A Study on Edge Proactive Caching for Vehicular Networks Using Fed-EASE

Ida Nurcahyani, Jeong Woo Lee*

Chung-Ang University

idanurcahyani@cau.ac.kr, *jwlee2@cau.ac.kr

Abstract

Edge caching for vehicular networks is considered as an important solution to provide contents closer to the end user. In this study, we investigate the impact of caching strategy to the transmission delay by implementing Fed-EASE, a simple algorithm that combines Autoencoder and items neighborhood to predict interest score for items. Simulation result shows that compared to the popularity-based method, the Fed-EASE algorithm performs better by providing lower delivery delay to the end user.

Ⅰ. Introduction

Mobile Edge Caching (MEC) is one of the promising solutions to satisfy users' demand by integrating caching at the edge networks. It virtualized the network resources to enable service execution at the edge network. Users are able to download their requested contents within one hop, thus minimizing service latency. Furthermore, it can alleviate the backhaul traffic burden. Typically, an edge network has limited storage capacity to store chosen contents that can satisfy users' requests. Therefore, edge network should determine the caching strategy to identify popular contents that are interesting and attract most vehicular users.

Proactive caching is a type of caching strategy that uses machine learning (ML) to leverage users interests in predicting popular contents and cache it before the actual request arrives. This mechanism is more suitable for high mobility network caching scenarios than the reactive caching where contents are cached if there are request for them.

Several researches have been done to investigate the impact of ML in improving caching efficiency and accuracy [1], [2]. ML is a tool that can be utilized to extract hidden features between users and requested contents to predict popular contents based on users' interaction history. The transmission delay becomes an important matric to determine how well the designed system in providing real-time vehicle services. Therefore, in this paper we want to investigate the impact of implementing caching strategy in edge network to the end user's delay transmission performance.

Ⅱ. Method

We consider a vehicular network with one Base Station (BS) that is connected to the Content Data Center (CDC), several Road Side Units (RSUs) as edge clients, and vehicular users (VUs). Each RSU has its coverage with active VUs run along a road with certain topological structure. VUs arrive at its serving RSU based on a Poisson distribution with arrival expectation μ. Each arrival triggers a service request to the serving RSU. We assume that during a round, the number of active VUs in each RSU is fix, means that no vehicle is coming or exiting the coverage area of the RSU.

When user x requests a content y to its serving RSU m, local RSU then checks its local storage. If requested content is available, then serving RSU sends it directly to the requester. Otherwise, RSU fetches the requested content to the BS, copy it to its local storage, and sent it back to the requester.

Active VUs use the orthogonal frequency division multiplexing (OFDM) to communicate with its serving RSU, thus no interference is considered in the telecommunication model. RSUs communicate with each other and to the BS using wireless network.

We calculate the download delay experienced by user x in serving RSU m as:

$$d_{x,m}^{dl} = \frac{a_y}{r_{x,m}^{dl}} + \left(1 - v_y\right)\frac{a_y}{r_{m,BS}^{dl}} \tag{1}$$

Where a_y is the content y data size, $r_{x,m}^{dl}$ and $r_{m,BS}^{dl}$ are the achievable data rate for the downlink communication between client m and user x, and between client m and BS respectively. We denote v_y as a decision variable whether content y is placed in the edge client, where $v_y \in \{0,1\}$. The value of $v_y = 1$ means that file y is cached and 0 otherwise. The caching strategy tries to assign $v_y = 1$ to the highest predicted score according to the available storage capacity. Hence, the caching strategy tries to predict what users want to access to the local server by pre-storing the predicted contents.

We want to predict the best content that can be stored in the caching system so that we can minimize the delay transmission. This is done by computing the interest probability score P of item y by implementing prediction score calculation based on users' past interaction data.

Each VU stores its interaction historical data $D_{x,m}$ that reflects user's information and its past service requests. We leverage user's interaction historical data $D_{x,m}$ at time t when accessing an edge client and requesting a content. This historical data $D_{x,m}$ is used to predict the item interest score $\boldsymbol{P}$. We use user-item binary matrix $\boldsymbol{D} \epsilon \{0,1\}^{|\mathcal{X}_m| \times |\mathcal{J}|}$ to illustrate the relationship of a set user $\mathcal{X}$ in client $\boldsymbol{m}$ with a set of content $\mathcal{J}$. In this matrix, we assign a positive value 1 in $\boldsymbol{D}$, or $D_{x,j} = 1$, if a user has interaction with certain contents, and $D_{x,j} = 0$ if no interaction has been observed.

We aim to predict the non-zero entries in the user-item matrix $\boldsymbol{D}$ that indicates instances where contents would align with users' preferences. These predictions also serve as indicators that corresponding contents can attract users' interest and are thus suitable to be cached in the edge caching systems. We implemented Federated Embarrassingly Shallow Autoencoder for Sparse Data (Fed-EASE) that is based on [3] to calculate the prediction of content interest scores.

We construct an item-to-item weight matrix $\boldsymbol{V} \in \mathbb{R}^{|\mathcal{J}| \times |\mathcal{J}|}$ that is used to compute items neighborhood. The predicted interest score $\boldsymbol{P}$ is calculated as the sum of similarities between contents in the user's history and the available contents:

$$\text{score}(x, j) = \textstyle\sum_{y \in \mathcal{J}_x} \boldsymbol{P}_{y,j} = (\boldsymbol{D}_x \cdot \boldsymbol{V})_j \qquad (2)$$

Here $\boldsymbol{D}_x$ is the $\boldsymbol{x}^{\text{th}}$ row of $\boldsymbol{D}$. Computing all user interactions involves the calculation of matrix multiplication $\boldsymbol{DV}$, which will be more efficient if the matrix $\boldsymbol{V}$ is sparse. To learn the weight matrix $\boldsymbol{V}$, we use the square loss of the actual data $\boldsymbol{D}$ and the predicted score $\boldsymbol{P} = \boldsymbol{DV}$ with regularization and the value of $\text{diag}(\boldsymbol{V}) = 0$ as:

$$\|\boldsymbol{D} - \boldsymbol{DV}\|_F^2 + \lambda \cdot \|\boldsymbol{V}\|_F^2 \qquad (3)$$

The Frobenius norm of the formula reflects a matrix inversion of the L-2 norm and the Euclidean distance.

We utilized MovieLens 100k dataset to investigate our system performance. This simulation involves three edge clients that train their local model and upload it to the BS. Then BS aggregates it and sends the new global model to all clients. The result in Fig. 1 shows that Fed-EASE performs better than the popularity-based algorithm in terms of average delivery delay. The delay value decreases when caching storage becomes higher due to more contents are available in the edge local storage. Thus, shortening the access path of contents.

Ⅲ. Conclusion

This study investigates the effect of caching strategy using machine learning, particularly Fed-EASE method to determine the cached contents in edge network local storage. Although the result is better than the popularity-based method, the investigation to determine the correlation between the caching-accuracy and the end-delay should be performed furthermore. Therefore, we can determine the best strategy to minimize delay delivery with implementing proactive caching at the edge networks.

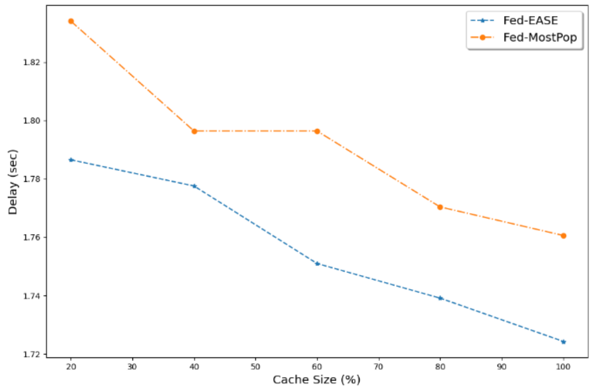

Fig. 1 Delay simulation result

ACKNOWLEDGMENT

"This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program(IITP-2024-RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)"

REFERENCES

[1] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," IEEE Access, vol. 5, pp. 6757–6779, 2017, doi: 10.1109/ACCESS.2017.2685434.

[2] B. Jedari, G. Premsankar, G. Illahi, M. DI Francesco, A. Mehrabi, and A. Yla-Jaaski, "Video Caching, Analytics, and Delivery at the Wireless Edge: A Survey and Future Directions," IEEE Communications Surveys and Tutorials, vol. 23, no. 1, pp. 431–471, 2021, doi: 10.1109/COMST.2020.3035427.

[3] H. Steck, "Embarrassingly shallow autoencoders for sparse data," The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, pp. 3251–3257, 2019, doi: 10.1145/3308558.3313710.