Hyperprofile-based Computation Offloading in Mobile Edge Networks

Andrew Crutcher[1], Caleb Koch[2], Kyle Coleman[3], Jon Patman[4], Flavio Esposito[3], Prasad Calyam[4]

[1]Southeast Missouri State University, [2]Cornell University, [3]Saint Louis University, [4]University of Missouri-Columbia

Background and Motivation

Problem: Crowd-sourced mobile devices data (e.g., photos, vidoes) needs to be processed at network edges where resources are limited to help in applications such as disaster incident response.

Challenge: Processing tasks need to be efficiently assigned to edge computation nodes using emerging paradigms such as software-Defined Networking (SDN) that allow for centralized network control.

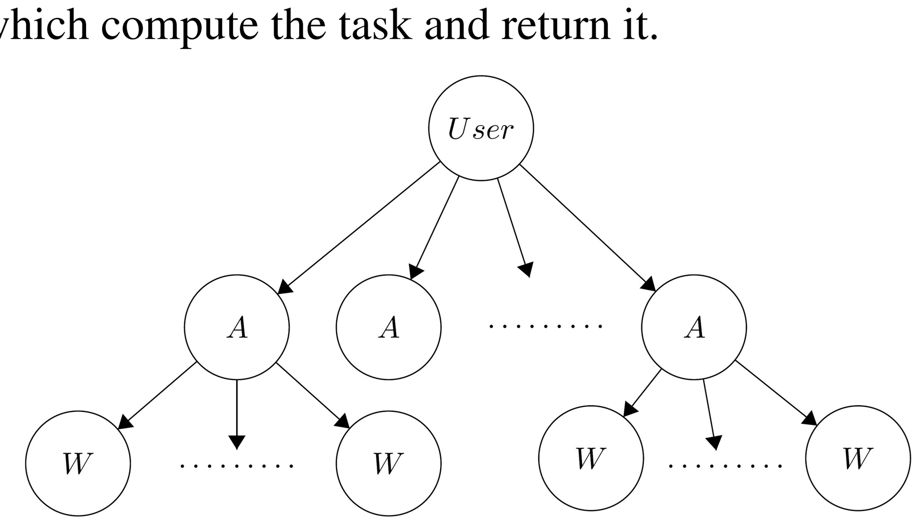
Our Solution Approach: We investigate machine learning approaches for computation offloading by using a Knowledge Plane coupled with SDN that allows for centralized data access and network analytics.

Objectives

- Develop network simulations to obtain a realistic network model and data sets to identify constraints such as e.g., energy, latency
- Use ML to dynamically predict network metrics and analyze network-wide resource status
- Develop profiles of edge network nodes from the metrics for efficient queries to assign computation offloading tasks

Hyperprofiles and Problem Representation

Figure 1: We consider a set of edge servers and a task to be offloaded as a *rooted tree* G. The root is the mobile device user. The branch vertices A are edge *aggregators* which collect a task, partition it and offload to workers/other aggregators. The leaves are *workers* W which compute the task and return it.



Our goal is to represent the root's children as a 'hyperprofile'.

Hyperprofile: A hyperprofile for a set of edge server nodes is a collection of profiles that consist dynamic metrics that are predicted in real-time by a predefined network model and related data sets.

****Acknowledgements:** This material is based upon work supported by the National Science Foundation under Award Numbers: CNS-1659134 and CNS-1647182 and CNS-1647084. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Network Modeling and Regression Analysis

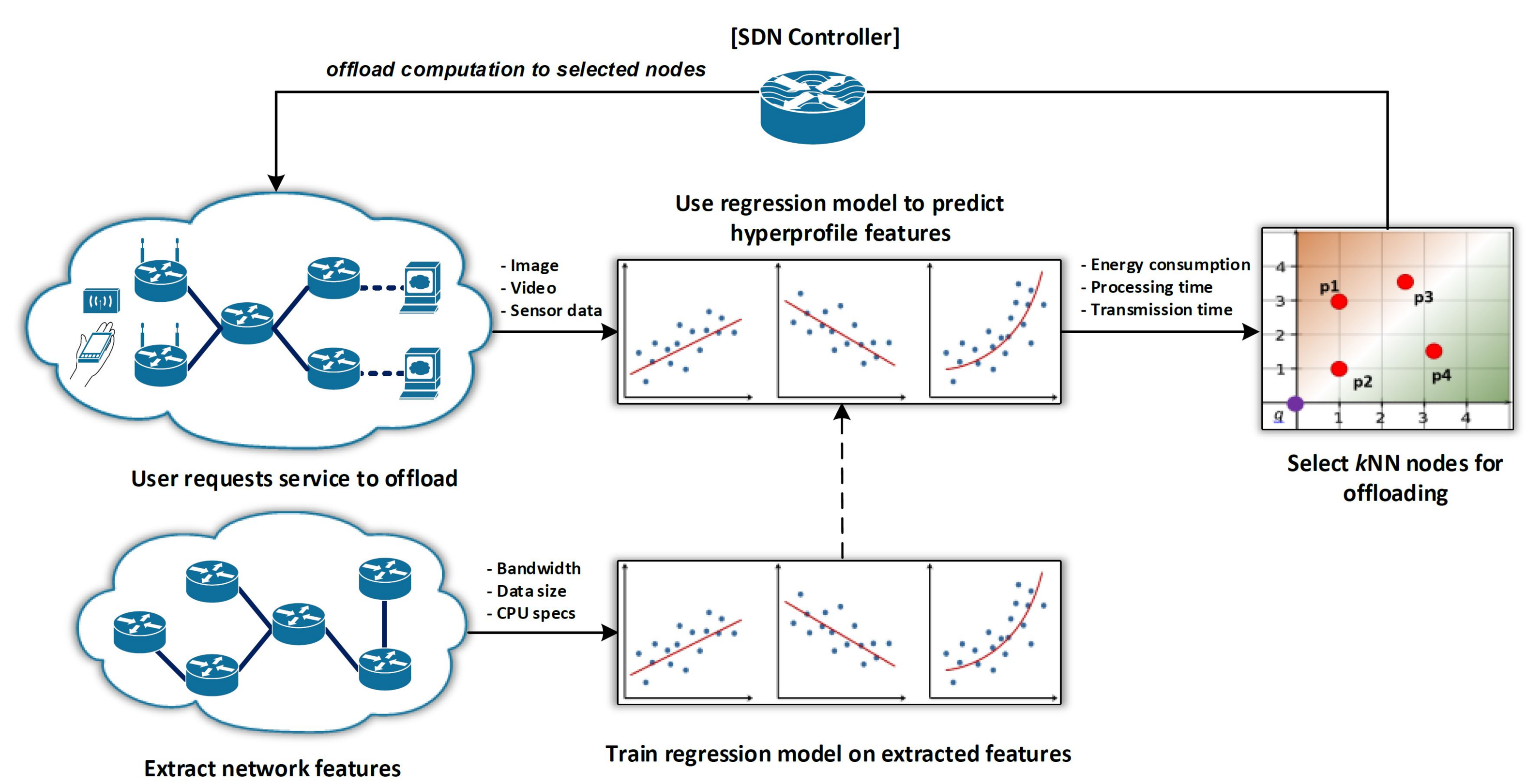


Figure 2: Diagram depicting process of creating a model to predict hyperprofile features used to determine nodes for use in computation offloading of a user request.

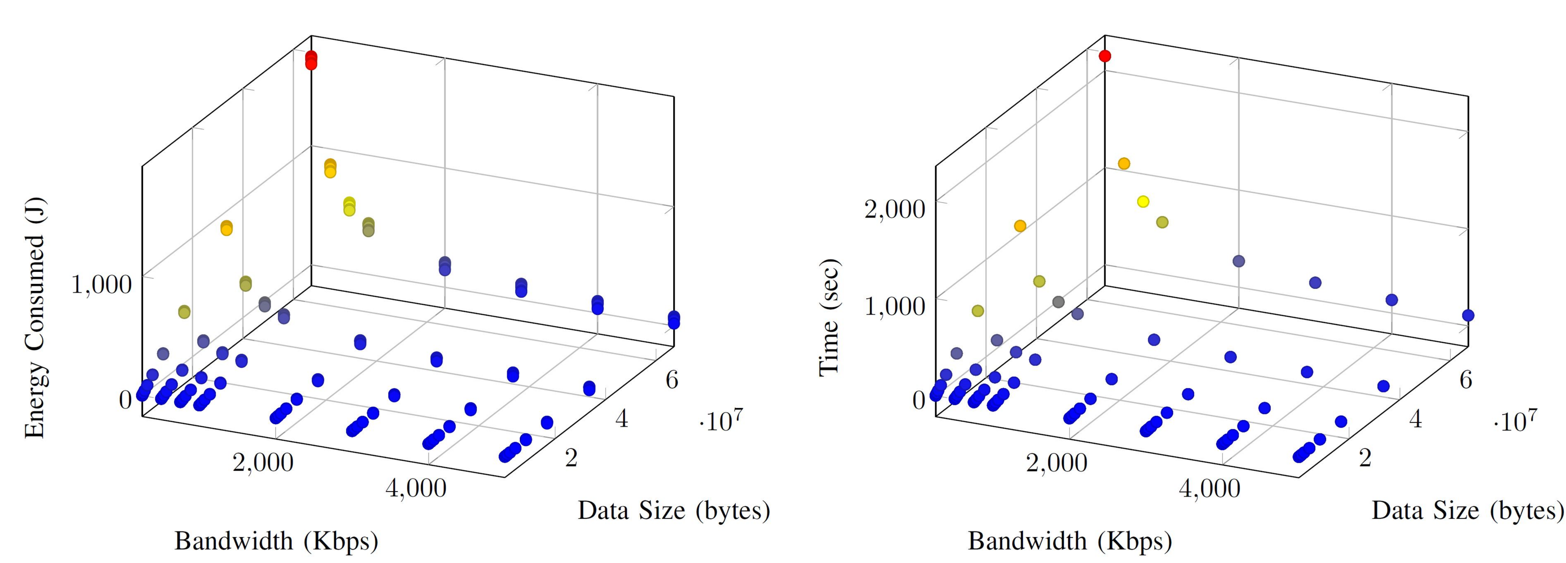


Figure 3: Three-dimensional plot of the energy consumption (left) and data transmission time (right) obtained through simulations using ns-3 simulator; the amount of data to send and bandwidth of the connection was varied across simulations.

We performed two-step regression that involved:

1. Developing linear models to correlated data size and energy consumption/transmission time for each bandwidth setting
2. Relating bandwidth to the linear model parameters

	Energy Consumption (e_c)	Time (t)
Bandwidth (b)	$m_1 = 0.015b^{-1.13}$ $R^2 = 0.997$	$m_2 = {}^{8.04 \cdot 10^6}/_b$ $R^2 = 1$ $c = 222873e^{0.0004b}$ $R^2 = 0.918$
Data Size (d_s)	$e_c = m_1 d_s$ Cross-validation: 0.99	$t = m_2 d_s + c$ Cross-validation: 0.99

Figure 4: The equations for calculating energy consumption and transmission time obtained from applying regression to the data we received from our simulations along with their R^2 and k-fold cross-validation score where $k = 10$.

Experimental Analysis

If the hyperprofile is a set of points, then our kNN approach effectively minimizes the sum of squares of the coordinates.

k–Nearest Neighbor

A point $p \in P$ is in the result of $kNN(q)$ if and only if

$$|\{o \in P | dist(o, q) < dist(p, q)\}| < k$$

We can use this to query points in the hyperprofile.

We evaluated using a min-heap versus an R-tree for kNN.

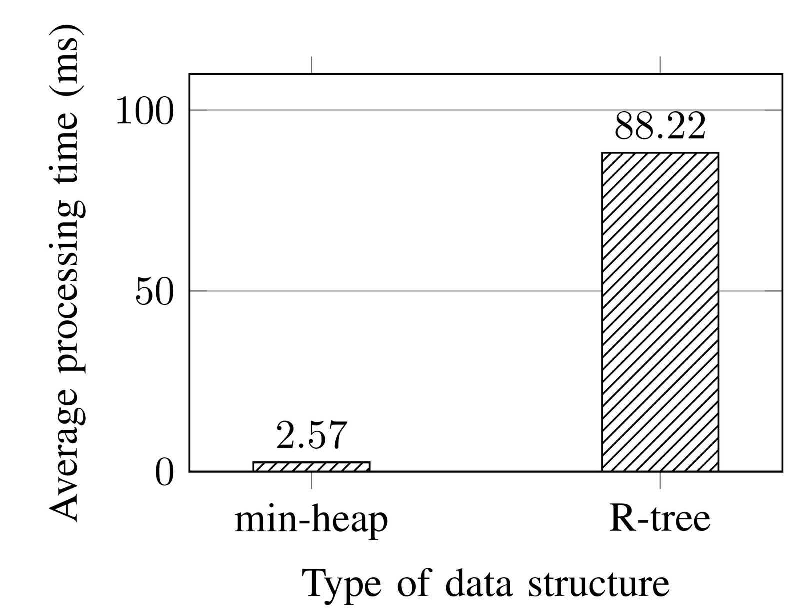


Figure 5: Running time results of inserting data points into the respective data structures and then determining the kNN of a query point at the origin.

Conclusion

Our main contributions include:

- We showed that network metrics can be encoded meaningfully into a multidimensional space
- Using machine learning to compute hyperprofiles in the knowledge plane is a viable approach to select nodes for computation offloading
- We investigated relevant data structures for kNN queries along with how kNN differs from other approaches

Future Work

The idea of using hyperprofiles and knowledge plane can be more broadly applied to other edge computing and networking problems:

- Setup an experiment to evaluate a hyperprofile based offloading scheme compared to standard schemes
- Expand our idea of hyperprofile-based resource allocation to areas (e.g., trust management) other than just computation offloading
- Explore the construction of feature spaces; investigate fitness profiles according to application requirements in edge networks