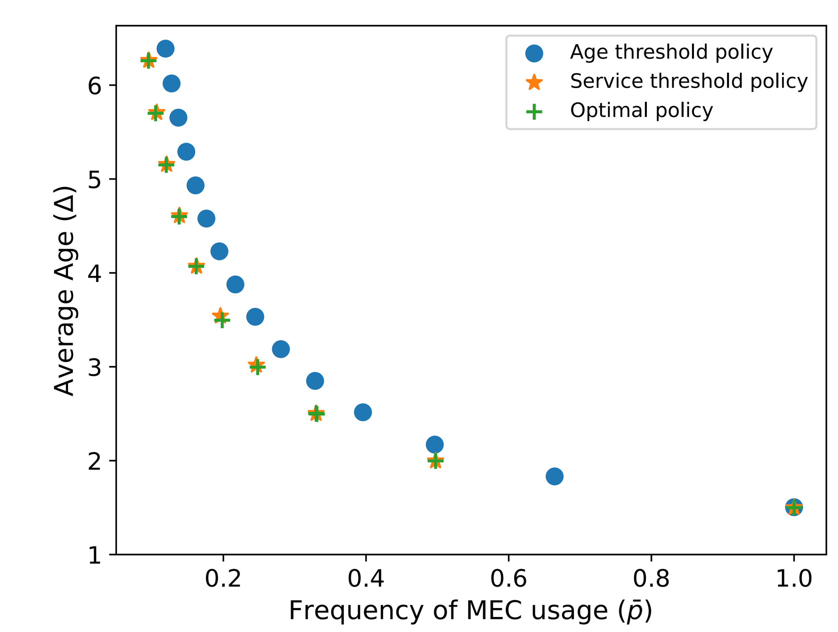
$$\bar{a}_0 \geq \bar{a}_1 \geq \bar{a}_2 \geq \ldots$$

Apply Relative Value Iteration (RVI) to find optimal policy.

Challenge: State space is infinite and cost is unbounded

Truncated state space: If age goes above $a_{\max}$, scheduler requests new update from source and sends it to the MEC.
State goes from $(a_{\max}, z) \to (1,0)$.
RVI converges to optimal solution in finite iterations if truncated MDPs are unichain.

- Any stationary policy for truncated MDP has only one recurrent class.
- Under any policy two consecutive service completions has non-zero probability.



Service rate of local server $\mu = 0.01$

As threshold (a^*, z^*)increases, MEC used more often and age decreases.
Need to compute relative values for each state recursively to implement optimal policy.

Service threshold policy is very close to optimal policy.

Conclusion

- In this work we balance timeliness of updates with MEC usage .
- Optimal Policy: Age threshold structure dependent on how long an update has been in service.
- Service threshold policy is close to optimal policy and is easier to implement in practice.

References

[1]Sennott, Linn I. "Average cost optimal stationary policies in infinite state Markov decision processes with unbounded costs." Operations Research 37, no. 4 (1989)
[2]Hsu, Yu-Pin, Eytan Modiano, and Lingjie Duan. "Age of information: Design and analysis of optimal scheduling algorithms." In 2017 IEEE International Symposium on Information Theory (ISIT), IEEE, 2017.

RUTGERS